CNNA-94 Third IEEE International Workshop on Cellular Neural Networks and their Applications
Rome, Italy, December 18-21,1994

A CNN Universal Chip in CMOS Technology

R. Domínguez-Castro, S. Espejo, A. Rodríguez-Vázquez, and R. Carmona
Centro Nacional de Microelectrónica-Universidad de Sevilla
Edificio CICA, C/Tarfia s/n, 41012-Sevilla, SPAIN
Phone: 34 - 5 - 423 99 23. Fax: 34 - 5 - 462 45 06. E-mail: rafael@cnm.us.es

***Abstract** - This paper describes the design of a CNN universal chip in a standard CMOS technology. The core of the chip consists of an array of 32 x 32 completely programmable CNN cells. Input image can be loaded in optical or electrical form. Accuracy is in the range of 7-8 bit, and cell density is of 33 cells/mm^2.*

1. Introduction

CNN Universal Chips are the main components of CNN Universal Machines [1]. Their universality [2], together with their ability to implement any CNN application, makes their electronic implementation extremely attractive.

Fig. 1 shows the conceptual block diagram of the implementation described in this paper. The system has the same fundamental capabilities as the original version proposed by Roska and Chua [1], along with the possibility of optical initialization.

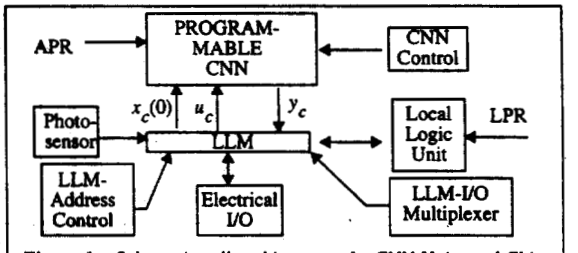

***Figure 1** - Schematic cell architecture of a CNN Universal Chip.*

During the system design process, special attention was dedicated to the obtention of precise static and dynamic operators. For this reason, a modified version of the original CNN model has been used [3]. This model has properties which are very similar to those of the original one, results in higher area and power efficiency, and is more tolerant to process parameter variations. It can be described by the following equation,

$$\tau\frac{dx^c(t)}{dt} = \begin{cases} \sum\{a_d^c x^d(t) + b_d^c u^d\} + k^c x_{sat} & , \forall|x^c| < x_{sat}^c \\ 0 & , \forall|x^c| = x_{sat}^c \end{cases} \quad (1)$$

where the sum extends to the neighborhood of cell c, and every symbol is used with its traditional meaning [3] except for the term $k^c x_{sat}$, which here represents a programmable (by factor k^c) offset term.

2. A synergy of analog and digital programmability

A hybrid strategy which combines the advantages of analog and digital programmability [4] has been used to control the programmed values of CNN coefficients. This approach is based

on a combined use of analog-programmable multipliers within the cells and of digital control signals from the outside of the cell array. Clearly, some interface circuitry is needed to generate the internal analog weight-signals from their digitally coded values. Due to the large number of multipliers in the network, even small reductions in multiplier-area justify the area dedicated to the interface circuitry. In our case, the cell-area is substantially reduced, thus system-area reduction is extremely high.

The interface circuitry consists of several identical blocks, one for each programmable parameter in the network. The functionality of each interface blocks is that of a nonlinear D/A converter, and its implementation follows the adaptive architecture shown in Fig. 2, in which the analog weight signal adapts its value until the scaling factor of the analog- and digitally-controlled multipliers coincide [4].

This approach merges most of the advantages of digitally- and analog-programmed multipliers, achieving low areas and a reduced number of control lines, simplifying the control of the weight values, and eliminating their sensitivity to global process parameter variations. In addition, it allows the realization of the APR [1] using a digital RAM memory.

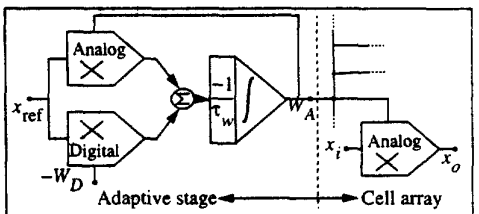

Figure 2 - Architecture of the D/A interface circuitry required for the hybrid-control approach.

The electrical implementation of one adaptive stage is schematically illustrated in Fig. 3, and can be briefly described as follows. Two low impedance loading stages, identical to those used in the cells, are driven by an analog-programmed multiplier, also identical to those used in the cells. The differential output current of the multipliers is converted to single-ended by a p-channel current mirror. The resulting single-ended current is substracted from the current generated by a D/A converter (the digitally-programmed multiplier) with current-form output. The current generated by the D/A converter is $2pI_{sat}$, where p is the digitally programmed weight, in the range $[0, W_{max}]$, and I_{sat} is a reference current related to the voltage saturation level of the state variables. The input signals V_{rp} and V_{rn} of the analog-programmed multiplier are driven by this voltage saturation levels.

The weight signals of the analog-programmed multiplier in the adaptive stage are driven

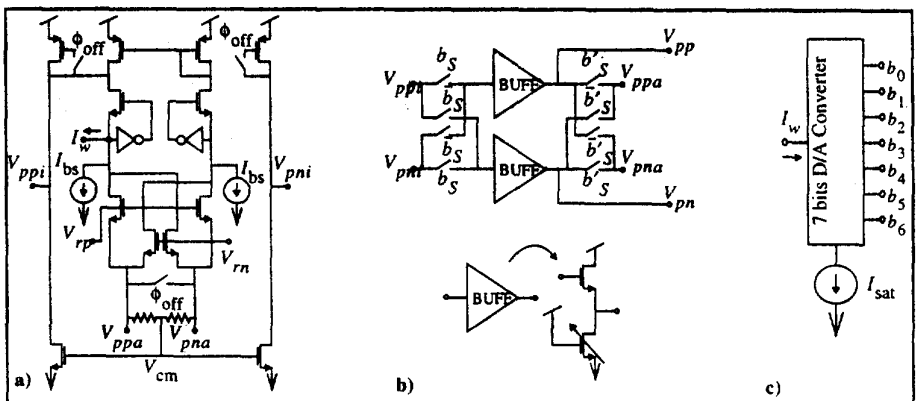

Figure 3 - Architecture of a weight adaptive stage. a) Low impedance loading stage. b) Very low output impedance buffer. c) 7 bits D/A converter with current output.

by two analog buffers with very low output impedance, whose output are also the weight signals transmitted to every cell in the system. During a precalibration step, controlled by ϕ_{off}, these buffers are disconnected from the multiplier, and the differential weight signals are shorted at the input of the multiplier. This has the effect of setting the common mode voltage V_{cm} to the voltage V_L at the input of the low impedance loading stages. The error current flowing out of the p-channel current mirror is stored in the p-channel transistors with gates driven by switches controlled by ϕ_{off}, which constitute the analog memories. After this calibration step, the output of the current memories are added to the difference of the output signals of the analog multiplier and the D/A converter, and the resulting current is integrated at the input nodes of the analog buffers. The output of these buffers control the weight signals of the multiplier, thus closing a feedback loop which settles after the analog weight signals have the correct, adapted value. The sign of the programmed weight is introduced by swapping the connections of the input and output nodes of the buffers, with respect to the adaptive core of the circuitry. The signals driving the cells do not have any switch in their path, in order to avoid output impedance degradation. The D/A converter is implemented by a binary weighted array of n-channel transistors in saturation.

A final important issue is the power dissipation of the buffers, which is extremely high. The amount of current required from the output of one buffer depends on the value of the programmed weight. For this reason, the buffers include a weight-dependent bias current, controlled by the most significative bits of the digital word encoding the weight value.

3. Error Sources and Evaluation Criteria

The design of area-efficient analog integrated circuits requires a careful analysis directed towards the identification, characterization, and reduction of all possible errors arising in the practical realization. A first subdivision of these errors can be into dynamic and static non-idealities. In addition, errors can be classified as deterministic and random. Deterministic errors are originated by any non ideality derived from the circuit implementation assuming that identically designed devices are actually identical, while random errors are originated by mismatch effects.

Mismatch errors are strongly dependent on the area of the devices, which is of extreme relevance in our application. Mismatch errors are the dominant error source whenever low area devices are used, and for CNN implementations, these errors are specially relevant on the multipliers.

3.1.Mismatch errors on the multipliers

We assume that every multiplier has a weight deviation and an output offset. That is, while the ideal characteristic of the multiplier with a generic weight p is, $x_o = px_i$, in practice we have $x_o = (p+\delta p)x_i + \delta x_{\text{off}}$, where both δp (weight error) and δx_{off} (offset error) are stochastic variables, assumed statistically independent, with zero mean. Using this model for all the multipliers in Eq. 1, and considering the cell to be in its linear region, we have,

$$\tau\frac{dx^c(t)}{dt} = \sum\{a^c_d x^d(t) + b^c_d u^d\} + k^c x_{\text{sat}} + \\ + \sum\{\delta a^c_d x^d(t) + \delta b^c_d u^d\} + \sum\{\delta x^d_{\text{off}} + \delta u^d_{\text{off}}\} + \delta k^c x_{\text{sat}} + \delta x^c_{\text{sat}} \tag{2}$$

where the first line of the equation contains the nominal terms, and the second the error terms.

The right-hand side of equation Eq. 2 constitutes the integrand $I^c(t)$ of the integral equation governing every cell (see Eq. 1). Assuming that the state variables and the input values are

limited to the same interval $[-x_{sat}, x_{sat}]$, the maximum nominal value of $I^c(t)$ is given, at any time instant by,

$$\max\{I^c(t)\} = x_{sat} \cdot [\sum \{|a_d^c| + |b_d^c|\} + |k^c|] \tag{3}$$

If for every multiplier, the *relative weight error* and the *relative offset error* are bounded by

$$\left|\frac{\delta p}{p}\right| \le \frac{\varepsilon}{2} \qquad \left|\frac{\delta x_{off}}{p x_{sat}}\right| \le \frac{\varepsilon}{2} \tag{4}$$

then, the error of integrand $I^c(t)$ relative to its maximum value is bounded by ε,

$$|E| = \left|\delta k^c x_{sat} + \delta x_{sat}^c + \sum_{d \in N(c)} \{\delta a_d^c x^d(t) + \delta b_d^c u^d + \delta x_{off}^d + \delta u_{off}^d\}\right| \le$$
$$= \varepsilon \cdot x_{sat}\left[|k^c| + \sum_{d \in N(c)} \{|a_d^c| + |b_d^c|\}\right] = \varepsilon \cdot \max\{I^c(t)\} \tag{5}$$

After a detailed analysis of a large number of multipliers [5], a multiplier based on four MOS transistors operating in the triode region (Fig. 4b) was selected. This multiplier exhibits low mismatch errors and a wide range of linearity. In addition, its parasitic dynamic behavior is negligible.

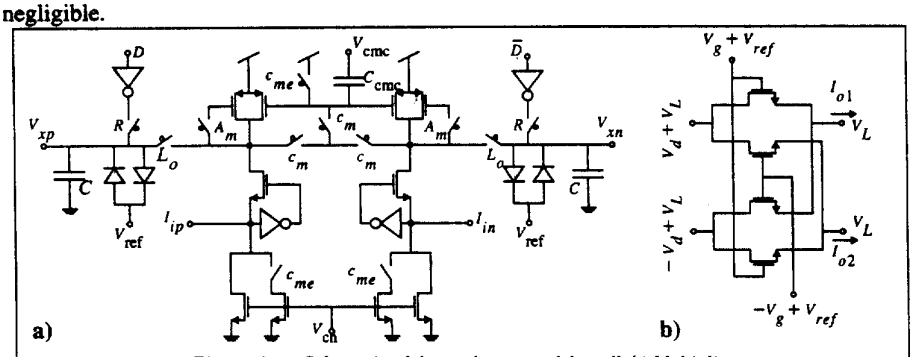

Figure 4 - a) Schematic of the analog core of the cell. b) Multiplier.

4. General characteristics and functionality

The external management of the chip is completely digital. Analog weights are specified and internally stored in digital form. For each template value, an adaptive stage transforms the digital code into an analog voltage, which is then transmitted to the network. This methodology results in weight insensitivity to process-parameter variations, as well as on accurate external control.

Every cell incorporates a photosensitive device, which allows the system to be optically initialized. Electrical initialization is also possible, 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 basis.

The digital circuitry at each cell includes a four-bit static memory (LLM), a completely programmable two-input digital gate (LLU), and initialization and control circuitry (LCCU) for many different operations. Memory contents can be moved from one location to another. The four-bit memory at each cell allows the network to store four complete images. Two additional "read-only-memories" with fixed +1 and −1 values are also available. Any memory can be used as input U or as initial conditions $X(0)$ of the network.

Data-transference processes (to or from the exterior of the chip, the CNN, the photosensors, or the LLU) are centralized on the LLM, as shown in Fig. 1.

Microinstructions (templates, offset terms and local logic operations) are digitally stored in an on-chip static RAM memory, which implements the analog and logic program registers (APR and LPR) and has a capacity of 8 instructions. The information contained in each of these instructions is described in Tab. 1. After an instruction set has been loaded into the digital memory, the individual CNN and logic operations can be used any number of times in any order. Minimum allowed increment of the analog parameters is smaller than the expected standard deviation in the weight of the multipliers due to mismatch effects. Therefore, coefficient discretization does not have a significant effect on the final precision. The allowed range of the coefficients can be scaled (at the expense of similar change in the time constant of the network) by an arbitrary value W_{max}. This is a consequence of using the FSR model [3].

Data description	Symbol	Number of coefficients	Bits per coefficient	Allowed values	Minimum increment
Feedback coefficients	a_d^c	9	8	$[-W_{max}, W_{max}]$	$W_{max}/128$
Control coefficients	b_d^c	9	8	$[-W_{max}, W_{max}]$	$W_{max}/128$
Offset term	d^c	1	8	$[-W_{max}, W_{max}]$	$W_{max}/128$
Boundary-cells state variable	x_s	1	2	$\{-1, 0, 1\}$	--------
Boundary-cells input value	u_s	1	2	$\{-1, 0, 1\}$	--------
LLU truth table	TT	1	4	Any	--------

***Table 1** - Information content of one set of coefficients in the joined APR and LPR static memory.*

5. Cell architecture and operating principles.

Fig. 4a shows the schematic of the analog part of the cell without the multipliers. In order for a circuit of this type to be practical it is necessary to have a relatively high precision, as well as high area efficiency. This is accomplished by using a simple and precise multiplier (see Fig. 4b), together with a technique that allows to reduce their number by a factor of two without loosing functionality. This technique uses the same multipliers to implement the weights of templates A and B. First each cell is initialized with its corresponding input u^c and the weights are programmed to represent the control template. The sum $\sum b_d^c u^d$ is evaluated in every cell, and the result stored in an analog memory (controlled by A_m). The cells are then initialized with the initial conditions $x^d(0)$, and the weights are programmed to represent the feedback template. In addition, a term $k^c x_{sat}$ representing the offset term is added to obtain $\sum a_d^c x^d(0) + k^c x_{sat}$. The current stored in the analog memory is then added from that resulting here, which results in Eq. 1. This technique eliminates 9 multipliers and, in addition cancels the offset of the processing circuitry. Since we are operating in a totally differential mode, the common mode of the differential state variables must be eliminated before processing. This is also accomplished using an analog memory (controlled by C_m). Network evolution begins when the feedback loops are closed using signal L_0. The results obtained can then be stored in one of the four local memories for later processing or for data downloading to the outside of the chip.

6. System architecture

System architecture is schematically represented in Fig. 5a. Adaptive stages are employed to tune electrical variables, compensate inaccuracies, and for automatic weight adjustment.

Bias and tuning stages, which generate the analog reference voltages required for the analog core of the cells, are located at every corner of the chip area, and connected among them. A digital decoder, placed at the left side of the cell array, is used to generate the 32 control signals 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. The circuitry located at the bottom of the cell array can be divided into two large sections. The first, located below, is the SRAM block implementing the APR and LPR, which contains 8 words (for 8 microinstructions) of 160 bits. The second, located above, contains 10 adaptive stages (nine weights plus the offset term). Finally, the blocks on the right side are used to generate some control signals and for miscellaneous purposes.

Fig. 5b shows the layout of one cell. The size of the cell is 180 x 170 μm^2, and the size of the complete chip is 7,7 x 6,8 mm^2. The design has been sent for fabrication in a standard 1μm CMOS technology, and is due from the foundry in the following weeks. Test results are expected to be available for presentation at the time of the conference.

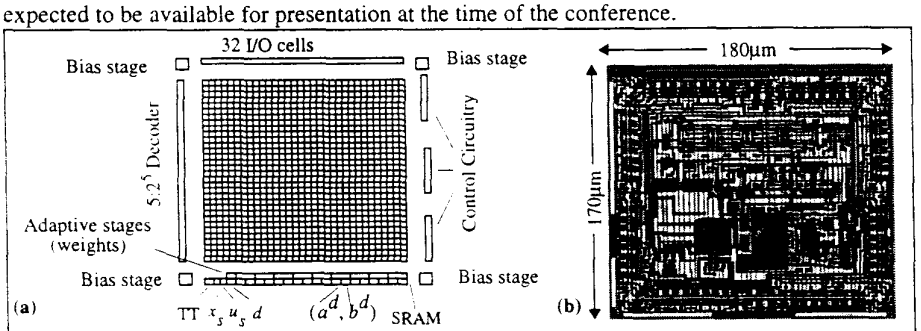

***Figure 5** - a) Schematic of the complete chip b) Layout of one cell of the FSR CNN universal chip.*

References

[1] T. Roska and L.O. Chua: "The CNN Universal Machine: An Analogic Array Computer", *IEEE Transactions on Circuits and Systems-II: Analog and Digital Signal Processing*, Vol., 40, No.-3, March 1993.

[2] L.O. Chua, T. Roska and P.L. Venetianer: "The CNN is Universal as the Turing Machine". *IEEE Trans. Circuits and Systems I: Fundamental Theory and Applications*, Vol. 40, pp 289-291, April 1993.

[3] S. Espejo, R. Domínguez-Castro, A. Rodríguez-Vázquez and R. Carmona: "Convergence and Stability of the FSR CNN Model". In this proceeding.

[4] S. Espejo, R. Domínguez-Castro, A. Rodríguez-Vázquez and R. Carmona: "Weight-Control Strategy for Programmable CNN Chips". In this proceeding.

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