Neurocomputing 000 (2017) 1–12

Contents lists available at ScienceDirect

Neurocomputing

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

The architecture of a fault-tolerant modular neurocomputer based on modular number projections

N.I. Chervyakov[a], P.A. Lyakhov[a,*], M.G. Babenko[a], I.N. Lavrinenko[b], A.V. Lavrinenko[a], A.S. Nazarov[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 8 January 2017
Revised 28 April 2017
Accepted 25 June 2017
Available online xxx

Communicated by Prof. R. W. Newcomb

Keywords:
Modular neurocomputer
Residue number system
Fault-tolerance
Digital neuron
Finite ring neural network
Field-programmable gate array

ABSTRACT

This paper suggests a rather efficient architecture for an error correction unit of a residue number system (RNS) that is based on a redundant RNS (RRNS) and applied in parallel data processing structures owing to its capability to improve information stability in calculations. However, the high efficiency of error correction is still not achieved due to the need in the expensive and complex operators that require substantial computational resources and considerable execution time. The suggested error correction method employs the Chinese remainder theorem (CRT) and artificial neural networks (ANN) that appreciably simplify the process of error detection, localization and correction. The key components of the error correction procedure are optimized using (a) the mixed radix conversion (MRC), i.e., the parallel conversion of the numbers from an RNS into the mixed radix number system (MRNS), and (b) the adaptation of neural networks to different sets of RNS moduli (bases) and also to the modular arithmetic during the computation of modular number projections and the restoration of the correct residue on a faulty module. Therefore, the expensive topological structures of neural networks are replaced with the reconfiguration using the weight coefficients switching. In comparison with the existing CRT-based method of projection calculation, the suggested method yields a 20%–30% reduction in power consumption, yet requiring by 10%–20% less FPGA resources for implementation.

1. Introduction

Nowadays the reliability of modular neurocomputing systems forms a challenging problem due to the growing complexity of their structure and functions. The reliability improvement problem is burning for the neurocomputers embedded in aerospace and airborne systems, cloud computing, security systems, digital closed circuit systems, control systems for crucial processes and objects, and so on [1–7].

High reliability must be achieved by further development of the element base of computing devices but, in the first place, by using coding methods with stability against random failures (or distortions) and dynamic data correction in the course of calculations if necessary.

Over the past few years the researchers have studied the detection and localization of faulty digits in modular computers [1–5]. However, the existing error correction approaches do not use residue number systems (RNS) together with artificial neural networks (ANN), which have high efficiency. A separate error in any element of a computing structure may cause an incorrect result of calculations. And so, it seems fruitful to design methods for fast error detection, localization and correction that guarantee a correct result. This feature is vital for the systems with high-integrity components such as Field-Programmable Gate Arrays (FPGAs) that are widespread in modern computing system design.

Modern computational tasks more and more require parallel data processing, also increasing the demand for multidigit computing devices. In this field of research, a promising approach is to apply residue number systems [1–5], which have powerful potential of high-speed and reliable data processing consistent with the parallel computing base in form of artificial neural networks.

Of special interest are redundant residue number systems (RRNSs) subject to error detection and correction [2]. There exist many publications considering RRNSs as a tool to restore an integer from a set of its remainders [3–7]. Error detection and

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

http://dx.doi.org/10.1016/j.neucom.2017.06.063
0925-2312/

correction codes involve such notions as admissible (legal) range and inadmissible (illegal) range, which are used to check numbers [3].

It is possible to distinguish between two main methods of error correction in a redundant residue number code. The first method calculates an error syndrome obtained by extending RNS bases (or moduli) and then compares them with a set of expected results in order to restore a correct number. Similar algorithms were studied in [2,5]. The papers [8–10] analyzed single errors correction with different modifications of the error syndrome method (still, the general principles remain the same). Next, the papers [10–12] were focused on multiple errors correction. In particular, error syndromes in [12] were calculated by a special algorithm requiring a complex computational scheme and time-consuming calculations.

The second method calculates errors from the remainders of RNS numbers using the Chinese Remainder Theorem [13–16]. For example, the algorithm suggested in [13] performs single errors correction based on the general idea of the projection method. The paper [14] considered a multiple errors correction algorithm by extending the approach from [13]. The authors [15] presented other algorithms using the same strategy.

The multiple errors correction method developed in [14] proceeds from the CRT. Original numbers are found using an iterative process that defines a faulty combination of modular number remainders within an admissible range. This method has restrictions due to the modification of the maximum probability decoding algorithm and use of the Garner algorithm, which increases time delays and hardware costs. Besides, there is no analytical expression for the number of iterations required to define an original number, while experimental calculations may yield a wrong number of checked situations that have to be rechecked.

Although the cited researchers propose certain modifications, the general principles remain the same. Direct analysis shows that all methods employed actually calculate the values of numbers according to the CRT or MRNS algorithms, thereby being computationally intensive.

In the current paper, we present a new error correction scheme based on the second method. This scheme resembles the ones given in [13–16]. However, the new scheme is simpler owing to the optimized projection calculation and error correction using an appropriate reconfiguration of artificial neural networks.

Ideologically, this algorithm is simple in the sense of application. Moreover, the theory and principles of the suggested error correction scheme can be used for its further improvement in terms of efficiency (i.e., less intensive calculations).

2. Corrective capabilities of residue number system codes

Consider the geometrical model of an RNS code, which is based on a set of unit n-dimensional cubes. Each RNS representation of a number that forms code combinations is associated with the vertexes of the unit n-dimensional cubes whose code points

$$\begin{aligned} &A_0\Big(A_0 \bmod p_1,\, A_0 \bmod p_2,\, \dots,\, A_0 \bmod p_n\Big)\\ &A_1\Big(A_1 \bmod p_1,\, A_1 \bmod p_2,\, \dots,\, A_1 \bmod p_n\Big)\\ &\qquad\qquad\dots\\ &A_p\Big(A_p \bmod p_1,\, A_p \bmod p_2,\, \dots,\, A_p \bmod p_n\Big) \end{aligned} \tag{1}$$

form $P_n = p_1 p_2 \dots p_n$ vertexes.

The number of the n-dimensional cubes is $N = (p_1 - 1)(p_2 - 1)\dots(p_n - 1)$.

Let the surface of the unit n-dimensional cubes have the property that any two vertexes on it can be connected by finite-length lines. Then for each pair of vertexes A_i and A_j on the surface of the n-dimensional cubes there exists a lower bound for the lengths of the lines lying on this surface and connecting these points. The lower bound of the lengths is the distance between the points A_i and A_j, further denoted by $d_{A_iA_j}$. The distance defined in this way satisfies the three basic laws of a metric, i.e., $d_{A_iA_j} \geq 0$, $d_{A_iA_j} = d_{A_jA_i}$, and $d_{A_iA_j} = d_{A_iA_k} + d_{A_kA_j}$.

Each set P_n of RNS complexes in which for any pair of its elements (points) A_i and A_j there is a well-defined number $d_{A_iA_j}$ satisfying the three basic laws of a metric is called a metric space. In this case, the function $d_{A_iA_j}$ is called the metric of this space, and its value for a certain pair of points (vertexes) of the unit n-dimensional cubes is called the distance between these points.

In the introduced metric, one error modifies one coordinate of the code point, two errors two coordinates, … , and k errors k coordinates. A single-base error changes one coordinate by the value Δ_i (the depth of this error). The distance between the two vertexes A_i and A_j of the unit n-dimensional cubes can be defined as the least number of edges that have to be passed while moving from the vertex A_i to the vertex A_j. In terms of distance, the adjacent edges that define the depth of a single-base error are treated as one edge. Note that it is possible to move along the edges of a given n-dimensional cube and also along the edges of the neighbor cubes (right, left, top, and bottom).

Best visualization is provided by the geometrical model for the RNS codes with three bases $p_1 = 2$, $p_2 = 3$, and $p_3 = 5$, where p_1 and p_2 denote informational (working) bases and p_3 acts as a check base. Then the working range is $P_2 = p_1 p_2 = 6$, and the complete range is $P_3 = p_1 p_2 p_3 = 30$. In this case, a number A is uniquely defined by a representation (α_1, α_2), and the digit α_3 can be considered redundant. The input alphabet consists of P_3 symbols $(0, 1, \dots, 29)$. Then the word length of the RNS code that represents and transfers all symbols of this alphabet is $A(\alpha_1, \alpha_2, \alpha_3)$, where $\alpha_1 = 0, 1$; $\alpha_2 = 0, 1, 2$; $\alpha_3 = 0, 1, 2, 3, 4$.

The depth of the single-base errors is $\Delta_1 = 1$ (for the first base) or may vary from $\Delta_2 = 1$ to $\Delta_2 = 2$ (for the second base) and from $\Delta_3 = 1$ to $\Delta_3 = 4$ (for the third base). Each symbol is represented by three decimal digits. The values of the digits belong to the range $0 \leq \alpha_i \leq p_i - 1$, where $i = 1, 2, 3$.

Assume that data processing employs binary codes. Then the binary code used in the representation and processing of each digit has the digit capacity

$$m_i = \lfloor \log_2 p_i \rfloor + 1, \tag{2}$$

where $\lfloor X \rfloor$ indicates the largest integer less than X.

In our example, $m_1 = 1$, $m_2 = 2$, and $m_3 = 3$.

Thus, the first, second and third digits of RNS are represented using one, two, and three binary digits, respectively. With binary codes, any number is represented by six binary digits. In this case, each of represented numbers can be identified with a vertex of the unit cube in the space whose points have the coordinates $A_i \bmod p_1$, $A_i \bmod p_2$, and $A_i \bmod p_3$. This geometrical model of an RNS code is illustrated by Fig. 1, with coordinates specified for each vertex.

Axis x has one unit segment corresponding to the module p_1; axis y two unit segments corresponding to the module p_2; finally, axis z four unit segments corresponding to the module p_3.

The distance between different code combinations (vectors) varies from 1 to n, where n denotes the number of moduli (in our example, $n = 3$). For instance, the distances between the points are $d_{A_0A_{10}} = 1$, $d_{A_0A_{25}} = 2$, and $d_{A_0A_1} = 3$. The distance between neighbor numbers that differ by 1 is n. This follows from the count-up or countdown rule. In the case of count-up, transition from a given number to the nearest large number is performed by replacing the existing digit with the closest admissible digit; for this position