A macromodel fault generator for cellular neural networks

Grimaila, M.R.; Pineda de Gyvez, J.

Published in:
Proceedings of the Third IEEE International Workshop on Cellular Neural Networks and their Applications, CNNA-94, 18-21 December 1994, Rome, Italy

DOI:
10.1109/CNNA.1994.381647

Published: 01/01/1994

Document Version
Publisher’s PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

- A submitted manuscript is the author’s version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher’s website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

Link to publication

Citation for published version (APA):

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

Take down policy
If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Download date: 05. Dec. 2018
A MACROMODEL FAULT GENERATOR FOR CELLULAR NEURAL NETWORKS

Michael Russell Grimalia and Jose Pineda de Gyvez
Department of Electrical Engineering, Texas A&M University
College Station, Texas 77843-3128, U.S.A.
tel.: +1-409-845-7477 fax: +1-409-845-7161
e-mail: grimalia@pineda.tamu.edu, gyvez@pineda.tamu.edu

Abstract - A CAD tool based on SPICE macromodels to simulate simplified faulty circuit realizations of a fully programmable, two dimensional cellular neural network (CNN) is presented. The models can be easily adapted to match the electrical parameters of real circuit implementations. Generic macromodels for both current mode and voltage mode CNNs are provided. The macromodels not only simulate the conceptual CNN cell, but also provide the capability to model actual CNN architectures and their nonidealities. Moreover, macromodeling provides the capability to determine the effect of parameter variation on the operation of the CNN efficiently without the need for computationally expensive, exhaustive circuit simulations. We have used the CNN macromodels to develop robust testing strategies for detecting faults in VLSI implementations of CNN arrays. Three fault cases are introduced into a CNN array to provide insight to the usefulness of macromodeling.

1. Introduction

Cellular Neural Networks (CNN's) are a special class of continuous time neural networks often used for real time image processing. The two dimensional CNN has proved useful for edge detection, noise removal, image thinning, hole filling, image filtering, motion detection, and character recognition[1]. The robust design of CNN's often requires a large number of simulations for design verification. Unfortunately, the circuit representations of the CNN are large and require much CPU time to simulate circuit faults. We propose a CAD tool that generates SPICE macromodels to simulate faults in the hardware of two different CNN topologies. These topologies are the voltage mode and the current mode CNN. The electrical parameters of each macromodeled element are annotated with parameters to match real hardware implementations. The complete CNN macromodel is then simulated using SPICE. The results of the simulation are used to characterize the behavior of a CNN and it's sensitivity to parameter variation, among other things. In other words, the macromodeling fault generator offers the capability for parametric and catastrophic fault simulations. Chua [2] originally introduced a piecewise linear SPICE simulator for simulating the CNN architecture. His model provided the capability for simple CNN simulations, but lacked the ability to model the interconnection impedances of real hardware architectures. Lee [3] and Varientos [4] both

1. This work was supported by the Office of Naval Research under contract grant N00014-91-1-0516

0-7803-2070-0/94/$4.00 © 1994 IEEE
presented software CNN behavioral simulators that yield quick results, but that lack the ability to model any of the electrical parameters found in hardware implementations. Macromodels provide the capability to model offsets, impedance mismatches, and other circuit nonidealities. The reduced complexity of the proposed CNN macromodels allow results to be obtained more quickly than a complete transistor level circuit model. Simulating macromodeled CNN arrays provides an attractive compromise between behavioral simulations and complete circuit simulations.

2. Behavioral Aspects
A two-dimensional CNN array contains MxN locally connected cells. A 5x5 CNN array is shown in Fig. 1. The CNN array is controlled by choosing the initial conditions, setting inputs, and selecting the desired mode of operation. The CNN provides three behavioral modes of operation. They are: i) Cell initialization, ii) Evaluation, and iii) Result extraction. The behavioral modes are selected by two mutually exclusive voltage controlled inputs, SetIC and Eval, that activate two voltage controlled switches inside the integrator. In the initialization phase, the summer is disconnected from the integrator to allow the state of the cell to start at a known initial condition. The initial condition is selected by activating the signal SetIC. When SetIC is activated the initial condition voltage (IC) is applied directly across the capacitor, which is the physical quantity that represents the state of the cell. In some applications it is desired to use the input image as the initial condition of the array. For this case, the IC signal is connected to the IN11 signal of each cell. The evaluate mode is entered when the Eval signal is activated which connects the summer to the integrator and allowing the dynamic processing to commence. If neither SetIC nor Eval is activated, the output stage is separated from the integrator and ideally the state of the cell is held indefinitely.

3. Electrical Topology
The macromodeled CNN cell contains 42 voltage inputs and 1 voltage output as shown in Fig. 2. The cell has both global and local voltage inputs. The global inputs are common to all cells and include the feedback template (A00-A22), the control template (B00-B22), the bias value (Bias), the mode control values (SetIC and Eval), the initial condition value (IC), and the
power supplies (VDD, GND, and VSS). The local inputs (IN00-IN22, OUT00-OUT10, OUT12-OUT22) are unique to each cell and are determined by the location of the cell within the CNN array. Each cell in the array has an activated output voltage (OUTPUT) and optionally a pre-activated output voltage (STATE). The addition of the STATE output is motivated by Pineda's research into the application of a multilayer CNN used to process color images[5]. The cells located on the outer edges of the array obtain their INxx and OUTxx inputs from the Border value, which is added to the basic CNN architecture to facilitate testing of the array.

The CNN macromodel consists of multipliers, a summer, an integrator, and an activation function. The internal processing of the cell may take place in either the current or voltage domains. Current mode operation is characterized by the use of transconductance multipliers, a current summer, and a current integrator. The voltage mode CNN makes use of voltage multipliers, a voltage summer, and a voltage integrator. In either case, the inputs and outputs of the CNN macromodel are voltages. Each element of the macromodel contains complex input/output impedance's to simulate interconnection parasitics, voltage sources to simulate offsets, and the corresponding elements to perform its original function. The use of the generic complex impedance block is considered a single element in the fault analysis. The complex impedance block may be used to introduce poles and zeros into the transfer function of each macromodeled block.

The current mode (transconductance) and the voltage mode multiplier are shown in Fig. 3. Current mode multipliers are implemented using a voltage controlled current source with an annotated transconductance value, gm. Voltage mode multipliers are based upon voltage controlled voltage sources with an annotated voltage gain, Av. The multiplier input voltages are developed across a line impedance, an offset voltage source, and across a load impedance. One can think of the line impedance as representing the impedance of the path between the output of a cell and the input to the specific multiplier. The multiplier generates an output that is proportional to the product of the two input load voltages. The current/voltage output is developed across/through a source impedance and then diode limited to the supply range to simulate the saturated region of actual multipliers. Although HSPICE simulator allows dependent sources to be annotated with a minimum and maximum value to model saturation, we used diode limited outputs in our macromodels for SPICE compatibility purposes.

Fig. 3 - Macromodeled Multipliers - (a) Transconductance Multiplier, (b) Voltage Multiplier

The macromodeled summers shown in Fig. 4 accept the outputs of the 18 multipliers and the Bias voltage to produce an output proportional to their sum. This output can be a current or a
voltage depending upon the CNN topology. The current mode summer requires that the Bias voltage is first converted to a current before the summing with the multiplier currents. The current is summed by a simple wire, developed through a source impedance, and diode limited to avoid unbounded output values. Note that the summer contains a load impedance to provide a DC path to ground when the summer is disconnected from the integrator. The voltage mode summer requires a dependent voltage source that produces the sum of the 19 voltage inputs. The voltage output is developed through a source impedance and is diode limited to the supply rails.

![Fig. 4 - Macromodeled Summers - (a) Current Summer, (b) Voltage Summer](image)

The integrator contains the processing elements of the CNN that primarily determine the dynamics of the array. Fig. 5 shows the block diagram of the current mode and voltage mode integrator. The current mode and the voltage mode integrator both contain two voltage controlled switches. The SW1 switch is controlled by the SetIC signal and when activated forces the state of the cell to the voltage applied at the IC input. The SW2 switch is controlled by the Eval signal and when activated connects the output of the summer to the integrator. If both SetIC and Eval are inactive, the state of the cell is held until the capacitor discharges. The mode control switches are modeled using ideal voltage controlled resistors that are annotated with "on" and "off" resistance to model real switches.

![Fig. 5 - Macromodeled Integrators - (a) Current Integrator, (b) Voltage Integrator](image)

The activation function of the CNN is identical in both the current mode and the voltage mode CNN. The activation function is modeled using a polynomial voltage controlled voltage source as shown in Fig. 6. The polynomial source can be adjusted to model any shape activation function, any gain, and any output offset voltage. The output voltage is developed through an output resistance and diode limited to the supply range. Note that the output voltage is internally fed back to the input of feedback multiplier A1I, eliminating the need for
an external OUT11 input signal. The result is that each CNN cell has only 8 inputs (OUT00-OUT10 and OUT12-OUT22) required from the surrounding cells outputs.

![Fig. 6 - Macromodeled Activation Function](image)

4. Fault Model Case Studies
To show the versatility of the CNN macromodel generator, we analyze the operation of both a current mode and a voltage mode CNN when processing an image using an edge detection template[6]. In both cases, a 5x5 CNN array is used to illustrate the useful properties of faulting a CNN macromodel. The integration resistor was chosen to be 100K ohms and the integration capacitor was chosen to be 1pF. This results in a time constant of \( t = 100 \text{ns} \) which determines the minimum time (5\( t \)) required for the CNN to converge. The supplies were selected to be \( VDD=\text{BLACK}=+5.0\text{V} \) and \( VSS=\text{WHITE}=-5.0\text{V} \). An activation slope of \( A_v=2 \) was chosen for this simulation to insure stable outputs when the absolute value of the state of the cell is greater than 2.5 volts. In the voltage mode CNN, the sum of the output resistance of the voltage summer and the integration resistor, along with the integration capacitance, determine the time constant of the array. Typically, the output impedance of the voltage summer is small, thus the integration resistance dominates. Observe that in the current mode CNN implementation, the output impedance of the multipliers is in parallel with the integration resistance during the evaluation phase. If the integration resistance is not smaller than the parallel combination of all the output stages of the multipliers, the integration time constant is greatly reduced. In this presentation, we discuss three fault cases: (1) a voltage offset in feedback multiplier A11 of 100mV, (2) a change of the initial condition switch impedance from 10M\( \Omega \) to 100K\( \Omega \) in the integrator, and (3) a time constant mismatch resulting from reducing the integration capacitance from 1.0 \( \text{pF} \) to 0.1 \( \text{pF} \). The center cell was selected to annotate all the fault cases. Simulation results showed that the voltage mode CNN converged to incorrect values for each of the fault cases. Conversely, the current mode CNN converged correctly for each fault case. The cause of this result is that the transconductance in the current mode multiplier is much less than one, which reduce the effect of faults by a factor of \( gm \). The voltage multiplier has a gain \( A_v \), which is set to one in this discussion. Fig. 7 depicts the inputs and the faulted and unfaulted outputs of the voltage mode CNN. Fig. 7a shows the applied input to the CNN array. Fig. 7b shows the correct outputs when using nominal component values. In Fig. 7c, we see the output of the CNN array when the center cell has the feedback multiplier A11 annotated with an offset fault of 1% of the supply voltage (100mV). Fig. 7d displays the output when the switch inside the integrator of the center cell has been reduced from 10M\( \Omega \) to 100K\( \Omega \). Fig. 7e shows the output when a time constant fault is introduced by reducing the integration capacitance from 1.0 \( \text{pF} \) to 0.1 \( \text{pF} \).
The voltage mode CNN cell states are displayed in Fig. 8a. The fault free case converges to WHITE correctly within 5τ. The offset and the switch impedance fault cases result in the cell converging incorrectly to BLACK. In the case of a time constant fault, the cell converges too quickly to the correct state of WHITE resulting in the surrounding cells converging incorrectly to BLACK. The current mode CNN cell states are displayed in Fig. 8b. In all fault cases the cell converged correctly to WHITE.

Fig. 8 - Center cell state during convergence - (a) Voltage mode CNN, (b) Current mode CNN

5. References