Neurocomputing 205 (2016) 32–44

Contents lists available at ScienceDirect

Neurocomputing

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

An efficient method of error correction in fault-tolerant modular neurocomputers

N.I. Chervyakov [a], P.A. Lyakhov [a,*], M.G. Babenko [a], A.I. Garyanina [a], I.N. Lavrinenko [b], A.V. Lavrinenko [a], M.A. Deryabin [a]

[a] Department of Applied Mathematics and Mathematical Modeling, North-Caucasus Federal University, Stavropol, Russia
[b] Department of Higher Algebra and Geometry, North-Caucasus Federal University, Stavropol, Russia

ARTICLE INFO

Article history:
Received 29 April 2015
Received in revised form
6 March 2016
Accepted 16 March 2016
Prof. R. W. Newcomb
Available online 17 May 2016

Keywords:
Residue number system
Chinese Remainder Theorem
Mixed Radix Conversion
Fault-tolerance
Artificial neural network

ABSTRACT

In this paper, we propose the architecture of a fault-tolerant unit in a modular neurocomputer that is based on decoding with computation of errors syndromes on redundant moduli and implemented using FPGA and a finite ring neural network. The computational complexity of the proposed architecture is about 80% less in comparison with that of the architecture based on number projections in the mixed radix number system.

1. Introduction

One of the main parameters in fault-tolerant modular neurocomputers design is the reliability of their functioning. On the one hand, the continuously growing requirements to the speed regulation characteristics of computing devices call for parallel computing organization; on the other hand, the frequency of failures increases, and the down time of processors caused by difficult search and elimination of failures raises accordingly. Obviously, irrespective of the characteristics demonstrated by a computing device, a single error in any of its units can disconnect or damage the whole system, generating a fatal failure in certain cases [1–5]. The high reliability of information transfer and processing is an especially topical problem in modern real-time systems, where equipment errors must be detected and corrected immediately. Note that transition to the latest submicron technologies only aggravates this problem, since the complexity of Integration Circuit (IC) manufacturing increases manyfold, thereby raising the probability of failures [3,4]. Thus, high reliability in this case should be achieved not so much by enhancing the technical means of information transmission and processing, but by applying information coding methods that would be stable against possible random distortions, allowing data correction if necessary. In this context, the most promising solution of the considered problem is making computing devices fault-tolerant [6–8]. A computing system is considered to be fault-tolerant if, in case of failure occurrence, it saves the functional capabilities fully (fail-safe) or partially (fail-soft). Thus, fault-tolerance is guaranteed by combining system redundancy with an error trapping mechanism and automatic recovery procedures for its correct functioning. A fail-safe computing system is able to ensure correct functioning despite failure occurrence, but with quality losses (i.e. in the state of gradual degradation [4,5]).

General analysis of the high-reliability problem of neuroproccessors identifies three most widespread methods of fault-tolerance assurance, namely, hardware reservation, application of special positional codes, and application of arithmetical correction codes.

In case of instrumental redundancy, reserve modules can be activated only if faulty modules are replaced.

When an error occurs at the output of a module, it is detected, localized and then fixed using correction codes or repeated calculation. Alternatively, a failure is removed by replacing a faulty module or performing system reconfiguration. Thus, system self-recovery after faults and hardware failures is provided. This method has high redundancy.

Another way of fault-tolerance assurance consists in application of special positional codes widely used in communication channels. Such codes are intended to find and correct random

[*] Corresponding author.
E-mail address: ljahov@mail.ru (P.A. Lyakhov).

http://dx.doi.org/10.1016/j.neucom.2016.03.041
0925-2312/© 2016 Elsevier B.V. All rights reserved.

errors arising in the course of information storage or transfer; however, it is impossible to monitor the correctness of arithmetical operations in these codes [1,3]. This nonarithmetic property of special positional codes hinders with their application in computing systems, as check digits do not control the result of an arithmetical operation. Yet, such monitoring is no less important for a computing system than information transfer monitoring.

Arithmetical codes, i.e. non-positional codes in modular arithmetic, possess much more ample correction capabilities. Redundant modular arithmetic or a redundant residue number system (RRNS) enjoys the unique properties of error detection and correction [1–5]. Firstly, the order of digits in number designation is not significant in such a system. And secondly, both codes and checked numbers are represented in the form of residuals, which makes such codes completely arithmetical. Proceeding from these properties, one can conclude that application of modular arithmetic serves for efficient creation of fault-tolerant systems, being a powerful tool for automatic detection and correction of errors.

RRNS attracts many researchers as a basis for designing fault-tolerant computing structures. In the last decade, investigations in this field have been intensified dramatically, as evidenced by a large number of publications on RRNS usage in digital communication systems [9], parallel computing environments [10], global communication systems [11], wireless networks [12], fault-tolerant hybrid memory structures [13] and other areas.

The papers [9,10,12] analyzed the properties of error correction codes in residue number systems (RNS) based on an evaluation of the Hamming distance of a code. The authors of [9,12] employed correction methods with the extensions of RNS bases. Each of the cited works proposed an original modification of the base extension method, but the general principles remained the same. In [9], MRC was used as a method of range extension. In Refs. [10,13], error localization involved various versions of the projection method and the Chinese Remainder Theorem. The projection method consists in analyzing the value of a number in the truncated systems of RNS bases. In [13], the projection method was combined with special sets of moduli; as a result, the method proposed by the authors is efficient for a series of applications. In [11], error localization and correction was performed using the interval method. Direct analysis shows that the above works suggest calculation methods associated with RNS and MRC, yielding time-consuming computational procedures.

2. Bases of redundant residue number system

The development of high-performance and reliable computing systems enjoying fault tolerance proceeds from the ideas of parallel-structure computing means design using parallel data representation and handling. Among such means, we mention position-independent codes (codes based on modular arithmetic), i.e. codes with data representation in RNS, see [9–13].

A fixed series of positive numbers $p_1,\ p_2,\ \ldots,\ p_n$ is said to be the bases (moduli) of RNS. Then a residue number system is a non-positional number system, where any positive integer A from Weighted Number System (WNS) is represented as a set of remainders (residues) in division of this number by the selected bases of the system:

$$A = (\alpha_1,\quad \alpha_2,\quad \ldots,\quad \alpha_n). \tag{1}$$

Here α_i specify the smallest nonnegative residues (remainders) of the number with respect to the moduli $p_1,\ p_2,\ \ldots,\ p_n$. The digits α_i in this representation using the selected moduli have the form

$$a_i = A \ (\mathrm{mod}\ p_i) = A - \left[\frac{A}{p_i}\right] p_i, \quad (\forall\ \ i \in [1,\ \ n]), \tag{2}$$

where $\left[\frac{A}{p_i}\right]$ is the integer quotient, p_i mean the bases (moduli) being coprimes. The following result was established in number theory. If $\forall\ \ i \ne j\ \ :\ \ \left(p_i,\ \ p_j\right) = 1$, then by the Chinese Remainder Theorem (CRT) the representation (1) is unique under the condition $0 \le A \le P$, where $P = p_1 p_2 \cdots p_n = \prod_{i=1}^{n} p_i$ gives the number representation range. In other words, there exists a number A such that

$$\begin{aligned} &A \equiv \alpha_1\ (\mathrm{mod}\ p_1);\\ &A \equiv \alpha_2\ (\mathrm{mod}\ p_2);\\ &\cdots\quad \cdots\quad \cdots\quad \cdots\\ &A \equiv \alpha_n\ (\mathrm{mod}\ p_n). \end{aligned} \tag{3}$$

The main advantage of this representation is simple execution of addition, subtraction and multiplication by the formula

$$\begin{aligned} A * B &= (\alpha_1, \alpha_2, \ldots, \alpha_n) * (\beta_1, \beta_2, \ldots, \beta_n) = \\ &= ((\alpha_1 * \beta_1)\ \mathrm{mod}\ p_1, (\alpha_2 * \beta_2)\ \mathrm{mod}\ p_2, \ldots, (\alpha_n * \beta_n)\ \mathrm{mod}\ p_n), \end{aligned} \tag{4}$$

where * signifies an appropriate operation (addition, multiplication or subtraction). These operations are called modular, as (a) their execution in RNS requires merely one clock period of numerical values processing and (b) this processing runs in parallel and the value of a number in each digit is independent from other digits.

The RNS codes considered below can be used for correction of errors arising in information transfer or execution of arithmetical operations.

Adding the digit a_{n+1} to the representation (1) yields

$$A = (\alpha_1,\quad \alpha_2,\quad \ldots,\quad \alpha_{n+1}), \tag{5}$$

which is associated with some number from the range $[0,\quad R)$, where $R = P\,p_{n+1}$. In case of information transfer or execution of any operations, we will use the same representations (1) associated with the numbers belonging to some part of the range $[0,\quad R)$. This allows error correction.

If the representation (5) satisfies the condition

$$0 \le A < \frac{R}{p_{n+1}} = P, \tag{6}$$

then the number A is uniquely defined by the n-bit representation $(\alpha_1,\quad \alpha_2,\quad \ldots,\quad \alpha_{n+1})$. Therefore, the bit with the base p_{n+1} can be considered redundant. Let us estimate the correction capabilities of RNS codes having one and two redundant bits with the bases p_{n+1} and p_{n+2}.

Introduce the concept of an error. A single error is comprehended as the distortion of a certain digit in the representation (5), and distortion is restricted only to the value of the base. A k-tuple error is comprehended as the distortion of k digits in the representation (5).

Let $A = (\alpha_1,\quad \alpha_2,\quad \ldots,\quad \alpha_n,\quad \alpha_{n+1})$ be a sent message and denote by $\overline{A} = (\overline{\alpha_1},\quad \overline{\alpha_2},\quad \ldots,\quad \overline{\alpha_n},\quad \overline{\alpha_{n+1}})$ the received message. The message is said to be faultless if $0 \le \overline{A} < P_{n+1}$. If $\overline{A} \ge P_{n+1}$, then the corresponding representation is called faulty.

If the message (5) is transferred, then generally any of R messages can be received. Thus, P_{n+1} messages among them are received as faultless, and $(R - P_{n+1})$ ones as faulty; and so, the quantity $(R - P_{n+1})$ shows the total number of possible errors detected. The relations $\frac{R - p_{n+1}}{R} = \frac{p_{n+1} - 1}{p_{n+1}}$ can serve as a measure of the error detection capabilities of the RNS code. In case of arranged RNS (the bases placed in the ascending order of their value, i.e. $p_1 < p_2 < \ldots < p_{n+1}$), the numbers $P_1,\ \ P_2,\ \ \ldots,\ \ P_{n+1}$ satisfy the inequalities $P_1 > P_2 > \ldots > P_{n+1}$. According to [1–4], the existence of one redundant base p_{n+1} is sufficient for detecting all single errors. In other words, if $\overline{\alpha_1} = \alpha_i$ for $i \ne k$ and $\overline{\alpha_1} \ne \alpha_i$ $(i = 1,\ \ 2,\quad \ldots,\quad n+1)$, then $\overline{A} \ge P_{n+1}$, which detects an error. Note that, by introducing one check base, it is possible to detect not only any single error (in a digit with one base), but also 95% of double ones (in digits with two bases) [1,4].