An electronic system for simulation of neural networks with a micro-second real time constraint

Arsenia Chorti, Bertrand Granado, Bruce Denby and Patrick Garda

Laboratoire des Instruments et Systèmes
Université Pierre et Marie Curie
Paris France

Abstract. Neural networks implemented in hardware can perform pattern recognition very quickly, and as such have been used to advantage in the triggering systems of certain high energy physics experiments. Typically, time constants of the order of a few microseconds are required. In this paper, we present a new system, MAHARADJA, for evaluating MLP and RBF neural network paradigms in real time. The system is tested on a possible ATLAS muon triggering application suggested by the Tel Aviv ATLAS group, consisting of a 4-8-8-4 MLP which must be evaluated in 10 microseconds. The inputs to the net are dx/dz, x(z=0), dy/dz, and y(z=0), whereas the outputs give pt, tan(phi), sin(theta), and q, the charge. With a 10 MHz clock, MAHARADJA calculates the result in 6.8 microseconds; at 20 MHz, which is readily attainable, this would be reduced to only 3.4 microseconds. The system can also handle RBF networks with 3 different distance metrics (Euclidean, Manhattan and Mahalanobis), and can simulate any MLP of 10 hidden layers or less. The electronic implementation is with FPGA's, which can be optimized for a specific neural network because the number of processing elements can be modified.

INTRODUCTION

Neural networks implemented in hardware can perform pattern recognition very quickly, and as such have been used to advantage in the triggering systems of certain high energy physics experiments. But the time constraint of such implementation is about few microseconds which is a very difficult. In our laboratory we have developped,MAHARADJA, an electronic system to simulate neural networks with real time simulation constraints. In this paper, we present the evaluation of this system to a possible ATLAS muon MLP triggering application suggested by the Tel Aviv ATLAS group with a time constraint of 10 microseconds.

MAHARADJA

MAHARADJA is realized to simulate two kind of neural networks models:

- Radial Basis Function Neural Networks
- Multi-Layer Perceptrons

In this article we only discuss the Multi-Layer Perceptrons implementation, the Radial Basis Function implementation is described in (1).

Our goal here is to know if our system can simulate High Energy Physics MLP with very difficult constraint. To investigate this question we benchmark MAHARADJA with a real network defined by Tel Aviv ATLAS group.

SIMULATED NETWORK

The simulated network is used in the L2 trigger and has 4 layers:

- a 4 neuron input layer and the inputs are dx/dz, x(z=0), dy/dz, and y(z=0)
- two 8 neuron hidden layers
- a 4 neuron output layer containing pt, tan(phi), sin(theta), and q, the charge

The required latency of this MLP is 10μs.

MAHARADJA DESCRIPTION

Our system is based on four principal components:

- **A sequential processor**: this processor can execute sequential part of the algorithm and manage the system.

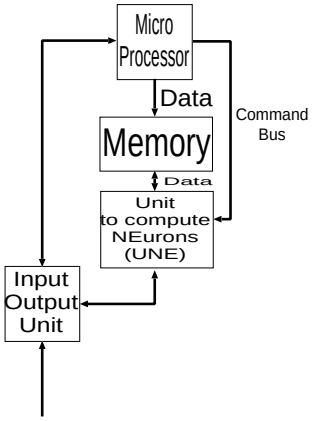

FIGURE 1. Architecture ofMAHARADJA

- **A Unit to compute NEurons (UNE)**: this unit accelerate the neuron computations to obtain very fast simulation.
- **An Input-Output unit**: this unit can provide a high input bandwidth to the UNE.
- **A shared memory**: this memory is shared between the UNE and the processor. It contains the neural network parameters, such as weights and size of layers.

ARCHITECTURE OF *UNE*

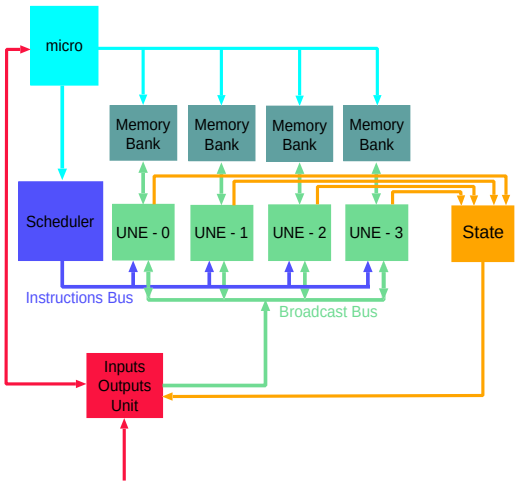

FIGURE 2. *UNE* unit organization

The *UNE* unit has two parts:

- four processors to compute post-synaptic potentials
- a component to compute the neuronal states

The processor of the *UNE* unit of MAHARADJA is organized in a SIMD [1] fashion. The interconnections are made by a 16-bit broadcast bus connected to the INPUT-OUTPUT unit.

A *Scheduler*, shown on Fig. 2, controls this unit by a 10-bit command bus.

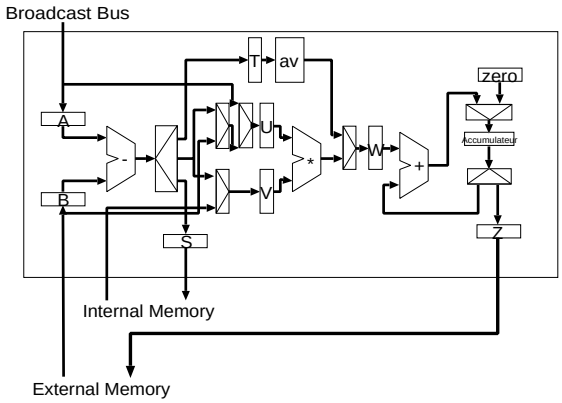

FIGURE 3. a processor of the *UNE* unit - A, B, S, T, U, V, W and U are registers that synchronize computation. - AV is the absolute value unit and ACCU the accumulation register.

Each processor of *UNE* unit compute one or more post-synaptic potentials as shown in Fig. 3.

The architecture of a *UNE* processor has

- a 16-bit subtractor to compute the first step of a distance computation in Radial Basis Functions.
- a 16-bit multiplier to compute the second step of Euclidean or Mahalanobis distance in Radial Basis Function or to compute the first step of Multi-Layer Perceptrons.
- a 16-bit absolute value unit to compute the second step of the Manhattan distance in Radial Basis Functions.
- a 32-bit adder to accumulate the result provided by the multiplier or the absolute value unit. This is the third step in computing distance in Radial Basis Functions or in computing Multi-Layer Perceptrons.

With these 4 operators, it is possible to compute all the post-synaptic potential for Radial Basis Functions with Manhattan, Euclidean or Mahalanobis distances and for Multi-Layer Perceptrons.

Behind the processor of the *UNE* unit there is a component to compute the neuron states. This computation is realized with a Lookup Table store in a memory.

[1] Single Instruction stream Multiple Data stream

UNE CONTROL

The *scheduler* can place the *UNE* unit in a functional mode. This mode can be:

- RBF with 3 differents metrics:
  - Manhattan distance
  - Euclidean distance
  - Mahalanobis distance
- MLP

This gives 4 modes. When a mode is choosen, it is impossible to switch to another in a dynamic way. To realize this, one must reinitialize the system.

The management of the *UNE* unit is realized by request. The beginning and the end of the computation is made by control signals, a *begin* signal and an *end* signal.

MEMORY

With each processor of the *UNE* unit, there is 256-KB of associated memory to store the simulated neural network parameters such as post-synaptic weights or sizes of layers.

INPUT-OUTPUT UNIT

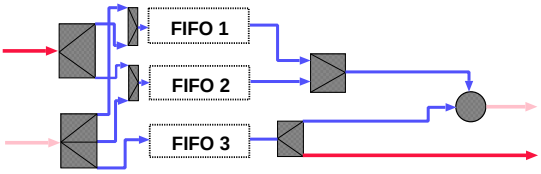

FIGURE 4. The input-output unit

This unit realizes the connection between MAHARADJA and the external world, for example the converter of the analog signals in the calorimeter of a collider. To obtain a high input bandwith we use a 3 FIFO structure, shown in Fig. 4. With such a structure the system makes a pipeline beetween the computation in the *UNE* unit and the acquisition of new data in the *Input-Output* unit.

In Fig. 5 we can see how the *Input-Unit* calculation for the HEP MLP is simulated. As we can see, the 2 fifo [2] (FIFO1 and FIFO2) are used to acquire new data from the external world. When the FIFO1 is used to store new

[2] first-in first-out

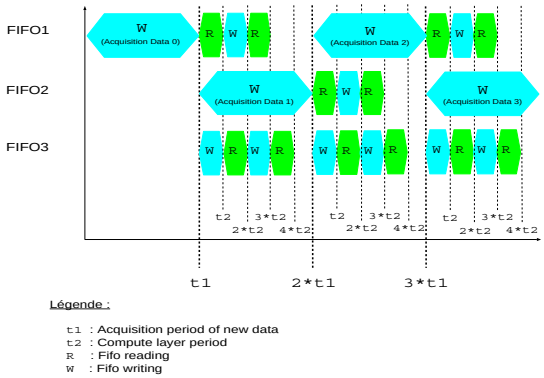

FIGURE 5. Use of the 3 FIFOS

data, FIFO2 is used for the computation and vice-versa. The third fifo (FIFO3) is used only for the computation and to store intermediate and final results.

IMPLEMENTATION

MAHARADJA is implemented with Altera APEX20K FPGA. We can with such an implementation modify some parameters like the number of processors to accelerate neural network computation.

TIMING ANALYSIS

We have carried out a time analysis of the MAHARADJA system, based on a prediction and evaluation methodology that we developed in our laboratory (2, 3). We first extract an analytical model of the evaluated system. This model is shown in table 1.

The variables in Table 1 are:

- *nbp* : Number of post-synaptic potentials (Terminal layer of the connection)
- *nbe* : Number of neurons (Initial layer of the connection)
- *N* : Neural Nework layer number

Now we can use the analytical model to predict the simulation time of the HEP neural network. This time is given in Table 2. Note that MAHARADJA can simulate this MLP with a simulation 1.5 times less than the desired 10μs if we use a 10 MHz clock and 4 processors in *UNE* unit, and it takes 6.5 times less if we use a 20 MHz clock and 8 processors in *UNE* unit.

Table 1. Time analysis of MAHARADJA

Multi Layer Perceptrons
$\sum_{i=1}^{N} \lceil \frac{nbe_i}{4} \rceil * (nbp_i + 2) + nbe_i + 1$
Manhattan Distance
$\left\lceil \frac{nbp}{4} \right\rceil * (nbe + 4) + \lceil \frac{nbe_s}{4} \rceil * (nbp_s + 2) + nbe_s + 1$
Euclidean Distance
$\left\lceil \frac{nbp}{4} \right\rceil * (nbe + 4) + \lceil \frac{nbe_s}{4} \rceil * (nbp_s + 2) + nbe_s + 1$
Mahalanobis Distance
$\left\lceil \frac{nbp}{4} \right\rceil * (nbe^2 + 6 * nbe + 2) + \lceil \frac{nbe_s}{4} \rceil * (nbp_s + 2) + nbe_s + 1$

Table 2. Comparison of Simulation Time for an MLP Predicted by the Analytical Model.

System	Frequence (MHz)	Time (μs)
MAHARADJA 4 UNE	10	6.5
MAHARADJA 8 UNE	10	3.2
MAHARADJA 8 UNE	20	1.6

CONCLUSION

In this article we propose an electronic system , MAHARADJA, which can calculate the result of a HEP MLP in 6.5 microseconds at 10 MHz, or at 20 MHz, 3.4 microseconds. The system can also handle RBF networks with 3 different distance metrics (Euclidean, Manhattan and Mahalanobis), and can simulate any MLP of 10 hidden layers or less. The electronic implementation is with FPGA's, which can be optimized for a specific neural network because the number of processing elements can be modified.

REFERENCES

1. Granado, B., *Architecture des systèmes électroniques pour les réseaux de neurones - Conception d'une rétine connexionniste*, Ph.D. thesis, Université Paris XI, November 1998.
2. Granado, B., and Garda, P., in *Proceedings of ICANN'96*, Juillet 1996 .
3. Granado, B., and Garda, P., in *Proceedings of IWANN'97*, Lanzarote - Canary Islands, Spain, June 1997 .