An On-Line Arithmetic-Based Reconfigurable Neuroprocessor

Jean-Luc Beuchat and Eduardo Sanchez

Logic Systems Laboratory, Swiss Federal Institute of Technology,
CH – 1015 Lausanne, Switzerland
E-mail: {name.surname}@di.epfl.ch

Abstract. Artificial neural networks can solve complex problems such as time series prediction, handwritten pattern recognition or speech processing. Though software simulations are essential when one sets about to study a new algorithm, they cannot always fulfill real-time criteria required by some practical applications. Consequently, hardware implementations are of crucial import. The appearance of fast reconfigurable FPGA circuits brings about new paths for the design of neuroprocessors. All arithmetic operations are carried out with on-line operators. This short paper briefly describes reconfigurable FPGA-based neural networks and gives an introduction to on-line arithmetic.

1 Introduction

The brain is a "computer" which exhibits characteristics such as robustness, distributed memory and calculation, interpretation of imprecise or noisy sensorial information. The design of electronic devices demonstrating such characteristics would be very interesting from the engineering point of view. Applications of these circuits include artificial vision, autonomous robotics or speech processing. Artificial neural networks constitute a possible way to realize such devices. The appearance of fast reconfigurable Field-Programmable Gate Arrays (FPGA) offers new paths for neuroprocessor implementation. However, FPGAs are too small to implement a large fully connected neural network and its learning rule. We exploit two philosophies to solve this problem:

- All arithmetic operations are carried out with on-line operators. Section 2 describes the advantages of this methodology.
- A neural algorithm is split into several steps executed sequentially, each of which is associated with a specific FPGA configuration. Such an approach leads to an optimal use of hardware resources. The efficiency of this method has been proven by J. G. Eldredge [1]. The reconfiguration paradigm also allows the implementation of multiple algorithms on the same hardware [2].

2 On-line Arithmetic

A neural network contains numerous connections, thus requiring a substantial amount of wires. As parallel arithmetic requires large buses, it is not well suited

for such implementations. Bit-serial communications seem to be an effective solution.

Bit-serial data transmission may begin with the least significant bit (LSB) or with the most significant bit (MSB). Though the LSB paradigm seems more natural (the digits of the result are generated from right to left when we carry out an addition or a multiplication using the "paper and pencil" method), it does not allow the design of algorithms for division or square root. In general, the implementation of complex serial operations makes use of LSB and MSB operators. Let us consider the example depicted by figure 1. All digits of $(a + b) + (c \cdot d)$ must be known before the computation of the square root.

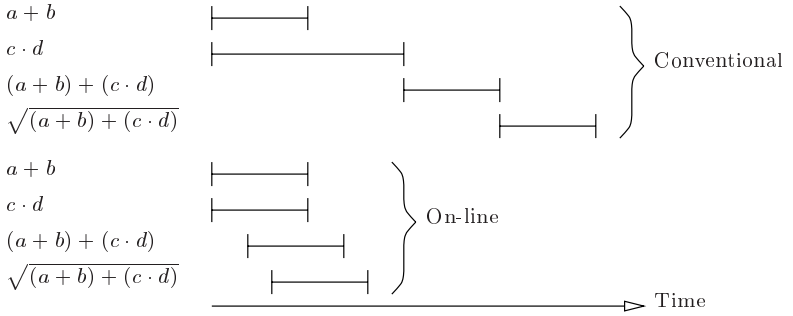

Fig. 1. *Computing* $\sqrt{(a + b) + (c \cdot d)}$.

The on-line mode, introduced in [3], *performs all computations in MSB mode thanks to a redundant number representation*[1]. It allows to overlap computation, leading so to a lower processing time (figure 1).

Usually, a number $a \in \mathbb{R}$ is written in radix r as $\sum_{i=-\infty}^{\infty} a_i r^{-i}$, where $a_i \in \mathcal{D}_r = \{0, 1, \ldots, r-1\}$. $\mathcal{D}_r$ is the digit set. On-line algorithms described in the literature use Avizienis' signed-digit systems [4], where numbers are represented in radix r with digits belonging to $\mathcal{D}_r = \{-a, -a+1, \ldots, a-1, a\}$, $a \leq r - 1$.

Our neuroprocessor exploits on-line arithmetic in radix 2, i.e. the digit set is $\{-1, 0, 1\}$. It uses a bit-level representation of the digits, called *Borrow-Save*, defined as follows: two bits, a_i^+ and a_i^-, represent the ith digit of a, such that $a_i = a_i^+ - a_i^-$. Consequently, digit 1 is encoded by $(1, 0)$, digit -1 is encoded by $(0, 1)$, while digit 0 is represented by $(0, 0)$ or $(1, 1)$.

Many on-line operators are described in the literature. Each of them is characterized by a delay δ, which indicates the number of clock cycles required to compute the MSB of the result. Figure 2a depicts the schedule diagram of an operator of delay 2. An input signal i is provided at time $t = 0$. Two clock cycles are required to elaborate the MSB of the output signal o. A new bit of the result is then produced at each clock cycle.

[1] In a redundant system, additions can be performed without carry propagation. This property allows the design of MSB adders.

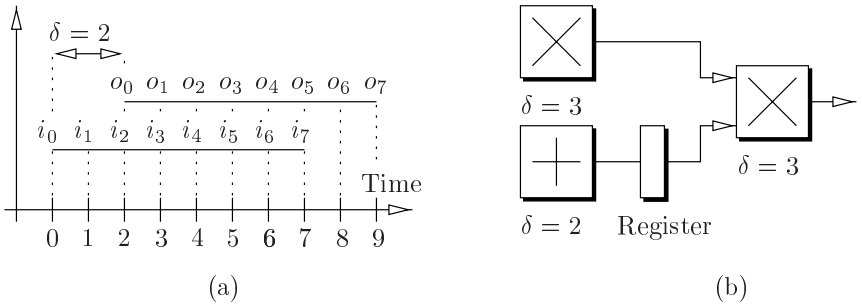

Fig. 2. *(a) Delay of an on-line operator. (b) Pipeline with on-line operators.*

Suppose now that the result of a first operator Γ is needed for further computations. The second operation may begin as soon as Γ has generated the MSB of the partial result. Figure 2b shows how to chain different on-line operators. The addition and the first multiplication are carried out in parallel. As the delay of the adder is smaller than the one of the multiplier, we use a register to synchronize the inputs of the second multiplier.

3 Conclusions

We have exploited the reconfiguration principles and on-line arithmetic to implement a neuroprocessor with on-chip training. The reconfigurable approach allows an optimal use of hardware resources and on-line arithmetic leads to a space-efficient implementation of data transmission. This approach is effective to implement large neural networks on a single FPGA, which is for instance intersting for autonomous robotic applications.

References

1. J.G. Eldredge and B.L. Hutchings. Density Enhancement of a Neural Network Using FPGAs and Run-Time Reconfiguration. In Duncan A. Buell and Kenneth L. Pocek, editors, *Proceedings of the IEEE Workshop on FPGAs for Custom Computing Machines*, pages 180–188. IEEE Computer Society Press, 1994.
2. Jean-Luc Beuchat and Eduardo Sanchez. A Reconfigurable Neuroprocessor with On-chip Pruning. In L. Niklasson, M. Bodén, and T. Ziemke, editors, *ICANN 98*, Perspectives in Neural Computing, pages 1159–1164. Springer, 1998.
3. Kishor S. Trivedi and Milos D. Ercegovac. On-line Algorithms for Division and Multiplication. *IEEE Transaction on Computers*, C-26(7), July 1977.
4. Algirdas Avizienis. Signed-Digit Number Representations for Fast Parallel Arithmetic. *IRE Transactions on Electronic Computers*, 10, 1961.