A CMOS GENERAL-PURPOSE SAMPLED-DATA ANALOGUE MICROPROCESSOR

Piotr Dudek and Peter J. Hicks

Department of Electrical Engineering and Electronics
University of Manchester Institute of Science and Technology (UMIST)
PO Box 88, Manchester M60 1QD, United Kingdom
pdudek@sn1.ee.umist.ac.uk, p.j.hicks@umist.ac.uk

Abstract

This paper presents a general-purpose sampled-data analogue processing element that essentially functions as an analogue microprocessor (AμP). The AμP executes software programs, in a way akin to a digital microprocessor, while nevertheless operating on analogue sampled data values. This enables the design of mixed-mode systems which retain the speed/area/power advantages of the analogue signal processing paradigm while being fully programmable, general-purpose systems. A proof-of-concept integrated circuit has been implemented in 0.8 μm CMOS technology, using switched-current techniques. Experimental results and examples of the application of the AμPs in image processing are presented.

1. Introduction

Although analogue integrated circuits can offer advantages over their digital counterparts, in terms of speed, power dissipation and silicon area consumed by the circuitry, digital circuits are often a preferred solution in cases where programmability is required. In particular, digital signal processors or microprocessors offer a large degree of flexibility, as the functionality of the system can be determined solely through software development. Recent research in the field of programmable analogue circuits resulted in the development of reconfigurable devices [1-3] which can be generally thought of as the analogue equivalent of digital FPGAs. Algorithmically programmable analogue chips, based on the Cellular Neural Network (CNN) operation, have been also introduced [4]. In this paper we describe the analogue microprocessor (AμP), which executes software programs, and can be thought of as an analogue equivalent of a digital microprocessor.

The idea of an instruction-level programmable analogue processor has previously been described by Masuda *et al* [5]. The architecture presented there was based on charge-domain operations and the circuitry required high-gain amplifiers, capacitors and analogue switches. However, this discrete-component implementation, although providing proof-of-concept, exhibited rather poor performance. Recent advances in analogue sampled-data signal processing techniques, and in CMOS technology, allow for the efficient implementation of the analogue microprocessor as a switched-current circuit. Our results show that the AμP can achieve savings in terms of silicon area and power dissipation, when compared with digital processors. These, especially when combined with parallel processing techniques, can enable the design of low-cost high performance systems.

One of the application areas where it will be advantageous to use AμPs is in low-level image processing [6]. Massively parallel SIMD (Single Instruction Multiple Data) arrays of mesh-connected digital processing elements have long been known to be efficient in executing early-vision algorithms [7]. The area-efficient implementation of a processing element is of primary importance, as it enables the integration of thousands of processors onto a single die, and thereby fully exploits the inherent fine-grain parallelism of early-vision tasks by realising pixel-per-processor correspondence [8]. The AμP described in this paper exhibits high performance/area and performance/power ratios and therefore is very suitable to be used as a processing element in the massively parallel array.

2. Analogue microprocessor architecture

The block diagram of the generic AμP architecture is presented in Figure 1. The AμP consists of a register file (each register is an analogue memory cell, capable of storing a sample of data), an analogue ALU (Arithmetic Logic Unit), and an analogue I/O port. All the building blocks are interconnected via an analogue data bus. The processing of information is performed entirely on analogue values. However, in a way akin to a digital microprocessor, the AμP executes a software program, performing consecutive instructions issued by a digital controller. These instructions may include register transfer operations, which move the analogue samples of data between registers of the AμP, I/O operations which move the data to and from I/O ports, arithmetic operations, which modify the analogue data, and comparison operations, which allow for conditional branching. The program is stored in the local memory of the controller, which is a purely digital device. The complete processor is therefore a mixed-mode system, with an analogue data-path and a digital control-path.

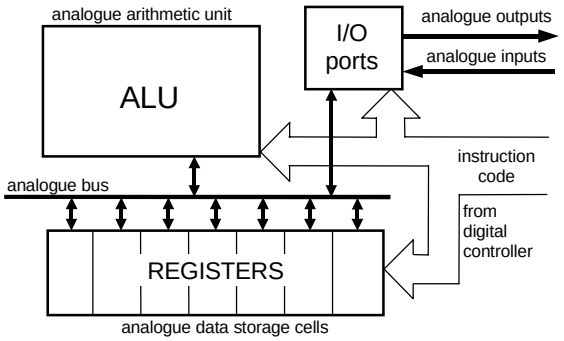

Figure 1. The block diagram of the AμP.

0-7803-5482-6/99/$10.00 ©2000 IEEE

3. Switched-currents implementation.

The AμP is implemented utilising switched-current (SI) techniques [9]. Switched-current circuits have become a feasible alternative to switched-capacitor (SC) circuits for the implementation of analogue, sampled-data systems. They can be realised in standard, digital CMOS process technology, and therefore are particularly suited for the design of mixed-mode systems. The simple structure of SI cells results in area- and power-efficient AμP implementations while offering adequate speed and accuracy to satisfy a wide range of applications.

3.1 The register file

The schematic diagram of a simplified AμP is presented in Figure 2. The depicted register file comprises four registers, each of which is a basic SI memory cell consisting of a memory transistor M_X (X=A,B,C,D), a current source, and two switches W_X and S_X. Consider simple register-transfer operation, which can be denoted as **A←B**. To execute this instruction the switches W_A, S_A and S_B are closed, the remaining ones are open. Therefore, the only non-zero currents entering the analogue bus node will be the current i_B, which is the current read out from register B, and the current i_A, which is being written to register A. Of course, it is also true that:

$$i_A = -i_B \tag{1}$$

Since W_A is closed, the transistor M_A is diode-connected and therefore its gate-source voltage, V_{gsA}, will set itself to the value corresponding to the drain current, as described by the saturation-region equation:

$$I_{REF} - i_A = I_{dsA} = K_A(V_{gsA}-V_t)^2 \tag{2}$$

If now the switch W_A is opened, a quantity of charge will be stored at the gate capacitance of the transistor M_A, and the gate-source voltage V_{gsA} will hold its value (for the purpose of this analysis we disregard any error effects). As long as the switch W_A remains open, each time the switch S_A is closed the drain current of M_A will be set by the gate-source voltage V_{gsA} (it is assumed that the memory transistors are in saturation when their corresponding S-switches are closed) and will be equal to $I_{dsA}=I_{REF}-i_A$, where i_A is the value of the input current at the time the switch W_A was opened. In this way, the current $i_A=-i_B$ is stored in register A. (By default, all switches revert to the open position after an instruction has been executed. For correct operation, it is also necessary to ensure that W-switches always open before S-switches.)

3.2 The analogue ALU

The analogue ALU is required to provide the basic arithmetic operations of addition, inversion and multiplication/division. However, as can be seen from (1), the inversion is inherent in the basic current-transfer operation. Moreover, the addition operation in a current-mode system is performed on the analogue bus with no area overhead, using current summation. For example, to execute instruction **D←B+C**, the switches W_D, S_D, S_B and S_C are closed, the remaining ones are open, and the current stored in register D is equal to:

$$i_D = -(i_B+i_C) \tag{3}$$

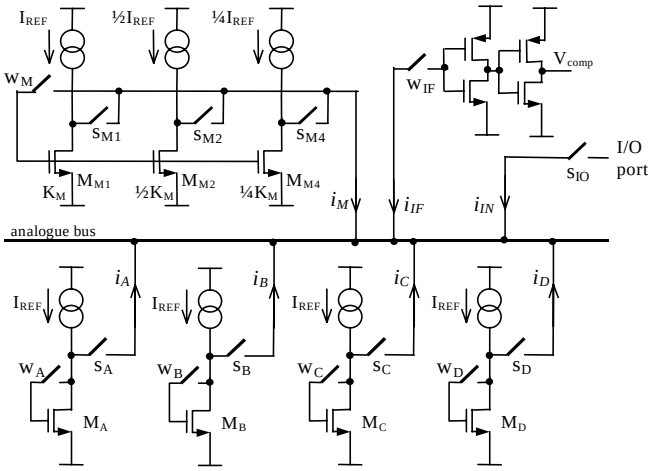

Figure 2. The conceptual circuit diagram of a simple, switched-current analogue microprocessor

Therefore only a multiplier/divider needs to be physically implemented. The multiplier is constructed as a set of current-mirrors, with binary scaled transistors M_{M1}, M_{M2} and M_{M4}. This is enough to realise the multiplication of an analogue value by a digital constant. The current i_M is stored in the multiplier, just like in another register, by closing switches W_M and S_{M1}.We get:

$$I_{dsM1} = I_{REF} - i_M = K_M(V_{gsM}-V_t)^2 \tag{4}$$

The current read out from the multiplier, i_M', depends on the multiplication factor k. This is a binary word which selects the appropriate mirrors using switches S_{M1}, $S_{M2,}$ S_{M4}:

$$i_M' = k\cdot I_{REF} - k\cdot K_M(V_{gsM}-V_t)^2 = k\cdot i_M \tag{5}$$

As a current comparator a simple CMOS buffer is used. The output voltage V_{comp} will be determined by the current charging or discharging its high-impedance input node. This voltage provides the controller with the comparison results, allowing for conditional branches.

An input/output port is simply realised by an analogue switch S_{IO}, which connects the port node to the analogue bus.

3.3 Program execution

All of the switches within the AμP are operated in response to logic-level voltages set by a digital controller. The complete set of these voltages, controlling all switches, forms the instruction-code word (ICW). The sequence of the ICWs issued by the controller constitutes a machine-level software program. This program dictates the way the samples of data are transferred and manipulated within the processor, allowing the software implementation of the required processing algorithm.

To further illustrate the operation of the AμP consider an example program presented in Table I. High-level description, resulting machine level code, ICWs and resulting current-transfer equations are shown.

4. Accuracy

An important issue with analogue circuits is the accuracy of processing. Apart from noise, the major error sources in SI circuits are charge injection effects in analogue switches [10], voltage

coupling through the gate-source capacitance of transistors and the finite output conductance of the transistor. The errors in SI memory cells will cause the AμP instructions to be performed with a limited accuracy. Consider the transfer instruction **A←B**. In the non-ideal case, the current transfer is performed with an error, which consists of a systematic part $\Delta_S(i_B)$, and a random noise $\Delta_N(*)$. The systematic error can be split into the signal-independent part Δ_{SI} and the signal-dependent part $\Delta_{SD}(i_B)$.

$$i_A = - i_B + \Delta_S(i_B) + \Delta_N(*) \quad (6)$$

$$\Delta_S(i_B) = \Delta_{SI} + \Delta_{SD}(i_B) \quad (7)$$

Many methods have been proposed to reduce the error effects in SI circuits [8], however, the more sophisticated methods of error compensation in SI cells require more complex circuitry, and therefore the design of an AμP will involve trade-offs between accuracy, speed, area and power. A particularly good compromise between cell area and accuracy can be achieved using the S^2I technique which offers significant signal-dependent error reduction [11].

Signal-independent error cancellation can be easily achieved by appropriate sequencing of the instructions. For example, consider variable assignment from A to B. The basic transfer instruction of the AμP performs inversion, so the assignment will be performed in two steps, using auxiliary register C: first transfer **C←A**, followed by **B←C**. Now, assume that each transfer instruction is performed with a constant signal-independent error Δ_{SI} (i.e. neglecting signal-dependent errors, noise and errors arising from device mismatches). For the first transfer we get:

$$i_C = - i_A + \Delta SI \quad (8)$$

And as a result of the second transfer instruction we get:

$$i_B = - i_C + \Delta_{SI} = i_A - \Delta_{SI} + \Delta_{SI} = i_A \quad (9)$$

The errors cancel out and an assignment operation that is free of signal-independent errors is achieved.

Similarly, as shown in Table II, complete cancellation of the signal-independent error can be achieved for the addition, subtraction and multiplication operations.

TABLE II. Machine-level instruction sequences for various arithmetic operations with signal independent error cancellation.

Operation	instructions	current transfers showing signal-independent error
Addition C := A + B	D ← A+B C ← D	$i_D = -(i_A+i_B) + \Delta_{SI}$ $i_C = -i_D + \Delta_{SI} = i_A+i_B$
Subtraction C := A – B	D ← A C ← B+D	$i_D = -i_A + \Delta_{SI}$ $i_C = -(i_B+i_D)+\Delta_{SI} = i_A - i_B$
Multiplication A := k * A	M ← A A ← M(k)	$i_M = -i_A + \Delta_{SI}$ $i_A' = -ki_M+\Delta_{SI} = ki_A+(1-k)\Delta_{SI}$
Multiplication (complete Δ_{SI} cancellation) A := k * A	B ← A M ← B B ← M(k) A ← B	$i_B = -i_A+\Delta_{SI}$ $i_M = -i_B+\Delta_{SI} = i_A$ $i_B' = -k\, i_M+\Delta_{SI} = -k\, i_A+\Delta_{SI}$ $i_A' = -i_B'+\Delta_{SI} = k\, i_A$

5. Test Chip

Our implementation of the AμP is targeted at a processor array, intended for low-level image processing. For this application, the primary consideration is the silicon area occupied by a single processing cell. Also the power consumption must be kept within certain limits. On the other hand, our analysis shows that for the majority of low-level image processing tasks a moderate level of accuracy, equivalent to 5 or 6-bits, is adequate. The use of six to eight registers and a multiplier resolution of 3 to 4 bits should also be sufficient.

To aid the evaluation of different design trade-offs, we have designed and fabricated an integrated circuit containing 15 AμPs, using various error-cancellation methods and different transistor sizes. The silicon area occupied by a basic register cell varies therefore from 17 μm × 39 μm to 54 μm × 57 μm. The chips were fabricated through EUROPRACTICE using the standard 0.8 μm CMOS process from AMS. The AμP circuits operate with a 3.3 V power supply voltage and were tested, performing various algorithms, using a laboratory data generator as an external controller. For different processor designs we have obtained magnitudes of the signal-dependent error of a single instruction from 0.2 % to 3.5 %, with processors operating at speeds from 70 kHz to 4 MHz.

Consider one of our AμPs, built using the S^2I error compensation technique. The processor works satisfactorily with a clock frequency of up to 2.5 MHz. The nominal reference current level is equal to 10μA. The total power dissipation within the processor is less than 100 μW. (To reduce power consumption only current sources required by a particular instruction are enabled). The effective area occupied by the processor, comprising six registers and a 3-bit multiplier, is equal to 11200 μm^2. As the typical assignment or arithmetic operation will take two clock cycles, the performance/area ratio for this processor is equal to 0.11 GOPS/mm^2 (Giga Operations Per Second per mm^2). The performance/power ratio is equal to 12.5 GOPS/W. As can be seen from Table III, these figures of merit compare quite favourably

TABLE I. An example program. High-level description is compiled to obtain a machine-level code. The bit values "0" in the instruction code word correspond to the appropriate switch opened, the values "1" denote closed switches.

high-level program	instruction mnemonic	instruction code word (ICW)														current transfers
		s_{IO}	w_A	s_A	w_B	s_B	w_C	s_C	w_D	s_D	w_M	s_{M1}	s_{M2}	s_{M4}	w_{IF}	
main () {	A←IN	1	1	1	0	0	0	0	0	0	0	0	0	0	0	$i_A = - i_{IN}$
VarA = inport(IN);	M←A	0	0	1	0	0	0	0	0	0	1	1	0	0	0	$i_M = - i_A$
VarB = 0.75 * VarA;	B←M(½+¼)	0	0	0	1	1	0	0	0	0	0	0	1	1	0	$i_B = -(½+¼)i_M = 0.75\, i_A$
VarC = VarA – VarB;	D←A	0	0	1	0	0	0	0	1	1	0	0	0	0	0	$i_D = - i_A$
}	C←B+D	0	0	0	0	1	1	1	0	1	0	0	0	0	0	$i_C = - (i_B+i_D) = i_A - i_B$

with those of digital processors. Of course, comparing "operations per second" figures for different architectures is somewhat difficult, as for example a RISC microprocessor [12] is a much more complex and versatile device than the AμP. Nevertheless, simple digital processors intended for massively parallel processors arrays [8,13] provide similar levels of functionality to the AμP. It must be recognised, however, that the AμP has the drawback of a limited accuracy, which needs to be considered.

The maximum magnitude of the signal-dependent error Δ_{SD}, measured for a register transfer instruction, for the S^2I processor, is equal to 250 nA, that is 2.5 % of the maximum signal level of 10 μA. Random errors associated with a single transfer $\Delta_N(*)$ were measured to be equal to 40 nA RMS, that is 0.4 % of the maximum signal level. At room temperature, the analogue value held in the register decays due to the leakage currents at a rate of 0.5 % per 100 ms. The design trade-offs were resolved in favour of small cell size, which resulted in small memory-transistor capacitances and in consequence relatively large errors. Nevertheless, many applications, particularly in low-level image processing, are not very sensitive to the errors introduced by the AμP. As an example, consider the edge detection problem. A software for the AμP, implementing the Sobel edge detection algorithm [14] has been written, and executed on the S^2I processor. The processing results are presented in Figure 3. In this example the image was processed serially on a single processor clocked at 2.5 MHz. Pixel values were fed to the processor using a D/A converter, and the result read out using an A/D converter. The processing speed was therefore relatively low. However, small cell size and low power dissipation are the key features that enable massive parallelism. A very high performance system could be built by integrating a large number of processors. An SIMD array of 128×128 such AμPs could be feasibly accommodated on a single die and, when clocked at 2.5MHz, perform algorithms with a speed of over 20 GOPS while dissipating less than 2 W of power.

6. Conclusion

We have presented a general-purpose analogue microprocessor whose architecture is analogous to that of its digital counterpart. The AμP executes software programs while operating on analogue data values. The AμP paradigm will find application in areas that can benefit from employing analogue signal processing techniques, but where nevertheless the flexibility of a software-programmable device is needed. As an example we have considered a massively parallel processor array, targeted at image processing applications. The high performance/area and performance/power ratios exhibited by the switched-current AμP will allow for a great number of processors to be integrated onto a single chip, resulting in the development of low-cost high-performance systems.

TABLE III. Performance/area [MOPS/mm^2] and performance/power [GOPS/W] ratios for the AuP and digital processors.

Processor	MOPS / mm^2	GOPS / W	Comments
AμP	110	12.5	S^2I analogue microprocessor
Pixel-Parallel [8]	64.1	9.34	Bit-serial PE from a massively parallel processor array. (per-formance for 8-bit additions)
IMAP [13]	36.3	3.67	8-bit PE from a 64 processor SIMD array.
DECchip A21064 [12]	1.72	0.013	High-performance 64-bit & floating point RISC micro-processor.

(a) (b) (c)

Figure 3. (a) An input image, (b) An ideal edge map calculated by applying the Sobel algorithm (c) Experimental results obtained by the execution of the algorithm on the AμP

References

[1] L. H. Lu and C. Y. Wu, "The design of the CMOS current-mode general purpose analog processor", in Procs. Int. Symposium on Circuits and Systems, ISCAS'94, New York, vol.5 pp.549-552, 1994

[2] D.L.Grundy, "A Computational Approach to VLSI Analog Design", in Journal of VLSI Signal Processing, vol.8, pp.53-60, 1994

[3] Bratt and I. Macbeth, "DPAD2 – a field programmable analog array", Analog Integrated Circuits and Signal Processing, vol.17, no.1-2, pp. 67-89, Sept. 1998

[4] T. Roska and L. O. Chua, "The CNN Universal Machine: An Analogic Array Computer", in IEEE Transactions on Circuits and Systems-II:Analog and Digital Signal Processing, vol.40, no.3, pp.163-173, March 1993

[5] S. Masuda, S. Yoneda and T. Kasai, "Sampled-data charge processor", in International Journal of Electronics, vol.58, no.5, pp.743-760, May 1985

[6] P.Dudek and P.J.Hicks, "An SIMD Array of Analogue Microprocessors for Early Vision", Procs. Conf. on Postgraduate Research in Electronics, Photonics and Related Fields (PREP'99), Manchester, UK, pp.359-362, 1999

[7] K.E. Batcher, "Design of a massively parallel processor", IEEE Transactions on Computers, vol.29,no.9, pp.837-840, Sept 1980.

[8] J. C. Gealow and C. G. Sodini, "A Pixel-Parallel Image Processor Using Logic Pitch-Matched to Dynamic Memory", IEEE Journal of Solid-State Circuits, vol.34, no.6, pp.831-839, June 1999.

[9] Toumazou, J. B. Hughes and N. C. Battersby (Eds.), "Switched-Currents: An Analogue Technique for Digital Technology", Peter Peregrinus Ltd., London, 1993.

[10] G. Wegmann, E.A.Vittoz and F.Rahali, "Charge Injection in Analog MOS Switches",IEEE Journal of Solid-State Circuits, vol.22, no.6, pp.1091-1097, December 1987.

[11] B. Hughes and K. W. Moulding, "S2I: A Switched-Current Technique for High Performance", in Electronics Letters, vol.29, no.16, pp.1400-1401, August 1993

[12] D.W.Dobberpuhl et al. "A 200-MHz 64-b dual-issue CMOS microprocessor", IEEE Journal of Solid-State Circuits, vol. 27, no. 11, pp. 1555-1567, Nov. 1992.

[13] N.Yamashita et al. "A 3.84 GIPS Integrated Memory Array Processor with 64 Processing Elements and a 2-Mb SRAM", IEEE Journal of Solid-State Circuits, vol. 29, no. 11, pp. 1336-1343, Nov. 1994.

[14] E. R. Davies, "Machine Vision: Theory, Algorithms, Practicalities", Academic Press Limited, London, 1990