Available online at www.sciencedirect.com

ScienceDirect

Procedia Technology 17 (2014) 736 – 741

Procedia Technology

Conference on Electronics, Telecommunications and Computers – CETC 2013

A systolic architecture for Hopfield neural networks

Hajar Asgari[a], Yousef S. Kavian[a,*], Ali Mahani[b]

[a]*Faculty of Engineering, Shahid Chamran University of Ahvaz, Ahvaz, Iran*
[b]*Faculty of Engineering, Shahid Bahonar University of Kerman, Kerman, Iran*

Abstract

Recently the Hopfield Neural Network (HNN) is employed as an optimization tool to solve shortest path problem in communication networks. The hardware implementation of digital Hopfield neural network is an important issue that is considered in this paper. An efficient systolic architecture is proposed for efficient implementing of digital Hopfield neural networks for solving shortest path problem on Field-Programmable-Gate-Array (FPGA) chips. The VHDL hardware description language is employed for hardware modeling of proposed systolic architecture. The results achieved from simulation and hardware synthesizing demonstrates that the proposed systolic architecture has superior performance than relevant architectures in literature for chip area utilization, convergence and maximum operating frequency.

Peer-review under responsibility of ISEL – Instituto Superior de Engenharia de Lisboa, Lisbon, PORTUGAL.

Keywords: Systolic architecture; Hopfield neural network; shortest path problem; FPGA; VHDL

1. Introduction

The Hopfield neural networks [1] have been successfully employed for practical issues such as modelling applications, solving NP-hard optimization problems like travelling salesman problem and solving routing problems in wireless and optical networks [2,3]. A typical Hopfield neural network consists of some processing units called neurons where all neurons process their inputs simultaneously and parallel structure such as Field-Programmable-Gate-Array (FPGA) chips is considered as implementing environment for digital Hopfield neural networks. There are some researches on FPGA implementation of neural networks for different applications [4,5] but a few research on FPGA implementation of Hopfield neural network for solving shortest path problem. In this paper a study is done on FPGA implementation of Hopfield neural network for solving shortest path problem by use of mathematical

* Corresponding author. Tel.: +98-611-3330010; fax: +98-611-3330010.
E-mail address: y.s.kavian@scu.ac.ir

Peer-review under responsibility of ISEL – Instituto Superior de Engenharia de Lisboa, Lisbon, PORTUGAL.
doi:10.1016/j.protcy.2014.10.195

concepts and such techniques like systolic architecture is proposed for efficient implementing of digital Hopfield neural networks for solving shortest path problem.
The rest of paper is organized as follows: section 2 presents a brief background of the Hopfield neural network models for solving shortest path problem. Section 3 describes the proposed systolic architecture of the network. Hardware description and results are explained in section 4 and finally paper is concluded in section 5.

2. Mathematical model for digital implementing

The network topology is represented as a finite, directed, and weighted graph $G(N,L,W)$ where N is the number of nodes, L is the set of connecting links in the network and W is the weight of links. The path $P_{s,d}$ is considered as a sequence of nodes from source node s to destination node d connected by links. For hardware description of HNN on FPGA for solving shortest path problem, the digital model of neurons should be employed and therefore the neuron dynamical model is discretized. The discrete model of neuron is as follows:

$$X_{ij}(k) = X_{ij}(k-1) + \Delta t\left[\sum_{k=1}^{N}\sum_{l=1}^{N} W_{ijkl}Y_{kl}(k-1) + I_{ij}\right] \tag{1}$$

where $X_{ij}(k)$ is the internal state or local voltage of neuron and $Y_{kl}(t)$ is the output voltage of neuron (i,j) in the network and $i \neq j$ means there is no neuron on diagonal situation. Each neuron has a sigmoid function and neuron has output voltage value between 0 and 1. In this model, N is the number of neurons in each row and column, I_{ij} is the neurons bias value at location (i,j) and W_{ijkl} is the network weight matrix which is a 4-dimensional symmetric array.

The main blocks for hardware implementing of proposed neuron model in (1) are REGISTERS for saving the neuron values, the ADDERS and MULTIPLIERS for mathematical operations and some combinational and sequential control units for providing controlling signals.

3. Systolic architecture

The main drawback of conventional architectures for implementing HNN is the use of many multipliers and for solving shortest path problem in large-scale communication networks on FPGA chips due to resources limitations; those architectures are not useful. There are some techniques such as systolic array that can be used to optimize hardware utilization of digital chips.

The systolic array includes a set of cells that are arranged by special arrangement and all of them process different data simultaneously. In this structure data flow between each cells and its neighbour and after some stage when all cells process their inputs, operation is completed. For direct multiplication of two 3×3 matrixes on FPGA, many multipliers are required and the hardware chip area is not suitable but when the multiplication is done by systolic array technique as shown in Fig. 1, by 9 cells and after 7 stages the multiplication operation is done. As in this system each cell is composed of only one multiplier and an accumulator so resource utilization of this system is less comparatively. In this system each cell gets data from top and left sides, multiplies them and accumulates its result in a register then transfers its input data to down and right sides. This example shows that by use of systolic array technique in an application, the amount of required multipliers is reduced significantly. Therefore the systolic array concepts in neurons and HNN systems are employed to reduce the resource utilization of HNN setup for solving shortest path problem.