Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Mixed-signal VLSI neural network based on Continuous Valued Number System

Babak Zamanlooy*, Mitra Mirhassani

Department of Electrical and Computer Engineering, University of Windsor, Windsor, ON, Canada

ARTICLE INFO

Article history:
Received 30 December 2015
Received in revised form
17 July 2016
Accepted 28 August 2016
Communicated by Prof. R.W. Newcomb

Keywords:
Mixed signal neural networks
VLSI implementation
Continuous Valued Number System
Analog digits

ABSTRACT

Mixed-signal neural networks have higher energy efficiency and lower area consumption when compared with their equivalent digital implementations. However, the signal processing precision in mixed-signal implementations is limited. A mixed-signal arithmetic method called Continuous Valued Number Systems (CVNS) is employed for development and implementation of different functions required in a neuronal network. Analog digits of this number system enable the higher accuracy of analog operations, and can efficiently relate with digital and binary values. In this paper, design and implementation of a 2-2-1 mixed-signal CVNS network structure is presented, to confirm the developed arithmetic method. In the proposed structure, weights are stored in the digital registers while the arithmetic is based on the CVNS. The CVNS features have been exploited to address the limited signal processing precision issue, which requires resolutions higher than 12-bit for online training of the network. As a result, the proposed structure satisfies this requirement of neural networks, while yielding an optimized network configuration. The proposed network is designed, fabricated, and tested in 0.18 μm technology.

1. Introduction

The ASIC implementation of neural networks has been exploited in various applications such as optical template matching [1], biomedical applications [2], and image classification [3].

There are several categories of hardware implementation of neural network which includes analog [4,5], digital [6–9] or mixed-signal. In the analog neural networks, both signal processing and synapse weights are implemented by using analog circuits. When implemented in analog, neural networks typically possess a higher energy efficiency and require less area in comparison with the digital implementations. However, storage of synapse weights using the capacitor-based method in analog designs requires constant refreshing, and its values are susceptible to leakage current, and process and power supply variations [10].

The mixed-signal networks utilize digital registers for the storage of synapse values while signal processing is based on the analog circuits. This method benefits from the ease of weight storage in digital registers while capitalizing on the advantages of the analog domain such as compact implementations of adders and nonlinear neurons. Therefore, the mixed-signal method is, in general, provides an optimized method for the hardware implementation of neural networks.

However, one of the main obstacles that the mixed-signal approach faces is the low accuracy of the analog circuits. The accuracy of the analog circuits is referred to as the environment resolution in the rest of this paper. To overcome this limitation, and at the same time benefiting from the analog circuits, the Continuous Valued Number System (CVNS) can be utilized.

In this number system, the system designer can set the information represented by each CVNS analog digit to be equal to the environment resolution. However, collectively a set of analog digits can preserve the accuracy even if implemented by analog circuits. This feature makes the CVNS appropriate for performing precise analog and mixed-signal operations [11–13].

The most basic operation in the network is addition operation, therefore in this work, CVNS addition is presented considering the low resolution of analog circuits. This algorithm is then employed in the multiplier and other required functions of the network. In the end, the proposed low-resolution arithmetic is used to implement a prototype network. The proposed structure a 2-2-1 network, and is used to test its operations for the classical nonlinearly separable XOR problem to verify the proof of concept. The proposed network is designed and fabricated in the available 180 nm technology. The performance of the network is confirmed by the experimental results.

This paper is organized as follows. A brief introduction to the CVNS is presented in Section 2. A new CVNS addition algorithm for low-resolution environment is introduced in Section 3. Afterward,

* Corresponding author.
E-mail addresses: zamanlo@uwindsor.ca (B. Zamanlooy), mitramir@uwindsor.ca (M. Mirhassani).

http://dx.doi.org/10.1016/j.neucom.2016.08.090
0925-2312/© 2016 Elsevier B.V. All rights reserved.

the proposed structure for the 2-2-1 CVNS network and its VLSI implementation are explained in Section 4. Experimental results are discussed in Section 5. Finally, conclusions are drawn in Section 6.

2. Continuous Valued Number System (CVNS)

The absolute value of a real number such as X, using the fixed-point number representation with a radix of B can be shown as follows:

$$X = \sum_{i=-N_f}^{N_i-1} x_i \times B^i \tag{1}$$

where N_i and N_f are the number of integer and fractional for each x_i digit.

Analog digits are used to represent the number in Continuous Valued Number System (CVNS). The digits representing the real number X are generated using the modular reduction operation as follows [14]:

$$((x))_i = \left(X \times B^{-i}\right) \bmod B \tag{2}$$

where $((x))_i$ is the CVNS digit and $-N_f \leq i \leq N_i - 1$.

A set of CVNS analog digits $((X))_{\equiv}\left\{((x))_{N_i-1}, \ldots, ((x))_0, ((x))_{-1}, \ldots, ((x))_{-N_f}\right\}$ represents the real number X in CVNS format.

Example: Finding the CVNS digit set of $X = 60.72$ *with a radix of 10.*

According to (1) two integer and fractional digits, N_i and N_f, are required for representation of $X = 60.72$. Thus, to represent X, the following CVNS digit ensemble using the Eq. (2) can be used:

$$\left(\left(X\right)\right)_{\equiv}\left\{((x))_1, ((x))_0, ((x))_{-1}, ((x))_{-2}\right\} = \{6.072, 0.72, 7.2, 2\}$$

3. CVNS addition in a low resolution environment

In this section, a new CVNS addition algorithm for the low-resolution environment is presented. The proposed CVNS addition algorithm is exploited in the next sections to implement the CVNS neural network structure. Since the CVNS neural network presented in this paper is implemented using CVNS numbers with a radix of two, the proposed addition algorithm is proven for the same CVNS radix. However, it can be utilized for CVNS numbers with an arbitrary choice of radix.

The CVNS analog digits can be implemented using the Eq. (2). However, this representation requires a high resolution environment. Therefore, to match the requirement of the CVNS to reliability of the environment resolution truncated analog digits can be used as as follows:

$$\left(\left(X\right)\right)_m = \sum_{i=m-\varphi+1}^{m} x_i \times 2^{i-m} \tag{3}$$

where φ is the environment resolution.

For example, CVNS digit set of $X = 1010.1111$ for a limited environment resolution of four $(\varphi = 4)$, can be found using the relation in (3) as follows:

$$\left(\left(X\right)\right)_{\equiv}\{1.25, 0.625, 1.375, 0.875|1.875, 1.625, 1.5, 1\}$$

If φ is chosen large and near the maximum required digits $(N_i + N_F)$, the outcome is going to be very similar to the original digit generation scheme as presented in (2). This choice defeats the purpose of applying the digit truncation.

On the other hand, a small φ introduces error and removes the overlap between CVNS digits. Applying the digit truncation only means that the binary digits with weights less than $2^{\varphi-1}$ are ignored.

Any radix-2 CVNS value is converted back to binary digits as follows:

$$x_m = \begin{cases} 1 & \left(\left(X\right)\right)_m \geq 1 \\ 0 & \left(\left(X\right)\right)_m < 1 \end{cases} \tag{4}$$

For example binary digits of $\left(\left(X\right)\right)_{\equiv}\{1.5, 1.125, 0.25, 0.625, 1.25, 0.5, 1, 0\}$ can be mapped back using the relation in (4) to binary number $X = 11001010$.

To perform an addition operation on two CVNS values based on the relation presented by (2), CVNS digits of the summation result, $\left(\left(Z\right)\right)_m$, are obtained as follows:

$$\begin{aligned} \left(\left(Z\right)\right)_m &= \left(\left(X + Y\right) \times 2^{-m}\right) \bmod 2 \\ &= \left(\sum_{i=-N_f}^{N_i-1} \left(x_i + y_i\right) \times 2^{i-m}\right) \bmod 2 \end{aligned} \tag{5}$$

The relation in (5) can be written as summation of two terms in the following form:

$$\begin{aligned} \left(\left(Z\right)\right)_m = &\left(\sum_{i=-N_f}^{m} \left(x_i + y_i\right) \times 2^{i-m}\right) \bmod 2 \\ &+ \left(\sum_{i=m+1}^{N_i-1} \left(x_i + y_i\right) \times 2^{i-m}\right) \bmod 2 \end{aligned} \tag{6}$$

Considering that all the terms in the second summation are even values and are greater than or equal to two, applying a mod 2 operation on those terms results in zero. Therefore, (6) can be written in the following form:

$$\left(\left(Z\right)\right)_m = \left(\sum_{i=-N_f}^{m} \left(x_i + y_i\right) \times 2^{i-m}\right) \bmod 2 \tag{7}$$

To follow up with the implementation limitations and to reduce the demand on each CVNS digit, digits can be truncated.

By applying the truncation method the $\left(\left(Z\right)\right)_m$ can be written as the summation of multiple terms, where each term has only φ terms as presented in the following form:

$$\begin{aligned} \left(\left(Z\right)\right)_m = &\left(\sum_{i=m-\varphi+1}^{m} \left(x_i + y_i\right) \times 2^{i-m} + 2^{-\varphi} \sum_{i=m-2\varphi+1}^{m-\varphi} \left(x_i + y_i\right) \times 2^{i-(m-\varphi)}\right. \\ &\left. + \ldots + 2^{-(n_m-1)\varphi} \sum_{i=-N_f}^{m-n_m\varphi} \left(x_i + y_i\right) \times 2^{i-(m-n_m\varphi)}\right) \bmod 2 \end{aligned} \tag{8}$$

One of most fundamental properties of the CVNS is that the digits have information overlap with each other. This means that a higher index digit can be constructed partially from lower index digits. Therefore, a lower index digit of the summation, $\left(\left(Z\right)\right)_{m-\varphi}$, by repeating the same process can be written as follows:

$$\begin{aligned} \left(\left(z\right)\right)_{m-\varphi} = C_{m-\varphi} \bmod 2 = &\left(\sum_{i=m-2\varphi+1}^{m-\varphi} \left(x_i + y_i\right) \times 2^{i-(m-\varphi)}\right. \\ &\left. + \ldots + 2^{-(n_m-1)\varphi} \sum_{i=-N_f}^{m-n_m\varphi} \left(x_i + y_i\right) \times 2^{i-(m-n_m\varphi)}\right) \bmod 2 \end{aligned} \tag{9}$$

Therefore a higher index digit can be described using the relation in (9). Therefore, the relation in (8) can be modified as follows:

Please cite this article as: B. Zamanlooy, M. Mirhassani, Mixed-signal VLSI neural network based on Continuous Valued Number System, Neurocomputing (2016), http://dx.doi.org/10.1016/j.neucom.2016.08.090