Realization of a CNN Universal Chip in CMOS Technology

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

Centro Nacional de Microelectrónica. Analog Design Department -Universidad de Sevilla
Edificio CICA, C/ Tarfia sn, 41012-Sevilla, SPAIN
Phone # 34 5 4623811, FAX # 34 5 4624506, email espejo@cnm.us.es

Abstract

This paper describes the design of a programmable Cellular Neural Network (CNN) chip, with additional functionalities similar to those of the CNN Universal Chip. The prototype, which contains 1024 cells, has been designed in a 1.0μm, n-well CMOS technology. A careful selection of the topology and design parameters has resulted in a cell density of 31 cells/mm^2 and an accuracy in the weight values in the range of 7-8 bits. Adaptive techniques have been employed to ensure accurate external control and system robustness against process-parameter variations.

I. Introduction

Massively-parallel analog-processing systems are natural candidates in those application fields in which nature has demonstrated their outstanding capabilities [1]. The processing front-end of biological vision systems (retina) has inspired the development of highly-parallel computation algorithms which represent a potential breakthrough in artificial vision applications [2]. However, the practical use of these algorithms is conditioned to their efficient and feasible physical implementation.

In this context, the cellular neural network (CNN) paradigm [3], [4] represents an interesting alternative, with a demonstrated wide range of applications [5] and particularly well suited for monolithic IC realizations.

This contribution is centered around the design of a CNN Universal Chip [6] with optical input capability in a standard 1.0μm, n-well CMOS technology, and describes the major trends, obstacles and the solutions adopted after a careful analysis of many alternatives. The prototype is due from the foundry in the next weeks. Experimental results will be available at the conference.

II. The CNN computation paradigm

A CNN is an array of *locally* interconnected processing units (*cells*), arranged on a two-dimensional square grid. Each individual cell i has three associated variables: the *state* $x^i(t)$, which conveys cell energy information as a function of time; the *input* u^i, representing external excitation; and the *output* $y^i(t)$, related to the cell state via a nonlinear double-saturation characteristic:

$$y^i = f(x^i) \equiv \frac{1}{2}\left(|x^i + 1| - |x^i - 1|\right) \tag{1}$$

Every cell i interacts with other cells located within a distance 1 in the grid. These cells constitute the (radius 1) *neighborhood* of the cell $N(i)$, which includes cell i itself. The dynamic behavior of the network is governed by a set of nonlinear differential equations, one for each cell,

$$\tau\frac{dx^i}{dt} = -g(x^i) + d^i + \sum_{j \in N(i)} \{a^i_j y^j + b^i_j u^j\} \tag{2}$$

Coefficients b^i_j and a^i_j are called *control* and *feedback* weights, respectively, and d^i is the *offset* parameter. These parameters determine the input-output mapping performed by the net. In the common case of *uniform* CNNs, coefficients a^i_j, b^i_j, and d^i are invariant with i, and are commonly defined in the form of *templates* [3].

In the full signal range (FSR) model [7] employed in our prototype, function $g(.)$ is given by,

$$g(x) = \lim_{m \to \infty} \begin{cases} m(x+1) - 1 & ; x < -1 \\ x & ; |x| \le 1 \\ m(x-1) + 1 & ; x > 1 \end{cases} \tag{3}$$

as opposed to the original proposal in which $g(x) = x$. The adoption of this model simplifies the circuitry required for the implementation of the basic cell, and increases the robustness of the hardware [7].

0-7803-2570-2/95 $4.00

III. A CT Full Signal Range CNN Universal Chip

A. Functionality and characteristics

The functionality of the chip is similar to that of a CNN Universal Chip [6]. Its central capability is that of a completely programmable CNN: controllable feedback, control and offset coefficients. In addition, every cell is equipped with a programmable digital gate, digital memory, and other I/O and control circuitry, as illustrated in Fig. 1

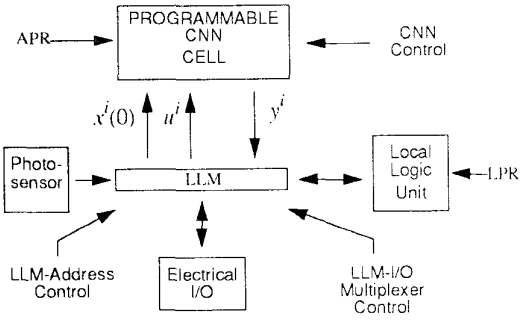

Fig. 1: Schematic Architecture of one Cell.

The CNN network is constructed on a rectangular grid, it is assumed to be uniform, and its neighborhood radius is always one. In the implementation of the cells, a strong emphasis has been placed on low area and power consumption, as well as on accuracy. Extensive analysis has been performed to optimize the architecture and basic building blocks against statistical *on-die* parameter variations (mismatch). A strong effort is dedicated to obtain robustness against *wafer-level* parameter variations. For this purpose, many of the internal variables are automatically tuned on chip.

The final topology forecasts weight and offset accuracies in the range of 7-8 bits, while area requirements allow a complete 32×32 cells system to be integrated in a $7.7\text{mm} \times 6.8\text{mm}$ prototype using a $1.0\mu\text{m}$, n-well, digital technology. In particular cell area is about $171\mu\text{m} \times 187\mu\text{m}$. Array dimensions are $5.5\text{mm} \times 6.0\text{mm}$. The remaining area is dedicated to boundary cells, weight adaptation stages, memories for template coefficients and local-logic-unit truth tables, adaptive bias stages, I/O circuitry and bonding pads.

Even though the processing circuitry is analog, chip management is completely digital, facilitating its control and communications. No external analog signals are required, nor for chip control, neither as references. Analog weights are specified and stored in digital form. For each template value, an internal adaptive stage transforms the digital code to an analogue voltage, which is then transmitted to the network. This methodology results in weight independence from particular process parameter values, as well as an accurate external control. The quantization error on the coefficients is lower than the expected statistical error of the analog multipliers. Hence, it is not relevant.

Every cell incorporates a photosensitive device, which allows the system to be optically initialized. These devices are CMOS compatible, and incorporate a tuning scheme for automatic adaptation to different illumination conditions [8]. Electrical initialization is possible as well, while output image is always downloaded in electrical form. Input and output images are assumed to be binary in every case. Electrical image uploading and downloading is realized through 32 I/O bonding pads, on a row by row schedule.

The digital circuitry at each cell includes a four-bits static memory (LLM), a completely programmable two-input digital gate (LLU), and initialization and control circuitry for many different operations. The four-bit memory at each cell allows the network to store four complete images. Two additional memories with fixed +1 and −1 values are also available.

Local information transference is centralized around the LLM, as shown in Fig. 1. Images to be processed, downloaded, or used as LLU input are always taken from the local memory. In the same manner, images obtained from a CNN or LLU operation, from the photosensors, or electrically uploaded, must be stored in one of the four LLM locations.

The program-control functionality is based on a large static *digital* memory, located at the periphery of the cell array. This memory is used to store up to *eight* complete *sets* of coefficients. Each set specifies all of the parameters required to define the CNN operation (APR) and the truth-table of the programmable digital gate (LPR). In particular, the following CNN parameters are stored in every set: feedback template (nine values), control template (nine values), offset term, and boundary conditions. These boundary conditions represent the time-independent state-variable and input value of the external neighbors of the cells at the border of the array. All analog values are codified by 8-bit words (7 plus sign). Weight ranges are [-4,+4], and offset term range is [-8,+8].

After the internal memory has been loaded, the eight sets of coefficients can be used in any order, any number of times. Making an analogy with digital microprocessors, each set of coefficients can be viewed as the definition of a microinstruction.

IV. Cell Circuitry

A. Multiplier selection

Programmable scaling blocks or *multipliers* are the key block in a programmable CNN cell, due to the large number of them required (one per coefficient).

We have selected the multiplier shown in Fig. 2 [9]. This structure is a fully differential, four-quadrants multiplier with high linearity, and presents an excellent area/accuracy figure. The four transistors are identical, and operate in the triode (ohmic) region. Nodes *Iop* and *Ion* are connected to the differential input of the cells in the neighborhood. The state variable of the cell is represented, in differential form, by signals V_{xp} and V_{xn}. The differential weight signal is compound of signals V_{pp} and V_{pn}. The scaled signal, in current form, is differential as well, and given by the difference of the currents flowing out of the multiplier from nodes I_{op} and I_{on}. Analysis of the structure in Fig. 2 yields

$$I_{op} - I_{on} = 2\beta (V_{pp} - V_{pn}) (V_{xp} - V_{xn}) \qquad (4)$$

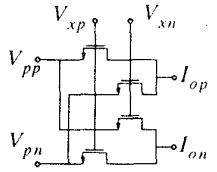

Fig. 2: Selected Analog Four-Quadrants Multiplier.

Note that all transistors are n-channel, resulting in high area efficiency (no wells required). Both input signals (the state variable and the weight) are given in voltage form, while output is represented by a current. These characteristics facilitate the following three operations: a) the distribution of the state variable of each cell to the multipliers used to implement the different coefficients (one per neighbor), b) the distribution of the programmed coefficient values to every cell in the array (weights are invariant from cell to cell, since the network is uniform), and c) the summation of the contributions coming into every cell from the different cells in the neighborhood.

B. Cell architecture and operating mode

Fig. 3 shows the analog core of the cell, excluding the multipliers. The architecture is fully differential. Contributions from all neighbors are added at the low impedance nodes I_{ip} and I_{in}. This is accomplished by connecting the output nodes I_{op} and I_{on} of the corresponding multipliers in the neighboring cells to nodes I_{ip} and I_{in}, respectively. Correspondingly, multipliers implementing the scaled replicas of the state variable of this cell, with output directed towards the cells in the neighborhood, have their input nodes V_{xp} and V_{xn} connected to the nodes with the same name in Fig. 3, whose voltages represent the differential state variable of the cell.

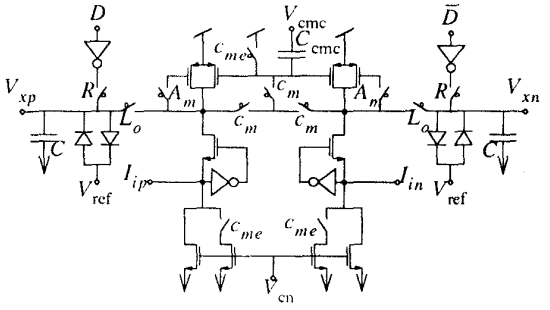

Fig. 3: Schematic of the Analog Core of the Cell.

Switches labeled L_o are used to set the cell in open or closed-loop configuration. That is, while signal L_o is high (switches are *off*), no dynamic evolution occurs.

Diodes in the figure are actually realized by a p-n-p vertical device and a diode-connected n-channel transistor. Integrating capacitors C at nodes V_{xp} and V_{xn} are realized by parasitic capacitors, corresponding to the sum of the gate capacitance of the multipliers.

Only nine analog multipliers (not shown in Fig. 3) are included in each cell. In a first step, control contributions are generated by connecting the multipliers input nodes to a pair of voltage levels representing the input value u^j of the cell, and the weight signals to the values corresponding to the control template. In this manner, each cell receives the sum of the control contributions from its neighborhood, and stores it in an analog memory by turning off switches labeled A_m in Fig. 3. In a second step, multipliers are used to generate the feedback contributions. Weight signals are set to the values corresponding to the feedback template, initial state values $x^j(0)$ are introduced in the integrating capacitors, and the network is allowed to perform its dynamic evolution. This strategy halves the number of multipliers in every cell. Furthermore, the input-stage offset is stored together with the control contributions, and cancelled during the dynamic evolution.

V. System architecture and control

A. Architecture

System architecture is schematically represented in Fig. 4. As mentioned earlier, system operation relays on

a number of adaptive stages, used to tune electrical variables, to compensate inaccuracies, and for automatic weight adjustment.

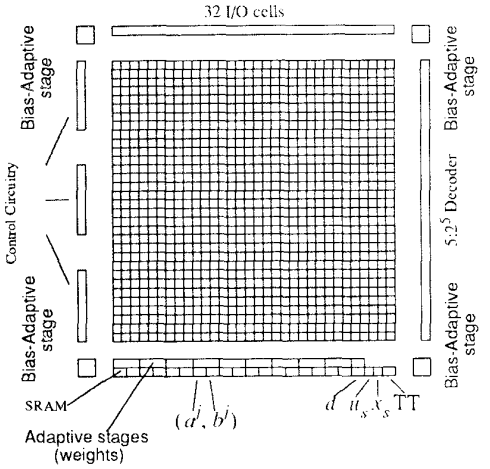

Fig. 4: Schematic System Architecture.

A digital decoder, placed at the right side of the cell array, is used to generate the 32 signals $R(i)$, required for the row by row I/O protocol.

The 32 I/O cells located at the top of the chip include input and output digital buffers, as well as the circuitry required to multiplex the *input* and *output* signals through the same 32 lines $EIO(j)$.

The circuitry located at the bottom of the cell array can be divided into two large sections. The first of them, located below, consists of a set of 20 SRAM blocks, each with eight words of eight bits. The second one, located above, contains 10 adaptive stages.

Of the 10 adaptive stages, nine of them are identical, and tune the weight voltages of the feedback *or* control templates. Remind that the feedback and control template are not used simultaneously by the network. The input to the adaptive stages are connected to the corresponding feedback or control coefficient in the SRAM by complementary global signals A and B. The last adaptive stage, slightly different to the others, tunes the value of the offset term.

B. Weight control

The use of either analog or digital programmability presents advantages and drawbacks. Even for a reduced number of bits, digitally programmable multipliers require areas much larger than that of common analog multipliers. In addition, a large number of control lines is required, which results, in general, in the dominant area requirement. On the other hand, analog multipliers are sensitive to process parameter variations, difficulting the accurate setting of the coefficients. In addition, the on-chip storage of the analog weight values requires analog memories, which, in general, present time-degradation problems.

In this design, we have used a hybrid approach, based on the use of analog-programmable multipliers within the cells, and digital control from the exterior of the network. This combines the advantages of analog and digital programmability [10].

The analog weight signal is generated from the digital word using an adaptive loop, which involves a linear D/A converter and an analog multiplier identical to those used within the cells. The adaptive control eliminates the dependency of the weight values on process parameters.

VI. References

[1] J.M. Zurada: *"Introduction to Artificial Neural Systems"*. West Publishing Co., 1992.

[2] Gupta, M.M. and Knopf, G.K. (Ed.): "Neuro-Vision Systems, Principles and Applications". IEEE Press, 1994.

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

[4] L.O. Chua and T. Roska: "The CNN Paradigm". *IEEE Trans. Circuits and Systems I: Fundamental Theory and Applications*, Vol. 40, pp 147-156, March 1993.

[5] Proceedings of the Second IEEE International Workshop on Cellular Neural Networks and their Applications, CNNA-92. Munich, Germany, 14-16 December 1992.

[6] T. Roska and L.O. Chua: "The CNN Universal Machine: An Analogic Array Computer". *IEEE Trans. Circuits and Systems II: Analog and Digital Signal Processing*, Vol. 40, pp 163-173, March 1993.

[7] S. Espejo, A. Rodríguez-Vázquez, R. Domínguez-Castro, B. Linares and J.L. Huertas: "A Model for VLSI Implementation of CNN Image Processing Chips Using Current-Mode Techniques", *1993 IEEE Int. Symp. on Circuits and Systems*, pp 970-973, 1993.

[8] S. Espejo, A. Rodríguez-Vázquez, R. Domínguez-Castro, J.L. Huertas and E. Sánchez-Sinencio: "Smart-Pixel Cellular Neural Networks in Analog Current-Mode CMOS Technology". *IEEE J. of Solid-State Circuits*, Vol. 29, pp 895-905, August 1994.

[9] N.I. Khachab and M. Ismail: "Linearization Techniques for nth-Order Sensor Models in MOS VLSI Technology". *IEEE Trans. Circuits and Systems*, Vol. 38, pp 1439-1449, December 1991.

[10] S. Espejo: *"VLSI Design and Modeling of CNNs"*. Ph. Dissertation, University of Sevilla, March 1994.