Structure Reconfigurability of the CNNUC3 for Robust Template Operation

P. Földesy, G. Liñán, A. Rodríguez-Vázquez, S. Espejo, and R. Domínguez-Castro

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: peter@imse.cnm.es

ABSTRACT

In this paper we demonstrate the importance of the reconfigurabilitiy of a 64x64 cells size CNN-UM chip. As we show, in such a high complexity mixed-signal VLSI circuit the switch and internal reference level reconfigurability and reprogrammability play a crucial role for the robust operation of the system. The methodology for exploring the possibilities is three-fold, we consider theoretical results, error compensation methods, and the usage of special features of the design.

1. Introduction[†]

It is widely accepted that Cellular Neural Networks [1] exhibit outstanding image processing capabilities when compared to conventional purely digital approaches. In fact, despite their simple local non-linear dynamic evolution CNNs are able to implement a vast set of complex image processing functions with a processing speed incomparably higher than their digital counterparts. The key for this result is the parallel computation that could be simply defined as *let all the cells (pixels) to process by themselves at the same time*, therefore, the computing power of a specific CNN implementation depends directly on the number of cells in the array. For that reason, the natural trend for many years in the silicon implementations have been the increase of the number of neurons in the chips.

Nevertheless the increasing complexity of CNN implementations strongly implies the usage of robustness oriented architectures and circuit design techniques. In that sense, several approaches and solutions have been published. This paper presents some topics on robustness increase that have been implemented in a new analog-input analog-output 64×64 CNN chip called CNNUC3 [2]. The basic idea for solving some inaccuracies and to explore for new operations is the free reprogrammability of the switches that control the data transferences and the flow of the processes.

This paper is organized as follows: Section 2 briefly presents the chip architecture, the cell block diagram and the implemented state equation. Section 3 presents two general aspects to increase the template robustness while Section 4 presents two special functionalities added to the CNNUC3 prototype, namely the possibility of reconfiguration for implementing differential convolutions and that for DTCNN operation.

2. Chip Description and State Equation on CNNUC3

As most CNN chips, the CNNUC3 prototype, can be basically described as an array of identical cells, whose main function is to perform CNN operations on images (pixel arrays) of the same size (64×64). The implemented CNN algorithm is continuous-time, spatially-invariant, with linear template elements, and a radius-1 neighborhood, while the CNN state equation follows the so called Full Signal Range model (FSR) [3]. All elements of the feedback and control templates, as well as the bias (or offset) term, are programmable with a resolution of seven bits plus sign. From an external point of view, images may be analog (gray-scale), binary (black & white), or they can be directly captured by using the gray-scale photosensor included within each cell. Internally, from a CNN processing perspective, pixel values are treated as analog in general, with black & white images having extreme analog levels corresponding to the limits of the linear region. However, specific memories and some logic processing functions are included for binary images. Image storage is possible in both analog and binary form.

The cell array comprises the 64×64 inner cells and a surrounding ring of border cells used to establish the necessary spatial boundary conditions for CNN processes. Other miscellaneous functions like analog and digital buffering, control, and I/O tasks, are also included within the border cells. In addition to the network circuitry, the prototype includes some global control and programming circuitry located in the periphery of the cell array. This includes memory for 32 arbitrary sets of CNN coefficients, which after being programmed can be arbitrarily selected from the outside. Some other analog values related to the CNN processing circuitry, like the limits of the

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

0-7803-6344-2/00/$10.00 ©2000 IEEE

linear region and others, can also be programmed. Digital to Analog (DA) converters generate the analog-program signal levels transmitted to the cell array from the selected set of coefficients.

Fig.1 (a) shows the chip architecture, the prototype incorporates some global-control and programming circuitry located in the periphery of the array. This includes memory for 32 arbitrary sets of CNN coefficients and for 64 arbitrary sets of 35 digital signals that are used as digital instructions to configure properly the cell in order to perform different task ranging from running a CNN process to configure the cell I/O circuitry. These memories can be randomly addressed from the hosting platform once they have been programmed. Fig.1 (b) shows the chip microphotography.

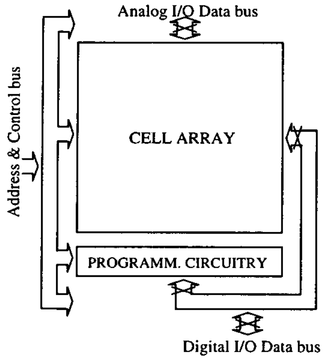


Fig. 1: (a) Chip Architecture.

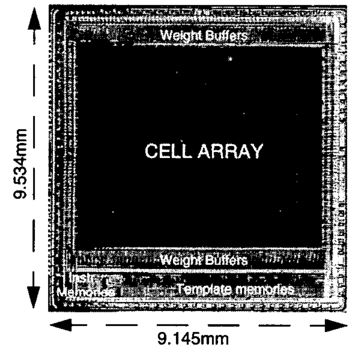


Fig. 1: (b) Chip Microphotography.

Fig.2 shows the basic structure of a template execution core containing the convolution masks, the possible input sources, the insertion point of the fixed-state mask, and a synapse current calibration circuit. The data and operation flow is controlled by several switches and reference values. A general template execution contains up to two calibration phases, initial state and input capacitor initialization, transient evolution, and result storing. During the calibration the convolution sum of the used template and the mid-gray (zero) level is stored as a reference for the image processing. In the transient evolution phase, this sum is subtracted from the incomi ng synapse current producing the current that is integrated in the state capacitor.

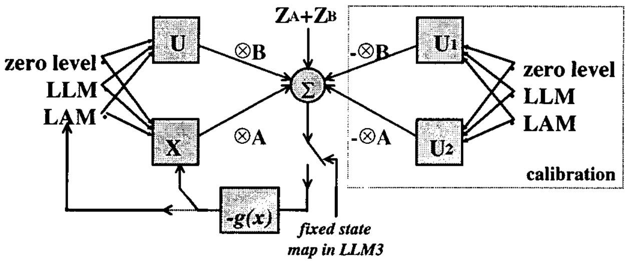


Fig. 2: The structure of the cell processing core of a cell in the CNNUC3 chip.

This differential structure allows a high-precision operation.The result of the transient evolution can be stored in any of the Local Memories of the cell, either analog or digital.

Equation (1) shows the state equation of the cells including the calibration sum (note that the cells have been designed using the full-signal range (FSR) model [3]).

$$:\frac{dx^{ij}(t)}{dt} = -g[x_{ij}(t)] + \sum_{C(k,\,l) \in S_r(i,\,j)} A(i,\,j;k,\,l) \cdot x_{kl}(t) + B(i,\,j;k,\,l) \cdot u_{kl} + Z_A + Z_B - \qquad (1)$$
$$- \sum_{C(k,\,l) \in S_r(i,\,j)} A(i,\,j;k,\,l) \cdot u^1_{kl} + B(i,\,j;k,\,l) \cdot u^2_{kl}$$

$$g[x_{ij}(t)] = \begin{cases} m_L & ,x^{ij}(t) < -1 \\ 0 & ,\left|x^{ij}(t)\right| < 1 \\ m_R & ,x^{ij}(t) > 1 \end{cases} \qquad \begin{matrix} m_L \rightarrow -\infty \\ m_R \rightarrow \infty \end{matrix} \qquad (2)$$

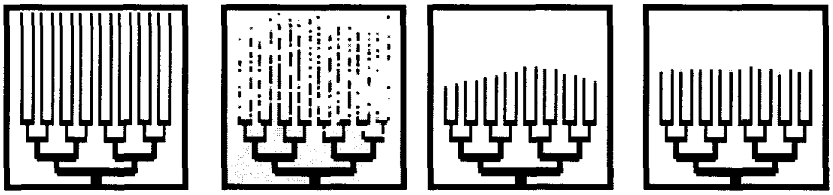
3. General Considerations on Template Robustness

3.1 Reducing template function complexity by the fixed-state map

The usage of the fixed state in order to avoid pixels to change seems to be a trivial method, but it produces very good results when applied to optimatization of the template robustness. From a theoretical point of view, with the help of *"frozen"* cells having black (or white) values, the truth-table of the template function can be simplified by introducing several *don't care* elements (see more about optimal function to template transformation [4]). This method directly leads to a simpler input-output mapping and a more robust template structure. On the other hand, in practical cases when the error caused by the random deviation on some crucial technologycal parameter cannot be eliminated by template optimization, the transient freezing can help to reduce this amount of error.

Let us consider as an application example the reconstruction and the hole-filling templates [5]. The task is to recover black areas over white background marked by black parts. In the published template, a strong feedforward connection controls the propagation of the recovery transient. The functionality of this control is to stop the transient at white pixels. However if this functionality is replaced by disabling the possibility of change only for white pixels, the remaining task is easier to fulfil, actually it consists into maintain and propagate a black wave starting from any black pixels (see Fig.3).

The idea behind the hole-filling operation is very similar and consequently, the same performance enhancements can be achieved when the task is re-formulated as the recovery of any white area which have connection to (or marked by) the white boundary cells.



(a) Original Form

(b) Result with the Published Template

(c) Result with Fixed-State mask.

(d) Result with Fixed-State mask and offset compensation

Fig. 3: Example of applicaiton of the Fixed-State map to the Reconstruction Operation.

Fig.3 shows an example of the usage of the fixed-state map technique in order to increase the robustness of a template execution. The task is the reconstruction of the tree-shaped form starting from the frame connection point. Fig.3 (a) shows the roginal form, Fig.3 (b) shows the result of the published reconstruction template [5], Fig.3 (c) shows the result of using the fixed-state mask, and image (d) shows the result of the same method when a current offset error compensation (see Section 4.1) scheme is applied.

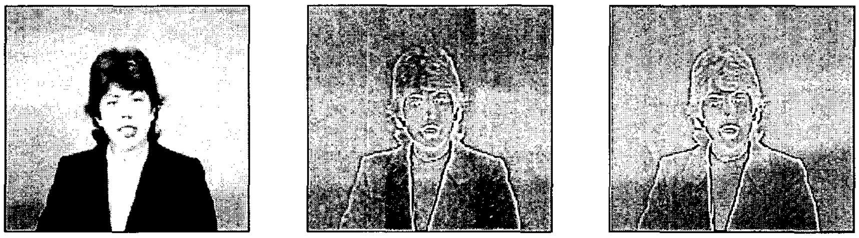
3.2 Strong negative self-feedback in uncoupled templates

It is well known that negative feedback increases stability and robustness of any system. In this section the importance of negative self-feedback for robust template operation is demonstrated through experimental results. The analysis of the dynamic routes of the state variables demonstrates that negative self-feedback produces an unique (not bistable nor time dependent) equilibrium point at the end of the transient evolution of the network. Furthermore, this equilibrium point does not depend on the initial value of the state variable, consequently, all the problems arising from the initialization of the CNN transient can be neglected.

Two important reasons for using negative self-feedback even in binary output cases can be found.

- In high-speed VLSI implementations the time constant of control signal propagation and reference level distribution is in the range of the cell time constant. Hence the initialization of CNN transient evolution can be disturbed by mean of clock signal cross-talk or voltage drop on the reference levels. It causes that the cells will slightly behave differently depending on their position in the cell array. Or with other words, the robustness of the operation decreases in an unpredictable manner. In order to avoid the initialization problems the value of the output should not depend on the initial conditions of the cells. Or, the used template should guarantee its insensitivity.

- The negative self-feedback compresses the voltage swing of the integrated synapse currents. This phenomenon also helps to increase the working precision because higher values on the template elements can be used thus increasing the signal-to-noise (here noise refers to either electrical noise and spatial noise).



(a) Input Image

(b) Small negative self-feedback

(c) Large negative self-feedback

Fig. 4: Example of the usage of negative self-feedback in a high-pass filtering process.

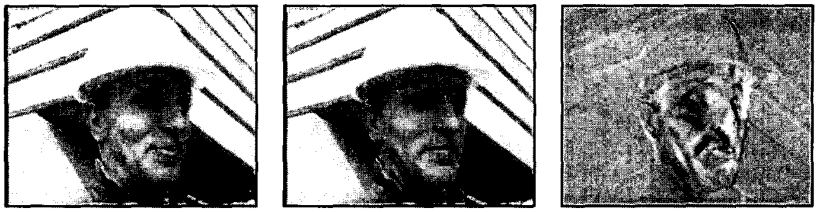
Fig.4 shows the application of negative self-feedback for the high-pass filtering operation in gray-scale input images. The used image size is 176x144 pixels (QCIF) and was divided into chip nine overlapped pieces to fit the chip size. Producing a black and white image containing only the edges on the input image from the result in Fig.4(c) only requires a thresholding process.

4. Two Spetial Functionalities on CNNUC3

4.1 Differential input convolution

The first circuit specific extension that we present is the change of the role of the synapse current calibration circuitry. Since this current memory is not restricted inherently to provide the zero offset current that correspond to the zero input level, it can be used to store the convolution sum of non-zero images. Therefore, it is possible to perform fully differential input convolutions that cannot be implemented on the original CNN-UM architecture [6].

The original intended values for the parameters u^1, u^2 in equation (1) were the uniform analog zero level for input and state variables. However if their values are changed to meaningful pixel values, the differences among the two pixels become the argument of the applied convolution mask. Applications of this enhanced functionality are the possibility of having linear arithmetic operations using analog images. Among those operations, the substarction of two images (see Fig.5) is specially interesting since it can be used as a early step in motion detection algorithms.



(a) 4th Frame (b) 5th Frame (c) Difference

Fig. 5: Example of analog image substraction.

A more sophisticated application is the low-spatial offset error compensation. Due to the differential input of the convolution masks, if a proper error map is used, the errors of low-spatial frequencies can be eliminated efficiently (see Fig.3d).

4.2 Reconfiguration for DTCNN

The second extended operation provides a fast binary result using two independent input and feed-forward convolution mask. Now, the basic idea is to integrate the current of the synapses directly into a local analog memory instead of the state capacitor. In this configuration the feed-back path is opened and so the convolution matrix of the feedback template can be used as a second feedforward convolution mask. The memory capacitance is approximately nine times smaller than the state capacitance causing faster saturation and binary output. By continuous feeding back the output to one of the inputs, the chip behaves as a DTCNN architecture [7] .

5. Conclusions

We have demonstrated, by experimental results, that the possibility of having some free reconfiguration and reprogrammabality of the switches controlling the data paths and the process executions generally enhaces the robustness of a CNN chip and extends its processing capabilities almost in an unpredictable manner.

6. References

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

[2] G. Liñán, S. Espejo, R. Domínguez-Castro and A. Rodriguez-Vazquez, "The CNNUC3: An Analog I/O 64 x 64 CNN Universal Machine Prototype with 7-bit Analog Accuracy", *Proc. of CNNA2000*, submitted.

[3] 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, No. 3, pp 341-356, May-June 1996.

[4] L. Nemes, L. O. Chua, T. Roska, "Implementation of Arbitrary Boolean Functions on the CNN Universal Machine". *Int. J. Circuit Theory and Applications*, Vol. 26. No. 6, pp. 593-610, 1998.

[5] 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.

[6] 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.

[7] H. Harrer, J. A. Nossek, "Discrete-Time Cellular Networks". *Int. J. Circuit Theory and Applications*, Vol.20, pp. 435-467, September 1992.