IWINAC2007

2nd. INTERNATIONAL WORK-CONFERENCE on the INTERPLAY between NATURAL and ARTIFICIAL COMPUTATION

HANNA: A Tool for Hardware Prototyping and Benchmarking of ANNs

Javier Garrigós, José J. Martínez, Javier Toledo and José M. Ferrández

Dpto. Electrónica, Tecnología de Computadoras y Proyectos - Universidad Politécnica de Cartagena

C/. Doctor Fleming s/n, 30202 Cartagena - Spain

1. Introduction

For some applications, designers must implement an ANN model over different platforms to meet performance, cost or power constrains, a process still more painful when several hardware implementations have to be evaluated. Continuous advances in VLSI technologies, computer architecture and software development make it difficult to find the adequate implementation platform. HANNA (Hardware ANN Architect), is a tool designed to automate the generation of hardware prototypes of MLP-like neural networks over FPGA devices. Coupled with traditional Matlab/Simulink environments the model can be synthesized, downloaded to the FPGA and co-simulated with the software version to trade off area, speed and precision requirements.

2. Software/Hardware prototyping

System Generator™ for DSP (Xilinx Inc.), a plug-in to the Simulink® environment (The Mathworks Inc.) designed to develop high-performance DSP systems for FPGAs. Advantages are:

- high-level schematic flow,
- powerful Matlab visualization tools for test-benches,
- successive refined software versions developed with reduced additional effort, each one introducing more hardware friendly characteristics, to observe the effect of latency, quantization, etc.
- hardware in the loop capability. Hardware version of the ANN can be developed and validated against the previous versions.

As an example, Figure 1 demonstrate implementation and co-simulation of three different versions of the XOR function: using the standard logical operator (green), a software full-precision ANN (blue), and its hardware –quantized- version (orange).

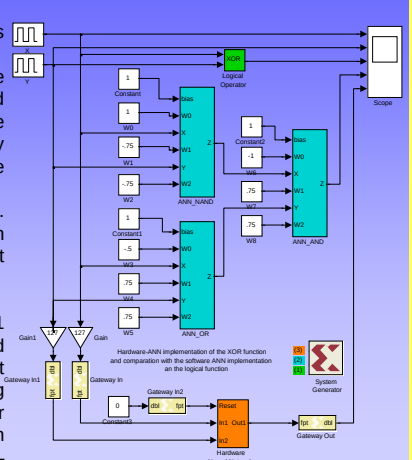


Fig.1. Net Architecture (right.

3. HANNA capabilities

However, using System Generator blockset, the hardware model has to be defined at the RTL, which requires certain knowledge of digital and logic design and the effort of generating a detailed description of the ANN algorithm. HANNA is intended to automatically generate a synthesizable System Generator description starting from just a high level definition of the network. It takes then minutes to have the hardware version downloaded and working on a FPGA prototyping board. Moreover, software and hardware outputs can be simultaneously verified with the same, unique, test bench.

HANNA is a set of Matlab scripts, primitives templates and a GUI. Starting from user's specifications and proper templates, VHDL for the net, control and neuron blocks is generated. After network generation, a Simulink model file is created which contains the network architecture, and a second file with same name and sufix _vars is generated to store workspace and general configuration parameters. The model generated includes a synthesis block configured for the custom tools and hardware available for circuit implementation, shown in Figure 3.(a).

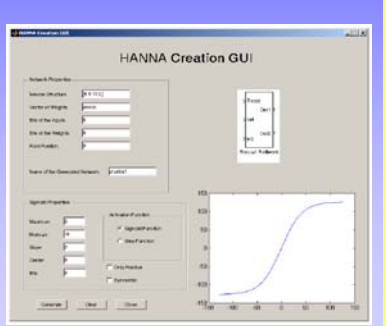
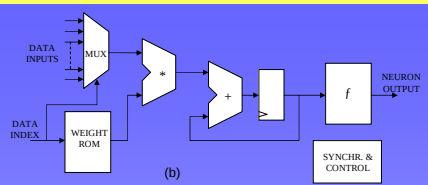


Fig.2. HANNA GUI showing some network parameters.

3. Proposed ANN Hardware Architecture

$$y = f(net) = f\left(\sum_{i=0}^{n} w_i x_i + b\right)$$

(a)



(b)

Fig.3. (a) Standard neuron transfer function for MLPs, (b) Neuron architecture, defined as highly parametrizable VHDL.

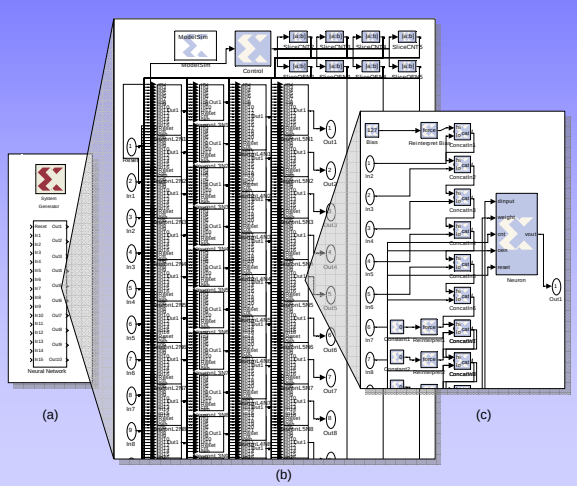


Fig.4. Net Architecture. (a) Net block (bottom) and vhdl synthesis block (top), (b) neuron layers (bottom) and control logic (top), (c) input casting (left) and neuron block (right).

5. Results for XOR and speaker recognition applications[1].

Results showed 500x/800x performance boost over Matlab (P4 3.2GHz, 2GB RAM).

Table 1. Implementation and performance results obtained for the two benchmark applications

Parameter	XOR	Speaker Rec. 1	Speaker Rec. 2
I/O quantiz. (bits)	8	10	8
Weight quantiz. (bits)	8	10	8
Sigmoid quantiz. (bits)	8	10	8
Net structure[1]	[2,2,1]	[12,20,11]	[12,20,11]
FPGA Device (Xilinx)	s2s200e-6	xc2v1000-4	xc4vsx55-12
Clock rate (MHz)	46.762	37.115	60.22
Latency (clock cycles)	8	36	36
Throughput (Msa/s)	5.845	1.031	1.67
Slices (%)	8	82	2

[1]Individually differentiating every member in a group of 11 spanish speakers.

6. Conclusions

HANNA is a tool for hardware prototyping of MLP-like ANNs over programmable devices. The benefits offered by system level design tools, like the suite Matlab/Simulink/SystemGenerator that allows targeting multiple implementation platforms, can be further extended with tools like HANNA in case of FPGA prototyping, allowing a high level (neuron level) description of the ANN, instead of the standard lower-level RTL description of the network. The tool and our design methodology have been validated through two examples, resulting in hardware design times reduced to minutes and acceptable performance over current FPGA devices.

Future work involves increasing the capabilities of HANNA by including learning mechanisms, a microprocessor-based network controller for sophisticated execution schemes, and taking advantage of FPGAs dynamic reconfiguration features for on-line network evolution. HANNA is available upon request for non-profit research institutions and organizations.

This research is being funded by Ministerio de Ciencia y Tecnología TIC 2003-09557-C02-02.