2000 6TH IEEE International Workshop on Cellular Neural Networks and Their Applications Proceedings

# The CNNUC3: An Analog I/O 64 x 64 CNN Universal Machine Chip Prototype with 7-bit Analog Accuracy

*G. Liñán, S. Espejo, R. Domínguez-Castro and A. Rodríguez-Vázquez*
Instituto de Microelectrónica de Sevilla – CNM-CSIC
Edificio CICA-CNM, C/Tarfia s/n, 41012- Sevilla, SPAIN
Phone: +34 95 4239923, Fax: +34 95 4231832, E-mail: linan@imse.cnm.es

**ABSTRACT**

This paper describes a full-custom mixed-signal chip which embeds distributed optical signal acquisition, digitally-programmable analog parallel processing, and distributed image memory – cache – on a common silicon substrate. This chip, designed in a $0.5\mu m$ CMOS standard technology contains around $1,000,000$ transistors, 80% of which operate in analog mode; it is hence one the most complex mixed-signal chip reported to now. Chip functional features are in accordance to the CNN Universal Machine [1] paradigm: cellular, spatial-invariant array architecture; programmable local interactions among cells; randomly-selectable memory of instructions (elementary instructions are defined by specific values of the cell local interactions); random storage/retrieval of intermediate images; capability to complete algorithmic image processing tasks controlled by the user-selected stored instructions and interacting with the cache memory, etc. Thus, as illustrated in this paper, the chip is capable to complete complex spatio-temporal image processing tasks within short computation time ( $\sim 200ns$ for linear convolutions) and using a low power budget (<1.2W for the complete chip). The internal circuitry of the chip has been designed to operate in robust manner with >7-bit equivalent accuracy in the internal analog operations, which has been confirmed by experimental measurements. Hence, to all practical purposes, processing tasks completed by the chip have the same accuracy than those completed by digital processors preceded by 7-bit digital-to-analog converters for image digitalization. Such 7-bit accuracy is enough for most image processing applications. CNNUC3 has been demonstrated capable to implement – either directly or through template decomposition – 100% of the linear $3 \times 3$ templates in reported [2].

## 1. Introduction†

Full exploitation of Cellular Neural Network capabilities for image processing can only be exploited through VLSI chips. Several CNN and CNN-UM chips have been made in the past; particularly, those having a size larger than $10 \times 10$ and whose operation have been actually demonstrated through experimental evidence are described in [3]-[6]. The chips in [3], [4] and [5] are intended for binary images, while that in [6] is intended for gray-scale images. Those in [4] and [5] have been designed by keeping analog accuracy and robustness as targets, while those in [3] and [6] are targeted for maximum cell density. Finally, only the chip in [5] embeds distributed optical sensors for direct optical image acquisition.

CNNUC3 also embeds distributed optical sensors – it is a true focal-plane analog programmable array processor – and is capable to acquire gray-scale inputs and produce gray-scale outputs. It has been designed to achieve around 7-bit equivalent resolution in the internal analog operations, and its robust operation has been experimentally demonstrated through implementation of 100% of the linear $3 \times 3$ templates in reported [2]. Besides, it can be directly interfaced to digital equipments and incorporate all functional features needed for the realization of complex image processing algorithms.

## 2. General Characteristics

CNNUC3 consists basically of an array of $64 \times 64$ identical cells. Its processing is continuous-time and spatially-invariant, with radius-1 neighbourhood and the cell state equation given by the FSR model [7].

Feedback and control templates, and the offset (or bias) term are programmable with a resolution of eight bits

†. This work has been partially funded by ONR-NICOP N68171-98-C-9004, DICTAM IST-1999-19007 and TIC 990826.

– seven + sign. Input and output pixel values are analog (gray-scale) in general. However, specific functions are included for binary (black&white) images, which can also be processed. Spatially-distributed image memories are available for storage of both analog and binary images on a pixel-by-pixel basis. This allows fully-parallel ($64 \times 64$ wide) data-transference between processors and memory.

The prototype incorporates global-control and programming circuitry, located at the periphery of the array. This includes memory for 32 arbitrary sets of coefficients which, after programmed, can be randomly selected from the outside.

External control is completely digital. The interface has been designed to be easily embedded in conventional digital systems centred around a CPU or a DSP unit. Two bidirectional data-buses, one analog and one digital, are employed for image loading and downloading.

The prototype has been designed and manufactured in a 0.5μm, single poly, three metal layer CMOS technology. Cell size is $102.2 \times 120 \mu m^2$ – necessary to guarantee 7-bit equivalent accuracy in the internal analog operations, while total die size is $9.145 \times 9.534 mm^2$. The cell array occupies 58% of the die area. Nominal power supply is 3.3V, and worst-case power consumption is 1.2W. Table 1 shows the most relevant physical and electrical data of the prototype.

Table 1: Prototype Data.

| | |
|---|---|
| # of Cells | 4096 (64 x 64 Array) |
| # of Transistors | ~1.000.000 |
| # Transistors on the cell | 172 |
| Cell Size | 120 μm x 102.2 μm |
| Cell Density | ~82cells/mm$^2$ |
| Signal Swing | [0.6, 1.4]V (Programmable.) |
| Weight Swing | [2.15, 2.95]V (Programmable.) |
| Time Constant | ~1.2μs |
| Time Constant for Linear Convolutions | ~200ns |
| Spatial Uniformity on the Weight Signals. | 7.6-bits |
| I/O Digital Rate | 10MHz |
| I/O Analog Rate | 1MHz |
| Power Supply | 3.3V |
| Power per cell | 250μW |
| Power Consumption | 1.2W (worst case) |
| # of Templates Memorized | 32 |
| # of Instructions | 64 |
| Die Size | 9145.10 μm x 9534 μm |

## 3. Chip Description

Fig.2 (a) shows the chip architecture. The prototype incorporates some global-control and programming circuitry located at the array periphery. This includes memory for 32 arbitrary sets of CNN coefficients and for 64 arbitrary sets of 48 digital signals that are used as digital instructions to configure properly the cell in order to perform the different tasks that the cell is designed for. These memories can be randomly addressed from the outside once they have been programmed. Fig.1 (b) shows the chip microphotograph.

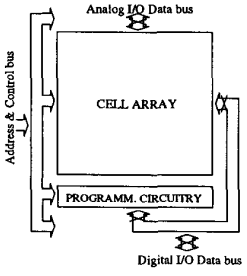


*Fig. 1: (a) Chip Architecture.*

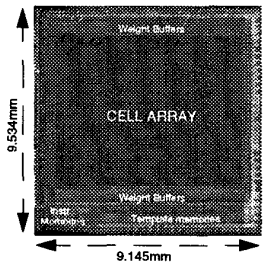


*Fig. 1: (b) Chip Microphotography*

### 3.1 Programming Circuitry

*3.1.1 Program-Memory Structure*

Fig.2 shows the peripheral programming blocks, including three parts. The first two parts are devoted to storage of 32 analog coefficient sets; 32 "analog instructions" can hence be stored on chip – one per coefficient set. Coefficients included into each set are classified in two groups. The first contains CNN processing parameters (9 feedback template elements, 9 control template elements, 2 bias terms, and 2 boundary condition levels). The second contains 8 analog reference levels that control electrical operation of the network (maximum, zero, and minimum values of the weight- and the state-variable signals, plus two additional analog biasing levels). Thus, a total of 30 analog coefficients are included at each of the 32 sets. Coefficient values are defined by 8-bits word, using a 7 bits+sign criterion, and are grouped into 16-bits words to match the width of the external digital data bus. Digital data stored at the RAM are used to drive digital to analog (DA) converters to obtain 30 analog programming levels, which are transmitted to the cell array trough global routing lines.The third part of the programming circuitry is dedicated to the storage of digital control words. The number of internal digital control signals required to perform the different operations is 35. In order to have sufficient flexibility for chip operation while at the same time having a simple external interface, values of digital control signals are grouped in vectors (digital/control instructions) of 48 ( $3 \times 16$ ) bits, which must be previously written (programmed) in a 64 words RAM. Afterwards, these vectors can be selected (and therefore applied to the network) using a small address bus.

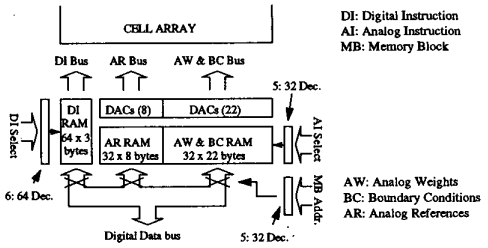


***Fig. 2:** Program storage and global control circuitry: functional description.*

### 3.2 CNN Circuitry

*3.2.1 Synapse*

Every cell in the array contains 20 synapses: 9 for he feedback template, 9 for the control template, and 2 for the offset term. Each synapse is driven by an input signal (either $v_x^c$, $v_u^c$, or $v_{sat}$) and by a global weight-programming signal ($v_A^d$ or $v_B^d$, with $d = 0, 1, \ldots, 9$ ), both of them in voltage form. Input signals $v_x^c$ and $v_u^c$ are local to cell and taken from the corresponding capacitors $C_x$ and $C_u$, while the weight signals (and also the saturation level $v_{sat}$) are global (common to all cells in the array), as required for a spatially uniform CNN. The weight signals and the saturation level (and some other 9 analog voltages) are generated by DA converters driven by the selected analog instruction in the programming circuitry. The output of the synapse is in current form. All current contributions to each cell are added at the cell' input node.

The synapse circuit is based on a circuit technique [8] that provides a four-quadrants-synapse behaviour from a single MOS transistor operating in its ohmic region, at the expense of a previous calibration step. The calibration step is needed because the synapse output current contains an "offset term". In particular, each synapse output, is of the form

$$I_s = I_s(v_i, v_w) \cong G(v_w)v_i + I_o(v_w) \tag{1}$$

where $v_i$ is some $v_x^c$, $v_u^c$, or $v_{sat}$, and $v_w$ some $v_A^d$ or $v_B^d$. Both $v_i$ and $v_w$ are relative to their corresponding

zero-levels: $v_{x0}$ and $v_{w0}$ respectively. It is clear that both $v_i$ and $v_w$ must be kept within some bounds for (1) to be valid. The signal ranges are limited to $[-v_{sat}, v_{sat}]$ and $[-w_{sat}, w_{sat}]$, respectively. Replacing the real form (1) of every synapse output into the integral form of the FSR cell state equation, yields

$$v_x^c(t) = v_x^c(0) + \frac{1}{C_x}\int_0^t \left\{ -I_g(v_x^c(\tau)) + \sum_{d=0}^{9} G(v_A^d) v_x^d(t) + \sum_{d=0}^{9} G(v_B^d) v_u^d + \; + \sum_{d=0}^{9} [I_o(v_A^d) + I_o(v_B^d)] \right\} d\tau \qquad (2)$$

The last sum on the right hand-side constitutes an undesired contribution to the offset term that must be cancelled. For this purpose, it needs to be "computed", stored, and substracted. All this is done very easily using the same 20 synapses (physically the same transistors: mismatch insensitive) in a previous step in which every $v_i$ is made zero (synapse input signals are all connected to $v_{x0}$). The resulting currents are added at the cell' input node, and the result (the term to be cancelled) is stored in a current memory. Substraction comes intrinsically associated to the current-memorization operation. Because the cancelled term depends on every weight signal, the cancellation must be repeated whenever the weight signals are changed.

Cancellation results in an effective elimination of any other offset current arising from circuitry imperfections. In fact, such elimination is needed or at least very convenient in most CNN hardware implementations because the output-referred random offsets of every synapse add together resulting in a large random (spatially variant) error for the "offset" or bias term. This offset term is usually the dominant error source in circuit implementations, and therefore, the small amount of additional hardware required for its elimination is commonly worth it.

*3.2.2 Current Memory*

The cancellation strategy employed in CNNUC3 follows a "store & substract" strategy. The main drawback of this alternative is that the resulting current-memory specifications are tight, with a simultaneous requirement of a large current range (maximum current to be stored) and low absolute current error. This has been solved using an extension of the $S^2I$ technique [9] based on the addition of a third current memorization stage. This results in a $S^3I$ current-memory. For optimum performance, the three current memories must be carefully sized because their corresponding signal ranges are different.

After the storage cycle, the resulting current source constitutes the biasing stage of the current conveyor employed at the cell' input node.

*3.2.3 Current Conveyor*

Because transistors employed at the synapses operate into ohmic region, and because a moderately large number of them (20) are connected to the same cell' input node, the input-impedance of the cell' input node must be very low. A class-II current conveyor is employed for this purpose. It is based on a common-gate amplifier with the input admittance boosted using an internal amplifier and negative feedback. The high-impedance output of the current conveyor is directly driven (through some initialization and control switches) to the integrating capacitor.

Because the random (spatially variant) component of the input-referred offset voltage of the current conveyors would affect the weights accuracy, a calibration circuitry can optionally be employed to cancel these offsets.

*3.2.4 Integrating and Sampling Capacitors*

The integrating capacitor is implemented by the input capacitance of the 9 synapses corresponding to the feedback template. An identical capacitor, implemented by the input capacitance of the 9 synapses corresponding to the control template is employed for the storage of the cell' input level ($u^c$). In fact, the role of each of the two capacitors can be selected for each CNN process. At first step, each of the two capacitors is precharged to the corresponding pixel value (gray or B&W) of one of two images. The distinction between $x^c(0)$ and $u^c$ (alternatively, between feedback and control templates) comes only afterwards, when one out of two control signals selects which capacitor ($C_x$) will receive the current conveyor' output current, while the other ($C_u$) remains disconnected. On the other hand since the transistors employed in the synapses operate in their ohmic region, the capacitances are fairly

linear.

*3.2.5 Voltage Limiter*

There are several very simple and hardware-efficient ways to implement the nonlinear resistor needed at the FSR cell state equation. A possibility is using two diodes and two reference levels. Diodes can be emulated using MOS transistors with moderately large aspect-ratio. This approach, however, has the disadvantages of smooth transition and finite slope in the saturation region and, much more important, its sensitivity to mismatch produces a random spatial variation of the cell saturation level. Note that the contribution of one cell to their neighbours, which is always proportional to the corresponding weight, is also proportional to the local value of $v_{sat}$ whenever the cell is saturated. Cell saturation occurs in many propagative templates, at the final steady state in binary output applications, and at the beginning of the transient in binary input applications. In other words: in practically all CNN processing functions. Therefore, the accuracy and uniformity of the local saturation levels is as important as the accuracy and uniformity of the weights.

Another alternative is based on using active diodes, which employ negative feedback to achieve abrupt transition, closer to the ideal, but still sensitive to mismatch due to amplifier offsets. A previous offset calibration cycle could be used to eliminate this effect, at the expense of a more complex circuitry and some additional global control lines. Still, one problem would be present: a substantial amount of power is needed in order to obtain sufficiently fast "diodes" without a significant overshoot (i.e., with a dominant time constant well below that of the CNN processing circuitry).

For these reasons, the limiter circuitry employed in CNNUC3 is somewhat involved. It is based on two comparators that detect when the cell' signal goes beyond either border of the linear region. In that case, the integrating capacitor is directly connected to one of two global wires driven by the corresponding saturation level $-v_{sat}$ or $v_{sat}$, whichever corresponds to the reached border. Although the input-referred offsets of the comparators will result in small errors, this deviations are effective only during the small transient (response time) of the comparator. Some minor additional tricks are needed to avoid possible instabilities in the proximity of the border points, and to allow for the state-variable signal to re-enter the linear region.

*3.2.6 Initialization and Control Circuitry*

A number of analog switches in every cell, and a similar number of global control lines are required to control the different cancellation circuits, the initialization process, and to actually launch the CNN transient. As a matter of fact, most of the control circuitry and global control lines are related to the enhanced functionalities described below.

### 3.3 Enhanced Functionalities

Additional functionalities have been incorporated for further improvement of the CNN Universal Machine capabilities [1] as required for relevant processing functions.

*3.3.1 Image Memories*

Every cell has the capability of storing four analog (gray-scale) and four binary (black & white) pixel values. At system level, this means that the chip can simultaneously store eight different images. These images can be used as inputs at any time during a processing sequence, and modified at any time as well; writting/reading time of the memories is around $0.1\mu s$. Binary memories employ conventional digital latches, while analog memories relay on "bottom-plate sampling" switched-capacitor stages following the guidelines given in [10]. By using these memories for storage of intermediate results significant computation time reductions are achieved in the realization of complex algorithms requiring iterative template applications, as well as in the realization of biffurcated-flow algorithms.

*3.3.2 Local Logic Unit*

The local logic unit (LLU) is a programmable boolean gate whose truth table is defined as part of the digital instructions stored in the programming circuitry. It allows a completely parallel realization of arbitrary bit-to-bit logic operations between images stored at two user-selectable binary memories. The resulting image can be down-loaded or stored in any of the four binary memories. Conventional digital circuitry is employed for this purpose.

*3.3.3 Freezing Mask*

Having a "freezing" mask means that the content of one user-selectable binary image memory can (optionally) be used as a flag which disables the evolution of the marked pixels during CNN processing transients, keeping their state variables time-invariant. The realization of this function requires just a few analog switches.

*3.3.4 Global Gates*

In many cases it is interesting to find out if some specific image is completely white or completely black, without wasting the time required to download the whole image. The prototype incorporates two global gates, one NAND and one NOR, to perform these logic operations over the pixel values of one user-selectable binary memory. With this functionality, the time required to check if some image is completely black or white is around $3\mu s$.

*3.3.5 Optical Input*

In many real-life high-speed applications, the information to be processed by the network is an image that is available in optical form while the output contains only a few details extracted from the input. In these situations, the read-out process is extremely simplified and hence speeded up. However, the input image is always a complete frame and therefore, the time needed to transfer the image to the array can constitute an actual bottleneck. In those cases, the capability of combining the sensory and the processing planes, provides a dramatic system performances enhancement, since it produces systems that do not only exploit the advantages of the fully parallel processing but also those of the fully parallel image acquisition that are provided by a matrix of photosensors merged with that of processors. CNNUC3 incorporates a photosensing device within each cell that allows the acquisition of images that are directly projected over the silicon surface. The sensing scheme is based on the integration, in the capacitor of any of the analog image memory, of the current that is generated by a diffusion-substrate photodiode.

## 4. Conclusions

This paper describes a recently designed analog programmable array processor chip. The new prototype, called CNNUC3, contains $64 \times 64$ cells arranged onto an array and follows the CNNUM computing paradigm. For that purpose it includes several specially designed modules like the Local Logic Unit, the Local Analog Memory, the Switch Configuration Register, the Global Gates or the Freezing Map, that increase prototype capabilities. The chip is able to process, store and provide gray-scale images. An optical acquisition mode is also available thus allowing not only the full exploitation of the parallel processing but also of the parallel acquisition.

## 5. References


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

[2] T.Roska, L. Kék, L. Nemes, Á. Zarándy, M. Brendel, *CSL - CNN Software Library - Version 7.2*, Analogical and Neural Computing Laboratory, Computer and Automation Institute, Hungarian Academy of Sciences, Budapest, 1998.

[3] A. Paasio, V. Porra, "A CNN Universal Machine with 295 cells/mm$^2$". *Proc. of the 1997 Int. Symposium on Non Lineal Theory and its Applications* (NOLTA'97), Honolulu, USA, 1997, pp. 221-224.

[4] P. Kinget and M. Steyaert, *Analog VLSI Integration of Massive Parallel Processing Systems*. Kluver Academic Publishers, ISBN: 0-7923-9823-8, 1997

[5] R. Domínguez-Castro et al., "A 0.8μm CMOS 2-D Programmable Mixed-Signal Focal-Plane Array Processor with On-Chip Binary Imaging and Instructions Storage". *IEEE J. Solid-State Circuits*, Vol. 32, pp. 1013-1026, No. 7, July 1997.

[6] J. Cruz and L. Chua, "A 16x16 Cellular Neural Network Universal Chip". *Analog Integrated Circuits and Signal Processing*, Vol. 15, pp. 226-238, March 1998.

[7] S. Espejo, R. Carmona, R. Domínguez-Castro and A. Rodríguez-Vázquez, "A VLSI-Oriented Continuous-Time CNN Model". *International Journal of Circuit Theory and Applications*. Vol. 24, pp 341-356, May-June 1996.

[8] R. Dominguez-Castro, A. Rodriguez-Vazquez, S. Espejo, R. Carmona, "Four-Quadrant One-Transistor-Synapse for High-Density CNN Implementations". *Proc. of 5th IEEE Int. Workshops on Cellular Neural Networks and their Applications*, pp. 243-248, London, April 1998.

[9] J.B. Hughes and K.W. Moulding, "S$^2$I: A Two-Step Approach to Switched-Currents". *Proc. 1993 IEEE Int. Symp. Circuits and System*, pp. 1235-1238, May1993.

[10] R. Carmona, S. Espejo, R. Dominguez-Castro, A. Rodriguez-Vazquez, T. Roska, T. Kozek, L.O. Chua, "A 0.5 μm CMOS CNN Analog Random Access Memory Chip for Massive Image Processing". *Proc. of 5th IEEE Int. Workshops on Cellular Neural Networks and their Applications*, pp. 271-276, London, April 1998.