SWITCHED-CURRENT TECHNIQUES FOR IMAGE PROCESSING CELLULAR NEURAL NETWORKS IN MOS VLSI

S. Espejo, A. Rodríguez-Vázquez, R. Domínguez-Castro and
J.L. Huertas.

Dept. of Design of Analog Circuits.
Centro Nacional de Microelectrónica
Edificio CICA, Avda. Reina Mercedes s/n, 41012-Sevilla, Spain.

Abstract - An architecture and related building blocks are presented for the realization of image processing tasks using current-mode analog-digital circuits. The architecture is based on the Cellular Neural Network paradigm while implementation is made using switched-current circuit techniques. Since just MOS transistors are required as circuit primitives, the proposed circuits are well suited for standard digital CMOS technologies. Also, the sampled-data nature of switched-current techniques allows for easy incorporation of programmability and reconfigurability issues. Empirical results are given for 1.5μm N-well double-metal CMOS prototypes.

I. INTRODUCTION

In the last few years, *current-mode* circuits have shown a great potential and applications in analog signal processing circuits, traditionally implemented on the basis of voltage-represented variables [1]. Some important potential advantages of current-mode techniques are *increased bandwidth* and *dynamic range*, specially in reduced voltage supply environments. Additionally, operation speed can generally be improved and, many times, area and power consumption can be decreased.

Other significant performance improvements in analog MOS signal processors were achieved by the widespread use of *sampled-data* techniques, instead of the traditional continuous-time mode. In fact, the introduction of switched-capacitor (SC) circuits paved the way for the monolithic implementation of analog systems in VLSI [2]. This technique provided higher yield, higher dynamic range, ease of controllability and programmability, and better stability and robustness trough parasitic-free operation.

The combination of sampled-data and current-mode techniques opened a new design style usually known as the Switched-Current (SI) approach [1,3]. Together with the potential advantages derived from the fact of being current-mode, the main advantages of SI techniques as compared to SC is that for SI the only components required are just MOS transistors. Hence, SI circuits are readily compatible with *standard digital CMOS* processes, where linear capacitors are not available.

Switched-current techniques have been explored in different application contexts [1]. In this communication we explore the use of these techniques for *image processing* MOS VLSI chips based on the *cellular neural network* paradigm [4,5]. Cellular Neural Networks (CNN), which are closely related to cellular automata, consist of arrays of basic computational units, each one connected only to its closest neighbors in the net. This local connectivity property is very convenient to simplify the routing in VLSI implementations. CNNs show potential applications in areas like image processing and pattern recognition [6,7,8] which combined to the local connectivity feature motivates research in the implementation of this kind of architectures.

The suitability of the SI approach is clear in this context, taking into account that only CMOS transistors are required and hence eliminating the need of expensive and/or inaccurate devices like linear capacitors and resistors reflected in previous implementation proposals [5, 9].

II. SAMPLED DATA CNNs

A CNN consists of a bidimensional distribution of elementary processing units *(cells)* each of them being connected to a limited number of neighbors. The interconnection scheme is identical for every cell in the array (with the obvious exceptions of the ones in the borders). For any arbitrary inner cell the connected neighbors define the so-called cell *neighborhood*, $N_r(i,j)=\{C_{kl}, |i-k|<=r \text{ and } |j-l|<=r\}$ where r is the neighborhood radius. Three variables are defined for each cell: the state x_{ij}, the input u_{ij}, and the output y_{ij}. The output of each cell is related to the corresponding state by the following nonlinear operator, depicted in Fig.1,

$$y_{ij} = \frac{1}{2}(|x_{ij}+1| - |x_{ij}-1|) \tag{1}$$

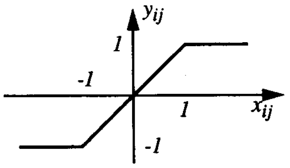

Figure 1: nonlinear operator.

In the original CNN model [4,5], the dynamic behavior of the CNN is governed by a system of first order nonlinear differential equations, one per cell:

0-7803-0593-0/92 $3.00 1992 IEEE

$$\frac{dx_{ij}}{dt} = \frac{1}{\tau}\{-x_{ij}+D_{ij}+ \\ + \sum_{C_{kl}\in N_r(i,j)} [A(i,j;k,l)y_{kl}+B(i,j;k,l)u_{kl}]\} \tag{2}$$

where *A(i,j;k,l)* and *B(i,j;k,l)* are *(2r+1)*^{2} matrices respectively called the network *feedback* and *control templates*. These templates, together with the offset terms D_{ij}, control the steady-state input/output operation of the network. On the other hand, parameter τ acts as a time scaling factor, controlling the speed of the network.

For sampled-data implementations, equation (2) must be substituted by a system of first order nonlinear finite-differences equations, via some discrete-time integration algorithm. Implementation consideration takes us to choose the Forward-Euler algorithm, resulting in:.

$$x_{ij}(n+1) = x_{ij}(n) + \frac{T}{\tau}\{-x_{ij}(n)+D_{ij}+ \\ + \sum_{C_{kl}\in N_r(i,j)} [A(i,j;k,l)y_{kl}(n)+B(i,j;k,l)u_{kl}]\} \tag{3}$$

where *T* is the clock period.

This system emulates the dynamic behavior of its continuous-time counterpart, the emulation accuracy increasing as T/τ decreases.

In practical applications we are not however interested in the accurate emulation of the transient behavior of the continuous-time model but in ensuring that both models exhibit the same equilibria. Hence the trade-off in choosing the value of T/τ does not concern accuracy but stability and operation speed. On the one hand, in order to increase the operation speed T/τ must be chosen as large as possible. On the other T/τ must be smaller than 2 due to stability considerations.

It is important to point out that making $T/\tau < 2$ does not however ensure asymptotic stability; in other words, the system could reach an oscillatory steady state or enter in a chaotic regime. This is less likely to happen if T/τ is well below 2. We have made a huge number of numerical simulations for different templates and initial states using different values of T/τ in the interval (0,2). Our results show that choosing $T=\tau$ leads to constant steady states in practical situations. This particular choice has also advantages when it comes to implementation, as it can be seen from equation (4), where equation (3) has been rewritten with the assumption $T=\tau$.

$$x_{ij}(n+1) = D_{ij}+ \\ + \sum_{C_{kl}\in N_r(i,j)} [A(i,j;k,l)y_{kl}(n)+B(i,j;k,l)u_{kl} \tag{4}$$

Note that the two summands containing the contribution of $x_{ij}(n)$ to the future state $x_{ij}(n+1)$ have been cancelled. This reduces the complexity of the implementation. Furthermore, since future state values do not depend on present state values, it is not necessary to store the state variables, but only the outputs.This means that the maximum value for any variable in the network is the saturation limit, and hence that the dynamic range is optimum:

$$y_{ij}(n+1) = f(D_{ij}+ \\ + \sum_{C_{kl}\in N_r(i,j)} [A(i,j;k,l)y_{kl}(n)+B(i,j;k,l)u_{kl}]) \tag{5}$$

where *f(.)* is the nonlinear operator in (1).

III. SWITCHED-CURRENT IMPLEMENTATION

From (5), we can see that the analog operations required to implement a CNN cell are *saturation nonlinearities*, *delays*, *summation*, and *weighted replication*. In addition, *current memories* are required if inputs u_{ij} are stored internally. Otherwise, in case the inputs were externally applied, the number of bonding pads would probably be extremely large. A block diagram of a cell C_{ij} is shown in Fig.2. It is assumed that summation is performed via KCL at the input node. Programmable weighted replicators at the output of the cell produce weighted replicas of y_{ij}, one for each cell in the neighborhood. Thus, input-weightings for cell C_{ij} are actually performed at the output of cells C_{kl} in the neighborhood.The nonlinear operator is the first stage in the cell, after the KCL summation at the input node, as in (5)

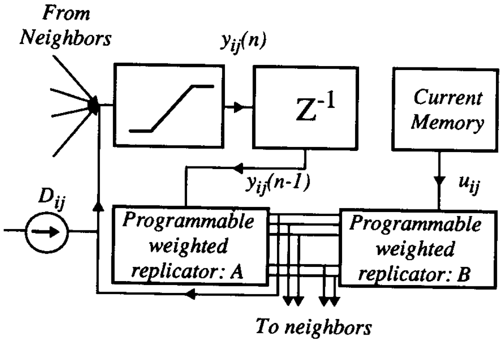

Figure 2: block diagram of cell C_{ij}.

Figures 3a,b,c,d show schematic diagrams for the nonlinear operator, delay, current memory and weighted replicator respectively. Programmability of the replicator is achieved by multiplexing current paths with analog switches.

Some of the blocks in Fig.3 can be simplified using complementary current mirrors and sources. However, this would require the duplication of the biassing circuitry, force the use of switches at both p-mos and n-mos gates, and produce a larger DC current offset due to appreciable V_{ds} variation among matched transistors.

Building-blocks in Fig.3 use simple current mirrors. However, higher performance implementations are readily available if needed [10,11]. In particular, if the output to input impedance ratio is critical, *cascode* implementations may actually reduce the area required, as they allow the use of moderate-length devices.

Just to illustrate the practical operation of the proposed building blocks, Fig.4 shows the input-output characteristics measured from a 1.5μm CMOS prototype of the CNN nonlinear block. In a similar way, Fig.5 shows the signal measured at the output of a 1.5μm CMOS prototype of a lossy integrator for a square input signal.

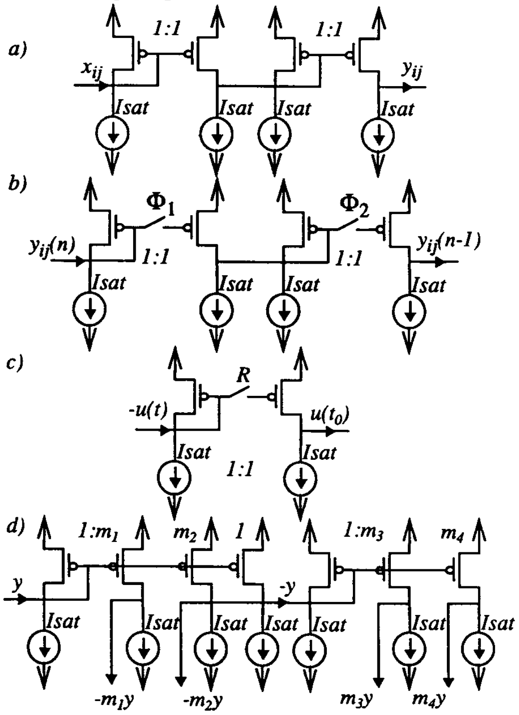

Figure 3: schematics of basic building blocks.

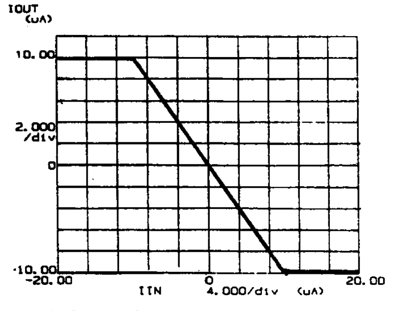

Figure 4: 1.5μm CMOS nonlinear operator response.

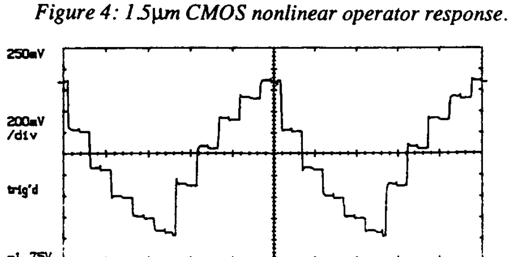

Figure 5: 1.5μm CMOS lossy integrator response.

IV. NETWORK LAYOUT

With regard to the layout, when a cell is designed to be used in a network implementation, it is important to anticipate the connectivity to the neighbor cells, in order to avoid cumbersome and error-prone repetitive routing when building the network. Fig. 6 depicts details of the layout of an entire network when a neighborhood of radius one is considered The floor-planning of the network resembles that of a digital RAM memory, except for the fact of local interconnections. Signals I/O and U/X in Fig 6 are used to configure the cell for the different tasks of loading, processing and unloading an image. PC bus is used to select the particular templates and offset term to be used, while CS and IO busses are the equivalent to the address and data buses of digital memories.

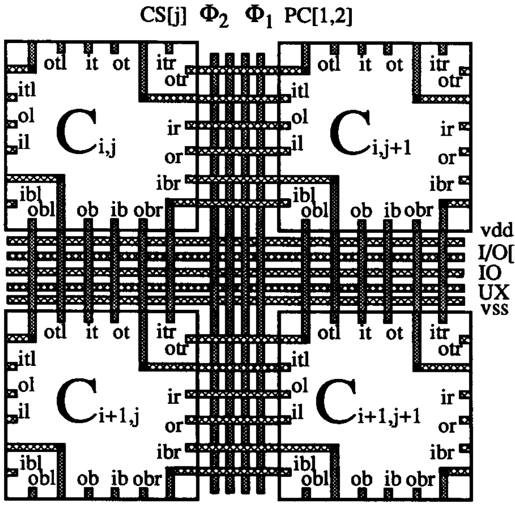

Figure 6: network floor-planning.

Current reference for each cell can be generated on the periphery of the network and distributed to each cell. However this requires a large number of bias lines, one per cell. Other options are to generate one reference current for each small group of cells, replicating this current locally, or making each cell generate its own internal current reference. A compromise must be assumed between area dedicated to bias routing and statistical dispersion of references.

Another issue that is worth mentioning is the fact that for each cell C_{ij}, the input component given by the product of the control template B and the inputs u_{kl} to the cells in the neighborhood, is constant during the evolution. Hence, instead of storing the input u_{ij} in the analog memory of each cell, the whole constant term in the dynamic equation could be stored, eliminating the need to implement the control template. This however requires external precalculation of the terms to be stored. Furthermore, in many application cases, it is common to make the input matrix u_{ij} equal to the initial image $x_{ij}(0)$. In this cases, if the control template is physically implemented, both the initial image and the input matrix could be written simultaneously, while different storing sequences would be required otherwise. The decision of wether to use one or the other approach must be based on the type of tasks required to the network and the time available for writing each image.

V. EMPIRICAL RESULTS

As mentioned in a previous section, we have performed a huge number of numerical simulations over networks with different sizes, different templates and different inputs and initial states. However, when it comes to electrical simulation, even small networks require large amounts of memory and CPU times. Nevertheless, the following figures depict some electrical-simulation results. Fig. 7 shows results of a 9 x 9 network for noise removal, simulated with MOS level 2 description of the whole cell (except the biassing circuitry) for a 1.5μm CMOS technology. Numerical simulation results are attached for comparison. Noise to signal power ratio was 1/3. Fig. 8 shows results of a 16 x 16 network for noise removal, edges extraction and corners extraction. These results were obtained with a mixed description of the cell, maintaining the MOS description of the delays and nonlinear operators and macromodeling the replicators.

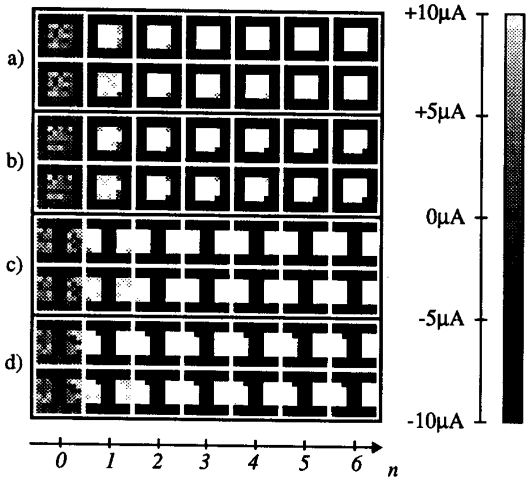

Figure 7: Four different electrical (bottom) and numerical (top) simulation results for noise removal.

VI. CONCLUSIONS

We have demonstrated the applicability of switched current techniques for the implementation of Cellular Neural Networks. In particular, several image processing tasks have been simulated at the device level, yielding high concordance with numerical simulations. Basic building blocks for the proposed architecture have been fabricated on a 1.5um N-well double-metal CMOS digital technology. Experimental results of these building blocks corroborate the validity of SI techniques for the proposed application. We are currently working on the design and test of whole-network prototypes.

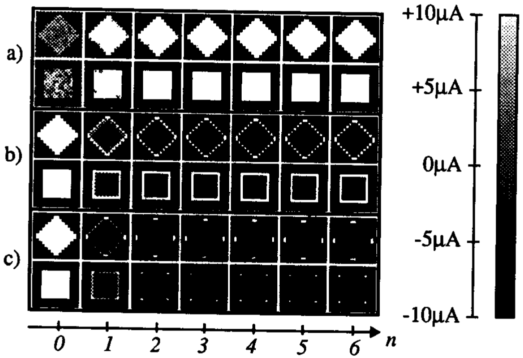

Figure 8: Electrical simulation results of 16 x 16 networks for noise removal, edges extraction and corners extraction.

VII. REFERENCES

[1] C. Toumazou et al (editors): *"Analog IC Design: The Current-Mode Approach"*. Peter Peregrinus 1990.

[2] K. Nakayama and Y. Kuraishi: "Present and Future Applications of Switched-Capacitor Circuits". *IEEE Circuits and Devices Magazine*, Vol.3, pp 10-21, September 1987.

[3] J.B. Hughes et al.: "Switched-Currents. A New Technique for Analog Sampled-Data Signal Processing". *Proc. IEEE ISCAS1989*, pp 1584-1587, May 1989.

[4] L.O. Chua and L. Yang: "Cellular Neural Networks: Theory". *IEEE Trans. Circuits and Systems*, Vol.35, pp 1257-1272, October 1988.

[5] L.O. Chua and L. Yang: "Cellular Neural Networks: Applications". *IEEE Trans. Circuits and Systems*, Vol.35, pp 1273-1290, October 1988.

[6] T. Matsumoto et al.: "CNN Cloning Template: Connected Component Detector". *IEEE Trans. Circuits and Systems*, Vol.37, pp 633-635, 1990.

[7] T. Matsumoto et al.: "CNN Cloning Template: Hole-Filler". *IEEE Trans. Circuits and Systems*, Vol.37, pp 635-638, 1990.

[8] T. Matsumoto et al.: "CNN Cloning Template: Shadow Detector". *IEEE Trans. Circuits and Systems*, Vol.37, pp 1070-1073, 1990.

[9] L. Yang et al.: "VLSI Implementations of Cellular Neural Networks". *Proc. IEEE ISCAS1990*, pp 2425-2427, 1990.

[10] Z. Wang: "Analytical Determination of Output Resistance and DC Matching Errors in MOS Current Mirrors". *IEEE Proceedings*, Vol.137 Pt. 6, pp 397-404, October 1990.

[11] T.S. Fiez et al.: "Switched-Current Design Issues". *IEEE J. Solid-State Circuits*, Vol.26, pp 192-201, March 1991.