

High Level Design Approach for FPGA Implementation of ANNs

Nouma Izeboudjen

Center de Développement des Technologies Avancées (CDTA), Algérie

Ahcene Farah

Ajman University, UAE

Hamid Bessalah

Center de Développement des Technologies Avancées (CDTA), Algérie

Ahmed Bouridene

Queens University of Belfast, UK

Nassim Chikhi

Center de Développement des Technologies Avancées (CDTA), Algérie

INTRODUCTION

Artificial neural networks (ANNs) are systems which are derived from the field of neuroscience and are characterized by intensive arithmetic operations. These networks display interesting features such as parallelism, classification, optimization, adaptation, generalization and associative memories. Since the McCulloch and Pitts pioneering work (McCulloch, W.S., & Pitts, W. (1943), there has been much discussion on the topic of ANNs implementation, and a huge diversity of ANNs has been designed (C. Lindsey & T. Lindblad, 1994). The benefits of using such implementations is well discussed in a paper by R. Lippmann (Richard P. Lippmann, 1984): “The great interest of building neural networks remains in the high speed processing that can be achieved through massively parallel implementation”. In another paper Clark S. Lindsey (C.S Lindsey, Th. Lindbald, 1995) posed a real dilemma of hardware implementation: “Built a general, but probably expensive system that can be reprogrammed for several kinds of tasks like CNAPS for example? Or build a specialized chip to do one thing but very quickly, like the IBM ZISC Processor”. To overcome this dilemma, most researchers agree that an ideal solution should relay the performances obtained using specific hardware implementation and the flexibility allowed by software tools and general purpose chips.

Since their commercial introduction in the mid-1980's, and due to the advances in the development

of both of the microelectronic technology and the specific CAD tools, FPGAs devices have progressed in an evolutionary and revolutionary way. The evolution process has allowed faster and bigger FPGAs, better CAD tools and better technical support. The revolution process concerns the introduction of high performances multipliers, Microprocessors and DSP functions. This has a direct incidence to FPGA implementation of ANNs and a lot of research has been carried to investigate the use of FPGAs in ANNs implementation (Amos R. Omandi & Jagath C. rajapakse, 2006).

Another attractive key feature of FPGAs is their flexibility, which can be obtained at different levels: exploitation of the programmability of FPGA, dynamic reconfiguration or run time reconfiguration (RTR), (Xilinx XAPP290, 2004) and the application of the **design for reuse** concept (Keating, Michael; Bricaud, Pierre, 2002).

However, a big disadvantage of FPGAs is the low level hardware oriented programming model needed to fully exploit the FPGA's potential performances.

High level based VHDL synthesis tools have been proposed to bridge the gap between the high level application requirements and the low level FPGA hardware but these tools are not algorithmic or application specific. Thus, special concepts need to be developed for automatic ANN implementation before using synthesis tools.

In this paper, we present a high level design methodology for ANN implementation that attempts to build a

bridge between the synthesis tool and the ANN design requirements. This method offers a high flexibility in the design while achieving speed/area performances constraints. The three implementation figures of the ANN based back propagation algorithm are considered. These are the off-type implementation, the on-chip global implementation and the dynamic reconfiguration choices of the ANN.

To achieve our goal, a design for reuse strategy has been applied. To validate our approach, three case studies are considered using the Virtex-II and Virtex-4 FPGA devices. A comparative study is done and new conclusions are given.

BACKGROUND

In this section, theoretical presentation of the **multilayer perceptron** (MLP) based back propagation algorithm is given. Then, discussion of the most related works to the topics of high level design methodology and ANNs frameworks are given.

Theoretical Background of the Back Propagation Algorithm

The back propagation is one of the well known algorithms that are used to train the MLP ANN network in a supervised mode. The MLP is executed in three phases: the *feed forward phase*, the *error calculation phase* and the *synaptic weight updating phase* (Freeman, J. A. and Skapura, D. M, 1991).

In the *feed forward phase*, a pattern x_i is applied to the input layer and the resulting signal is forward propagated through the network until the final outputs have been calculated; for each i (index of neuron) and j (index of layer)

$$\mu_j^l = \sum_{i=1}^{n_0} w_{ij}^l x_i \quad (1)$$

$$o_j^l = f(x_j) = \frac{1}{1 + \exp(-\mu_j)} \quad (2)$$

where, μ_j^l is the weighted sum of the synaptic weights and o_j^l is the output of the sigmoid activation function.

The *error calculation step*, computes the local error, δ for each layer starting from output back to input:

$$\delta_i^L = f'(u_i^L)(d_i - y_i) \quad (3)$$

$$\delta_j^{l-1} = f'(u_j^{l-1}) \sum_{i=1}^{N_l} w_{ij} \delta_i^l \quad 1 \leq i \leq N_l, \quad 1 \leq l \leq L \quad (4)$$

where, d_i is the desired output f' the derivative function of f

The *Weight update step* computes the weights updates according to:

$$w_{ij}^l(t+1) = w_{ij}^l(t) + \Delta w_{ij}^l(t) \quad (5)$$

$$\Delta w_{ij}^l(t) = \eta \delta_i^l y_j^{l-1} \quad (6)$$

where, η is the learning factor, Δw the variation of weights and l , the indices of the layers.

Background on ANN Frameworks

The most related works to ANNs frameworks are presented by (F. Schurmann & all, 2002), (M. Diepenhorst & all, 1999), and (J. Zhu & all, 1999).

In the other hand, and with the increasing complexity of FPGAs **circuits**, Core-based synthesis methodology is proposed as a new trend for efficient hardware implementation of FPGAs. In these tools a library of pre-designed IPs “Intellectual Property” cores are proposed. An example can be found in (Xilinx Core Generator reference) and (Opencores reference).

In the core based design methodology, efficient reuse is derived from the parameterized design with VHDL and its many flexible constructs and characteristics (i.e. abstraction, encapsulation, inheritance and reuse through attributes, package, procedures and functions). Beside this, the reuse concept is well suited for high regular and repetitive structures such as neural networks. However although all these advantages, seldom attention has been done to apply **design for reuse** for ANNs.

In this context our paper presents a new high level design methodology based upon the use of the **design for reuse** concept for ANNs.

In order to achieve this goal, the design must fulfill the following requirements (Keating, Michael; Bricaud, Pierre, 2002):

- The design must be block-based

7 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage: www.igi-global.com/chapter/high-level-design-approach-fpga/10340

Related Content

Enhancing Earthquake Prediction With Reinforcement Learning

Lalitha S. D., Madijagan M., Rajakumari S. and Manikandan R. (2024). *Novel AI Applications for Advancing Earth Sciences* (pp. 273-290).

www.irma-international.org/chapter/enhancing-earthquake-prediction-with-reinforcement-learning/336224

Application of Direct and Indirect Human-Centered Design Techniques With Dyslexic Users

Dominik Hagelkruys and Renate Motschnig (2019). *Handbook of Research on Human-Computer Interfaces and New Modes of Interactivity* (pp. 1-20).

www.irma-international.org/chapter/application-of-direct-and-indirect-human-centered-design-techniques-with-dyslexic-users/228515

Towards Intelligent Requirements

Robert B.K. Brown, Angela M.E. Piper and Ian C. Piper (2015). *International Journal of Intelligent Information Technologies* (pp. 1-11).

www.irma-international.org/article/towards-intelligent-requirements/128836

Game AGI beyond Characters

Alexander Zook (2017). *Artificial Intelligence: Concepts, Methodologies, Tools, and Applications* (pp. 463-484).

www.irma-international.org/chapter/game-agi-beyond-characters/173348

Thesaurus with Predicate-Argument Structure to Provide Base Framework to Determine States, Actions, and Change-of-States

Koichi Takeuchi (2016). *Computational and Cognitive Approaches to Narratology* (pp. 192-212).

www.irma-international.org/chapter/thesaurus-with-predicate-argument-structure-to-provide-base-framework-to-determine-states-actions-and-change-of-states/159626