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

SIRENA: A Simulation Environment for CNNs

R. Domínguez-Castro, S. Espejo, A. Rodríguez-Vázquez, I. García-Vargas,
J.F. Ramos 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 *- SIRENA is a general simulation environment for artificial neural networks, with emphasis towards CNNs. A especial interest has been placed in allowing the simulation and modelling of the non-ideal effects expected from VLSI implementations. SIRENA allows the simulation of CNNs in greater detail than conventional CNN simulators, and much more efficiently that SPICE-type electrical simulators.*

1. Introduction

CNNs are massively-parallel processing systems, specially suited for real-time image processing applications. Since its algorithmic description in 1988 [1], [2] many applications have been reported [2], [3], [4], [5], and several analog integrated circuits have been developed for real-time specific applications, [6], [7], [8].

Efficient simulators are needed to develop new applications (at an algorithmic level) as well as for functional verification of complex integrated circuits (lower description level). While algorithmic simulators are being widely used [9], proper CAD tools for the development of CNN VLSI integrated circuits are still lacking. The widely available SPICE-type electrical simulators give a high level of detail in the system description but are limited to working with around 10,000 transistors. For example, it has been estimated that an HSPICE simulator [10] running on a Sparc-10 workstation (100MIPS), can simulate circuits of at most 20,000 transistors and may take several days of CPU in a simple simulation. An average size (32x32) programmable CNN designed by the authors has around 250,000 transistors [11].

Consequently, conventional design tools are poorly suited for the verification of large-scale analog integrated circuits. On the other hand, actual algorithmic level simulators are unable to consider the non-ideal effects of real electronic circuits. SIRENA has been designed to overcome this limitation. This simulation environment allows the description and simulation of large neural networks with different levels of description. In addition to a high simulation efficiency, SIRENA is equipped with a powerful and friendly graphical interface under the X-windows framework.

2. SIRENA: A environment for CNN simulation and modeling

An schematic block diagram of SIRENA is shown in Fig. 1. Shaded areas represents the major parts of the system: the nucleus or actual simulator (NSS), the model generator (GMS), and the graphic interface between the user and the simulator nucleus.

0-7803-2070-0/94/$4.00 © 1994 IEEE

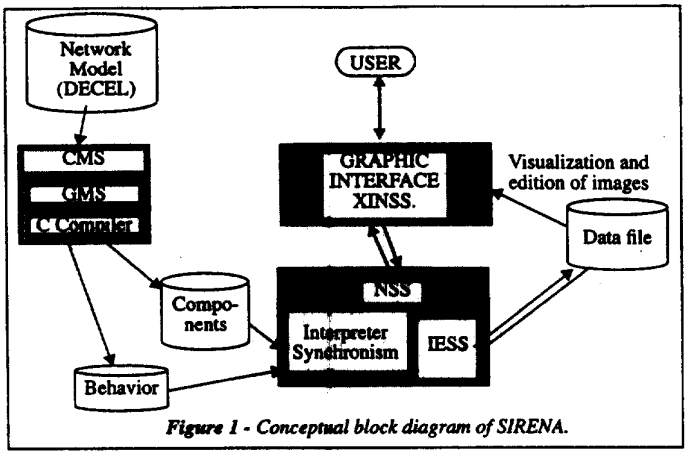

Figure 1 - Conceptual block diagram of SIRENA.

SIRENA allows simulation of general neural networks, and can be considered as functional simulator in which the description and use of general macromodels is possible and simple. However, its development has been focused on the simulation of CNNs.

The simulation environment has been developed using object-oriented programing techniques, and highly portable languages like C and C++. Currently, it runs under the UNIX operating system and the X-Window system. These characteristics make of SIRENA an efficient, actual and friendly simulation environment.

The following sections describe in greater detail each of the major components of SIRENA.

3. The Simulator Core (NSS)

The principal function of the simulator core is to solve systems of nonlinear differential equations, as required to simulate the dynamic evolution of CNNs [1]. The user can select among a set of well known integration algorithms (fourth-order Runge-Kutta with either automatic integration-time-step control or with fixed time-step). Other integration algorithms can be easily incorporated, (for instance Forward-Euler, Backward-Euler, Trapezoidal, Gear Methods, etc.), and will be included in future releases of SIRENA.

The simulator core allows the following types of analysis:

- Transient Analysis: this basic capability evaluates the time evolution of the network, providing the output signals or images specified by the user with time intervals also defined by the user. This is the main type of analysis, upon which the following ones are based.
- Sensibility Analysis: this functionality consists in the automatic realization of multiple transient analysis, sweeping the value of a single parameter or variable with a specified increment and range. The motivation for the inclusion of this capability is to facilitate the evaluation of the sensibility of a given network to variations on the different parameters of the algorithm (template coefficients, offset parameter, time constant, etc.) or on other parameters describing the behavior of real circuit blocks (for instance the power supplies or some process parameter). It is also very convenient to optimize the robustness of a given application (design-centering).
- Montecarlo Analysis: it allows the realization of multiple simulations of a given network, with random variations of one or more parameters following some specific probability

distribution (gaussian or uniform). This type of analysis is useful for the evaluation of the statistical yield of a given network under random deviations of its parameters, either local variations (different deviation on each cell) or global (same deviation on all cells). This type of evaluation is a prerequisite for the fabrication of high-complexity analog VLSI circuit employing small devices. It is however extremely expensive in terms of CPU time, and hence, its realization with traditional SPICE-like electrical simulators is virtually impossible.

- Noise Analysis: this feature is intended for the evaluation of noise influence on the evolution of a given network. Because CNNs are strongly nonlinear systems, the conventional approach of traditional electrical simulators, in which noise analysis is associated to small signal equivalents, can not be applied. In our case, the simulator injects noise on user-specified points (signals or parameters). The probability distribution and spectral density of the noise can be arbitrarily shaped. A combined noise and sensibility analysis allows the evaluation of the noise threshold that can be tolerated by the network.
- Multilayer Analysis: the simulator core of SIRENA allows the simultaneous simulation of several CNN layers, which can be connected among them. These layers can be described by the same cell model (three-dimensional CNNs), or by different models (multilayered CNN).
- Multinetwork Analysis: in addition to the simulation of three-dimensional and multilayered CNNs, NSS allows the evaluation of complex systems compound of several CNNs, which interchange information at specified time instants or following a described protocol (information interchange after convergency of the individual network processes). This functionality is controlled by the task-scheduler.

3.1. Task scheduler

NSS permits the independent simulation of several CNN layers, controlling the sequence of processes and allowing information interchange among the different layers at specific time instants. This permits the evaluation of sequential and multipath algorithms [12]. From a general point of view, NSS allows an arbitrary flow control with conditional and jump operators, as well as a wide catalog of algebraic operations. An example of this type of algorithms is shown in Fig. 2, which describes the object-counting application [12].

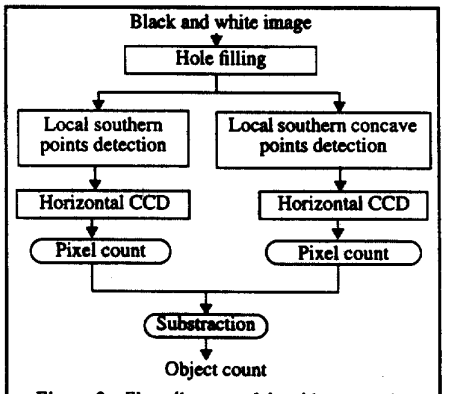

***Figure 2** - Flow diagram of the object counting application of CNNs [12].*

4. The Model Generator (GMS)

A previous step in the simulation of a CNN is the description of the individual cells. The capability of employing user-defined models of individual cells is a major feature of SIRENA. Cell model can range from the pure algorithmic level to highly detailed models with a large number of non ideal characteristics accounting for predictable effects in electronic implementations (impedance coupling among input and output nodes of the cells, undesired nonlinearities, parasitic element effects, non-uniformity caused by element mismatch, etc.)

GMS allows the user to define, in a simple manner, his own CNN models, which later on can be used for CNN simulations. The definition of a CNN requires the specification of its *components* and their *behavior*.

Components include the set of layer and sublayers of the network, as well as their characteristics (layer size in term of number of cells, type and parameters of boundary cells, neighborhood size, uniform or nonuniform type of layer, and others).

The behavior is described by a set of algebraic and differential equations relating the different pre-specified components, which determine the dynamic evolution of the network.

Behavior description is easily made using an specific purpose programming language developed for this purpose: DECEL. After the cell models have been entered, GMS compiles and links them with NSS in order to use them in the simulation.

5. The Graphics Front End

The graphic front end of the simulation environment has been developed under the X-Windows framework and performs multiple functions like the edition of input signals (usually two-dimensional images), the visualization of the simulation output (like individual element values as a function of time, images, or sequences of images), and interactive control of the simulation. The following is a brief description of the main tools included in the front-end package.

5.1. Input/Output interface (IESS)

IESS is in fact a library of dedicated functions for *information-file* management. It provides a common base for individual tools in SIRENA, allowing the integration of the different packages in a compact environment.

This interface include the parsers required to analyze the input and output files, detect their errors, and provide the proper error messages. In addition, it contains a set of utilities dedicated to translate different graphic formats required for the visualization of signals with external tools (for instance "xgraph").

5.2. Matrix visualizer (XVMS)

This tool allows the visualization of an image or a sequence of images specified in a command line. The major features are

- Zooming
- Image sequence visualization. The sequence can be compound of a set of images corresponding to different time instants in a particular CNN process, or by images obtained from different networks (for instance in sequential algorithm). Fig. 3 illustrates an example of this last possibility, borders extraction of the black and white image obtained using the noise-filtering template on a grey-level image.
- Visualization of different layers of three-dimensional networks.

5.3. Matrix editor (XEVMS)

XEVMS is a graphic editor, which can be considered as enhanced version of the matrix visualizer XVMS. Added capabilities are dedicated to allow the creation and modification of images, which can be easily used as inputs to the simulator. These capabilities are:

- File management operations: saving, reading, creating, etc.
- Import/export of specified sections of a given image from/to graphic files in conventional formats (PBM format).
- Image resolution modification, in order to adapt a given image to a particular network size.
- Graphical and numerical edition of individual pixel values.
- Image sequence order modification.

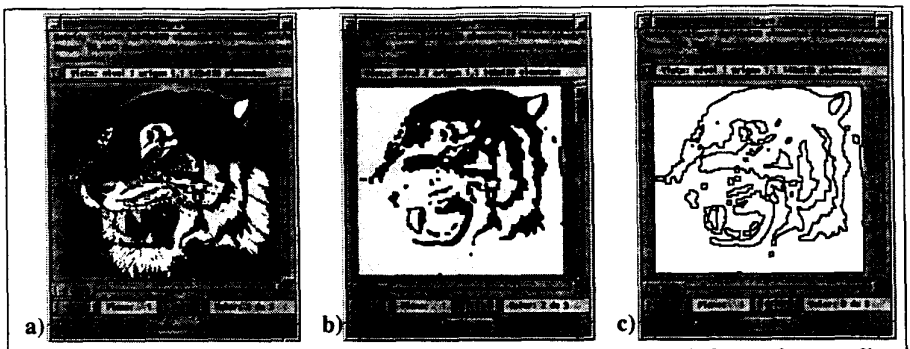

Figure 3 - Example of image-sequence visualization. a) Original grey-levels image. b) Image after noise filtering. c) Borders extracted from image b.

- Adding and deleting images from an image sequence.
- Drawing some common geometric elements (circles, ellipses, rectangles, etc.)

Fig. 3a illustrates the graphical appearance of this tool.

5.4. Simulator core interface (XINSS)

This tool is a graphical interface with the simulator core (NSS) of SIRENA under the X-windows system. Its main objective is to provide interactive access to the output generated by the simulator. This interactive access permits the visualization of the results of the simulation while it is running. In addition, the user is allowed to modify the simulation parameters without interrupting the simulator. In this manner, the user can observe and modify the simulation without reinitializing the process.

The major capabilities of this tool are:

- Starting a simulation on NSS.
- Simulation supervision by interactive observation of the output being generated.
- On-line parameter modification.
- Access to output files generated by previous simulations.
- Images visualization, individually or in sequences, calling XVMS.

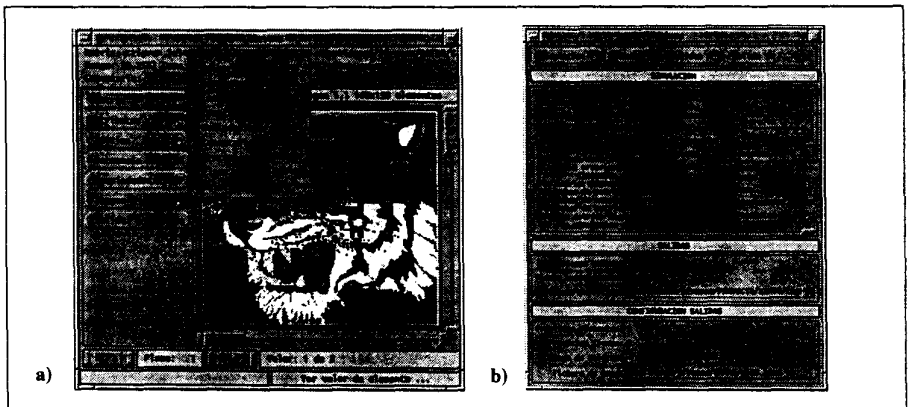

***Figure 4** - a) Matrix editor (XEVMS). b) Simulator core interface window (INSS).*

- Waveforms visualization using xgraph.
- Output storage on information-files.

6. Results

The efficiency of SIRENA is illustrated by the following example: the time required to simulate the borders extraction CNN process on the image shown in Fig. 3b (143x159 pixels) was of 52s, running on a Sparc Station 10 (100MIPS) and using the algorithmic level description.

References

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

[2] L.O. Chua and L. Yang: "Cellular Neural Networks: Applications". *IEEE Trans. Circuits and Systems*, Vol. 35, pp 1273-1290, October 1988.

[3] T. Roska, T. Boros, P. Thiran and L.O. Chua: "Detecting Simple Motion Using Cellular Neural Networks". *Proc. First IEEE Int. Workshop on Cellular Neural Networks and their Applications*, pp 127-138, Budapest, December 1990.

[4] T. Sziranyi and J. Csicsvari: "High-speed Character Recognition Using a Dual Cellular Neural Network Architecture". *IEEE Trans. Circuits and Systems II: Analog and Digital Signal Processing*, Vol. 40, pp 223-231, March 1993.

[5] H. Suzuki, T. Matsumoto and L.O. Chua: "A CNN Handwritten Character Recognizer". *Int. J. Circuit Theory and Applications*, Vol. 20, pp 601-612, John Wiley & Sons, September-October 1992.

[6] J. M. Cruz and L.O. Chua: "A CNN Chip for Connected Component Detection". *IEEE Trans. Circuits and Systems*. Vol. CAS-38, pp. 812-817, July 1991.

[7] H. Harrer, J. A. Nossek and R. Steltz: "An Analog Implementation of Discrete-Time Cellular Neural Networks". *IEEE Trans. Neural Networks*, Vol. 3, pp. 466-476, May 1992.

[8] A. Rodríguez-Vázquez, S. Espejo, R. Domínguez-Castro, J.L. Huertas and E. Sánchez-Sinencio: "Current-Mode Techniques for the Implementation of Continuous-Time and Discrete-Time Cellular Neural Networks", *IEEE Trans. Circuits and Systems II: Analog and Digital Signal Processing*, Vol. 40, pp. 132-146, March 1993.

[9] T. Roska, G. Bártfay, P. Szolgay, T. Szirányi, A. Radványi, T. Kozek and Zs. Ugray: "A Hardware Accelerator Board for Cellular Neural Networks: CNN-HAC", *IEEE Int. Workshop on Cellular Neural Networks and Their Applications*, pp. 160-168, Dec. 1990.

[10] "HSPICE User Manual" Meta Software Inc. 1988.

[11] R. Domínguez-Castro, S. Espejo, A. Rodríguez-Vázquez and R. Carmona: "A CNN Universal Chip in CMOS Technology". In this proceeding.

[12] L.O. Chua, T. Roska, P.L. Venetianer and A. Zarándy: "Some Novel Capabilities of CNN: Game of Life and Examples of Multipath Algorithms". *Proc. Second IEEE Int. Workshop on Cellular Neural Networks and their Applications*, pp 276-281, Munich, October 1992.