ROMANIAN JOURNAL OF INFORMATION SCIENCE AND TECHNOLOGY
Volume **20**, Number 2, 2017, 161–176

The Significant Bits Propagation Model in Fault-Tolerant System Design

Vladimir CIRIC[1], Ivan MILENTIJEVIC[1], and Aleksandar CVETKOVIC[2]

[1]University of Niš, Faculty of Electronic Engineering, Computer Science Department, Aleksandra Medvedeva 14, P.O.Box 73, Niš, Serbia
[2]University of Belgrade, Faculty of Mechanical Engineering, Department for Mathematics, Kraljice Marije 16, Belgrade, Serbia
E-mail: vladimir.ciric@elfak.ni.ac.rs, ivan.milentijevic@elfak.ni.ac.rs, acvetkovic@mas.bg.ac.rs

Abstract. With current VLSI technology approaching the scale of individual atoms, and nanotechnology fabrication processes such as self-assembly, the resulting structures become less predictable and less prone to fabrication defects. Fortunately, there are a lot of technical applications that doesn't require 100% correctness of the output. This fact can be used to increase the fabrication yield and reduce unnecessary hardware overhead. For example, in some DSP applications such as audio or video processing, correctness of the operation can be defined as a threshold up to which a user will not notice the error. The threshold is usually measured as a quantity of correct information in the output, i.e. the number of significant digits. In this paper we propose a novel technique for mathematical modeling of significant digits propagation under the presence of errors. The model incorporates Euclidean metric, and it can be used to obtain the difference in the number of significant digits between any two given intermediate results within the system. The information about loss of significant digits, due to the propagation of error through the architecture, will be used to design a partially fault tolerant system. The model is based on Min-Plus algebra, which is chosen to formalize structural dependencies. The evaluation of the proposed model is performed on the example of semi-systolic bit-plane array for FIR filtering, with error defined by Euclidean metric. It is shown that proposed model leads to the significant reduction of unavoidable fault-tolerant hardware overhead.

1. Introduction

The density of the transistors that can be placed on the single integrated circuit doubles approximately every two years, as Moore's law predicted half a century ago [1]. Integrated circuits nowadays contain up to 10^{10} devices, while slowly reaching lithography technological limits [2]. Current VLSI feature sizes go below 20nm, approaching the scale of individual atoms. The demand for precision to image such a small devices has increased the cost of lithography, while the resulting structures become less predictable. Expectations are that the chips in nanotechnology will be able to integrate 10^{12}, or even more devices [3, 4].

There are two basic approaches in nanoscale devices manufacturing: top-down or bottom-up [5]. Top-down fabrication reduces large pieces of materials all the way down to the nanoscale, while the bottom-up approach creates products by building them up from atomic- and molecular-scale components. Variety of processes include techniques from standard (nanoscale) lithography to biologically inspired self-assembly [5]. Regardless the technology or fabrication process, defects and variations coupled with stochastic assembly will likely to persist on every chip [4]. If not handled carefully, increasing process variations, defect rates, infant mortality rates, and susceptibility to internal and external noises are likely to decrease functional yield. Having in mind the number of devices per chip in both current VLSI and nanotechnology, the magnitude of the problem of modeling error propagation through chip becomes significant [6].

There are a lot of technical applications that doesn't require 100% correctness of the output. For example, in some DSP applications such as audio or video processing, correctness of the operation can be defined as an error threshold up to which a user will not notice the error. Applying fault tolerance techniques only to the part of the system where eventual defects might cause significant errors can improve overall fabrication yield [7]. One of the proposed methods is Partial Defect Tolerance (PDT), which, in essence, aims to obtain the size and the position of the most important part of the system, and to make it fault tolerant [8]. In order to determine the most important part of the system, the intensive calculations have to be performed and error propagation through the system needs to be analyzed [7].

The design of the PDT system strongly depends on the chosen metric between correct and erroneous result, i.e. on the definition of error [7, 9]. If the error is modeled using Hamming's metric, it is assumed that the output contains errors if any bit differs from the correct result. Modeling using Hamming's metric is illustrated in [8]. For some applications, such as multimedia systems, Euclidean distance is more suitable for error modeling [8]. However, models that include Euclidean metric are more difficult to develop and handle in the case of complex systems [8, 9]. To the best of our knowledge, there is no suitable model, based on Euclidean metric, for the analysis of significant digits propagation through the system represented by the data flow graph.

In this paper we propose a novel technique for mathematical modeling of significant digits propagation through the system under the presence of errors. The model incorporates Euclidean metric, and it can be used to obtain the difference in the number of significant digits between any two given intermediate results within the system. The information about loss of significant digits, due to the propagation of error through the architecture, will be used to design a partially fault tolerant system. Min-Plus algebra will be chosen to formalize structural dependencies within the system [10]. It will be shown that Min-Plus algebra can be used to simplify error-propagation modeling, which is required for system's partitioning in PDT design [8]. The evaluation of the proposed model will be illustrated on the example of semi-systolic bit-plane array for FIR filtering, with error defined by Euclidean distance. It will be shown that proposed model leads to the significant reduction of unavoidable fault-tolerant hardware overhead.

The paper is organized as follows. Section 2 gives a brief introduction to semi-systolic bit-plane FIR filtering and PDT method. Section 3 is devoted to the tropical algebra, as a basis for the mathematical model of significant digits propagation. Section 4 is the main section and presents the proposed significant digits propagation model in formal mathematical manner. Section 5 is devoted to model evaluation on the example of PDT semi-systolic bit-plane FIR filtering array, while in Section 6 concluding remarks are given.

2. Semi-systolic bit-plane FIR filtering array and partial defect tolerance

The model of significant bits propagation will be developed on the example of bit-plane array for FIR filtering. With aim to clarify the development of the model, we give a brief review of the bit-plane FIR filtering array (BPA).

Output words $\{y_i\}$ of FIR filter are computed as

$$y_i = c_0 x_i + c_1 x_{i-1} + \ldots + c_{k-1} x_{i-k+1}, \tag{1}$$

where c_0, c_1,..., c_{k-1} are coefficients while $\{x_i\}$ are input words. Computation (1) can be realized in different manners. Systolic arrays are good candidates when high performances are required. Semi-systolic arrays share with systolic arrays simplicity, regularity, and pipelining, while they introduce slight irregularities, such as broadcast lines. The BPA is a semi-systolic architecture with bit-plane operations [11, 12].

The following notation is adopted: $\boldsymbol{m}$ – coefficient word length; $\boldsymbol{k_C}$ - number of coefficients $(c_0, c_1,\ldots, c_{k_C-1})$; $\boldsymbol{n}$ - input word length; $\boldsymbol{c_i^j}$ - bit of coefficient c_i (with weight 2^j); $\boldsymbol{c_i} \equiv c_i^{m-1} c_i^{m-2} \cdots c_i^0$, where $c_i^0 c_i^1 \ldots c_i^{m-1}$ are the bits of coefficient c_i with weights $2^0, 2^1, \ldots, 2^{m-1}$, respectively; $\boldsymbol{c^j} \equiv c_{k-1}^j c_{k-2}^j \ldots c_0^j$, where $c_0^j, c_1^j, \ldots, c_{k-1}^j$, are the bits with weight 2^j of coefficients $c_0, c_1,\ldots, c_{k-1}$, respectively; $\boldsymbol{l_0}$ - the number of basic cells within one row of a BPA; $\boldsymbol{y_i^j}$ - the bit of output word y_i with weight 2^j;

Let us consider the example for $k_C = 3$ and $m = 2$. Eq. (1) can be decomposed into $m = 2$ bit-planes in the following manner:

$$\begin{aligned} y_i &= \underbrace{\left(c_0^1 2^1 + c_0^0 2^0\right)}_{c_0} x_i + \underbrace{\left(c_1^1 2^1 + c_1^0 2^0\right)}_{c_1} x_{i-1} + \underbrace{\left(c_2^1 2^1 + c_2^0 2^0\right)}_{c_2} x_{i-2} = \\ &= \underbrace{\left(c_0^0 x_i + c_1^0 x_{i-1} + c_2^0 x_{i-2}\right)}_{\text{bit plane 0}} 2^0 + \underbrace{\left(c_0^1 x_i + c_1^1 x_{i-1} + c_2^1 x_{i-2}\right)}_{\text{bit plane 1}} 2^1. \end{aligned} \tag{2}$$

The k-th bit-plane multiplies the input words $\{x_i\}$ and all the bits c_i^k with the same weight 2^k from all coefficients $c_i, i = 0, 1, \ldots, k_C - 1$. The transfer function of the bit-plane FIR filter, which captures the execution order of the particular operations from (2) in a pipeline, can be represented in the Z-domain as

$$\begin{aligned} G(z) = \frac{y(z)}{x(z)} &= z^{-1}(c_0^1 2^1 z^{-3} + z^{-1}(c_1^1 2^1 z^{-3} + z^{-1}(c_2^1 2^1 z^{-3} + \\ &+ z^{-1}(c_0^0 2^1 + z^{-1}(c_1^0 2^1 + z^{-1}(c_2^0 2^1))), \end{aligned} \tag{3}$$

where z^{-i} represents i delays in time. A BPA with $k_C = 3$ and $m = 2$ that corresponds to the equations (2) and (3) is shown in Fig. 1. There are $m = 2$ bit-plane elements that form the array shown in Fig. 1a, with basic cells shown in Fig. 1b. Each bit-plane (Fig. 1) is formed as a set of k_C rows. A row performs the basic multiply-accumulate operation between the intermediate result from the previous row and the product of the input word and one coefficient bit. Delays in Fig. 1a are represented with small circles. Delayed for one clock cycle per row, the output word is available after $k_C \cdot m$ clock cycles.

The BPA is semi-regular structure because of the broadcast line that feeds the input words $\{x_i\}$ into each row of one bit-plane at the same time (Fig. 1). Furthermore, there are irregularities

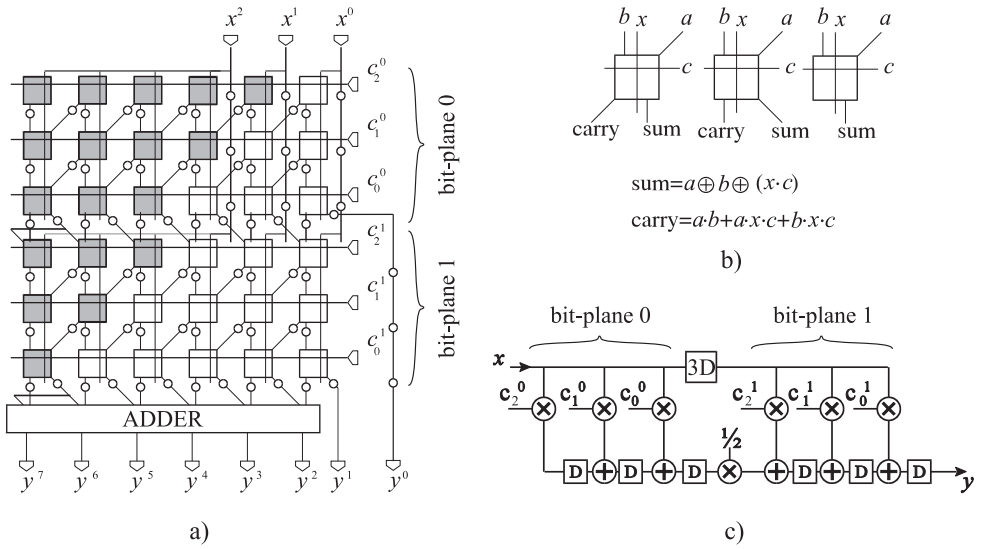

Fig 1.: The BPA for $k_C = 3$ and $m = 2$: a) functional block diagram, b) the functions of the array cells, c) data-flow graph.

in the structure of the array from Fig. 1a caused by shifting the intermediate results for one position to the right between bit-planes (multiplication of the intermediate results with $1/2$ in Fig. 1c), as well as an irregularity introduced by the adder at the end of the array (Fig. 1a).

Let us consider partial defect tolerance of the BPA.

It is said for the system to be Fault Tolerant (FT) if it has the ability to continue correct operation of its tasks after a fault occurrence [13]. Correct operation typically implies that no errors occur at any system output. However, for some applications, such as audio or video processing, correctness of the operation can be defined as an error threshold up to which a user will not notice the error. The cost of manufacturing, verification, and testing can be reduced by relaxing the requirement of 100% correctness for devices and interconnections [3, 7, 14].

The Partial Defect Tolerance (PDT) assumes that the application is tolerant to errors. PDT system is the system in which only the most significant part is fault-tolerant. PDT design process includes partitioning of the system, and it has the following steps [8]:

1. Define a metric and the state of the system fault.
2. Obtain the Error Propagation Graph (EPG) $\mathbf{G}$, taking into account possible vulnerability points of the system.
3. Form the adjacency matrix A for the graph $\mathbf{G}$.
4. Compute the Error Significance Map (ESM) from the adjacency matrix, which marks critical nodes of the system.
5. Obtain the PDT system by applying a suitable FT method to the part of the system which has been marked as critical by ESM.

If we choose Hamming metric to quantify errors and define the state of system fault, every single bit transition affected by existence of defects has the same weight and influence on the output result. This assumption simplifies the error propagation analysis, but it has a drawback that it doesn't distinguish errors in more details [8]. It doesn't take into the account the weight of the errors from the particular array parts in the output result. The PDT architecture with Hamming metric is presented in [8], and the partition of the array that requires the application of some FT technique in order to protect the most significant bit from faults is shown with gray cells in Fig. 1a. In other words, any transition caused by defects in any cell from the shaded part of the array from Fig. 1a will cause the erroneous transition in the most significant bit of the output result [8].

In order to add more details in the error propagation model, keeping the simplicity of the model, the shift to alternative algebraic structures is required.

3. Tropical algebra

With aim to formalize the significant bits propagation model, we give a brief introduction to the tropical algebra [10].

Min-Plus tropical algebra is the algebraic structure $\mathcal{M} = (\mathbb{N}_0 \cup \{\infty\}, \mathtt{min}, +)$ with basic arithmetic operations of addition $(\oplus)$ and multiplication $(\odot)$ defined by

$$x \oplus y := \min(x, y), \quad x \odot y := x + y.$$

In words, the sum of two numbers is their minimum, and the product of two numbers is their usual sum [10, 15]. For example, the tropical sum of 4 and 9 is $4 \oplus 9 = 4$. The tropical product of 4 and 9 equals $4 \odot 9 = 13$. Many of the familiar axioms of arithmetic remain valid in tropical settings [17]. For instance we have:

$$x \oplus y = y \oplus x, \quad x \odot y = y \odot x.$$

These two arithmetic operations are also associative, and the times operator $\odot$ takes precedence when plus $\oplus$ and times $\odot$ occur in the same expression. The distributive law holds for tropical addition and multiplication:

$$x \odot (y \oplus z) = x \odot y \oplus x \odot z. \tag{4}$$

Both arithmetic operations have neutral element. The neutral element for $\oplus$ in $\mathbb{N}_0 \cup \{\infty\}$ is ∞ and the neutral element for $\odot$ in $\mathbb{N}_0 \cup \{\infty\}$ is 0, i.e.

$$x \oplus \infty = x, \quad x \odot 0 = x.$$

In order to express the results we need extensions of the operations $\oplus$ and $\odot$ on matrices [16–18]. We consider matrices, $A = [a_{i,j}]$, with the elements in $\mathbb{N}_0 \cup \{\infty\}$. Tropical addition, denoted $\oplus$, of matrices A and B is matrix $C = A \oplus B$ such that $c_{i,j} = a_{i,j} \oplus b_{i,j}$. Tropical product of matrices A, of type $N\mathrm{x}K$, and B, of type $K\mathrm{x}M$, is matrix $C = A \odot B = AB$, of type $N\mathrm{x}M$, such that

$$c_{i,j} = \bigoplus_{k=1}^{K} (a_{i,k} \odot b_{k,j}), \tag{5}$$

where $i = 1, \ldots, N$, $j = 1, \ldots, M$, and the tropical sum $\bigoplus_{k=1}^{K} \nu_k$ represents the sum $\nu_1 \oplus \nu_2 \oplus \cdots \oplus \nu_K$.

Operation $\odot$ is distributive in the respect to $\oplus$. For example, matrix multiplication is associative operation, i.e.

$$A \odot (B \odot C) = (A \odot B) \odot C.$$

A neutral elements for tropical matrix addition and multiplication are zero matrix $O = [\infty]_{N \text{x} N}$ and identity matrix $I = [\delta_{i,j}]$, with elements

$$\delta_{i,j} = \begin{cases} 0, & i = j \\ \infty, & i \neq j \end{cases} . \tag{6}$$

As usual, we have $A \oplus O = A$, $A \odot O = O$, and $A \odot I = A$.

We denote a tropical k-th power of matrix M as a matrix M^k with elements $m_{i,j}^k$, where $M^1 = M$ and $M^{i+1} = M^i \odot M$.

Having in mind that 0 is the multiplicative identity element, it is shown in [10] that the tropical Pascal's triangle, which rows are the coefficients appearing in a binomial expansion, looks like this [10]:

$$\begin{array}{ccccccc} & & & 0 & & & \\ & & 0 & & 0 & & \\ & 0 & & 0 & & 0 & \\ 0 & & 0 & & 0 & & 0 \\ & & & \cdots & & & \end{array}$$

For example, the fourth row in the triangle represents the identity

$$\begin{aligned} (x \oplus y)^3 &= (x \oplus y) \odot (x \oplus y) \odot (x \oplus y) = \\ &= 0 \odot x^3 \oplus 0 \odot x^2 y \oplus 0 \odot x y^2 \oplus 0 \odot y^3 = \\ &= x^3 \oplus x^2 y \oplus x y^2 \oplus y^3. \end{aligned} \tag{7}$$

Now, we are in the position to develop the error propagation model of the BPA in tropical algebra.

4. Significant bits propagation model

In order to obtain the error propagation model, the impact of error from one part of the system to all other parts of the system should be considered [8]. The systems are usually represented by directed graphs, which stress some of the architecture's properties [8, 12].

Given a directed graph $\mathbf{G} = (V, E)$, where $\mathbf{V} = \{v_1, \ldots, v_n\}$ is a finite set of vertices and $\mathbf{E}$ is a finite set of edges, an edge $e \in \mathbf{E}$ is an ordered pair (v_i, v_j), where $v_i, v_j \in \mathbf{V}$ and an edge (v_i, v_j) means that vertices v_i and v_j are connected. Let v_i and v_j be vertices and let $e_{i,j}$ denote $(v_i, v_j) \in \mathbf{E}$. Each edge has a weight $w_{i,j} \in \mathbb{N}_0 \cup \{\infty\}$.

A path is ordered subset of edges $\mathbf{P} \subset \mathbf{E}, \mathbf{P} = \{e_{i,k_1}, e_{k_1,k_2}, \ldots, e_{k_n,j}\}$, which connects nodes v_i and v_j through nodes $v_{k_1}, v_{k_2}, \ldots, v_{k_n}$. The path length is equal to the cardinality $|\mathbf{P}|$, while the path weight is

$$w_{i,j} = (w_{i,k_1} + w_{k_1,k_2} + \cdots + w_{k_n,j}) . \tag{8}$$

We will define an adjacency matrix A with elements $(a_{i,j})$ of graph $\mathbf{G}$ as

$$a_{i,j} = \begin{cases} w_{i,j}, & e_{i,j} \in \mathbf{E} \\ \infty, & otherwise \end{cases} . \tag{9}$$

The dimensions of the matrix A are dxd, where d is cardinality of $\mathbf{V}$, i.e. $d = |\mathbf{V}|$.

Lemma 1. *For the directed graph* $\mathbf{G}$ *represented by the adjacency matrix A, elements* $a^k_{i,j}$ *of the matrix* A^k *are equal to the length of the shortest path that passes through exactly* k *edges between nodes* i *and* j *within the graph* $\mathbf{G}$.

PROOF We will prove the lemma using mathematical induction. For $k = 2$, according to (5), square of the matrix A is matrix A^2 with elements

$$a^2_{i,j} = \bigoplus_{k=0}^{K-1} \left(a_{i,k} \odot a_{k,j}\right), \tag{10}$$

or, in usual $(\mathbb{R} \cup \{\infty\}, +, \cdot)$ algebra (10) becomes

$$a^2_{i,j} = \min\left\{a_{i,0} + a_{0,j}, a_{i,1} + a_{1,j}, \ldots, a_{i,K-1} + a_{K-1,j}\right\}$$

Having in mind that the element $a_{i,k}$ of the adjacency matrix represents the weight of the edge between i-th and k-th node of the graph $\mathbf{G}$, and that the element $a_{k,j}$ represents the weight of the edge between k-th and j-th node, expression (10) is the minimum weight path between i-th and j-th node that has exactly 2 edges.

If there is n, such that A^n is a matrix which elements are equal to the weights of the shotest n-length paths, then, in the same manner, from (5) it can be shown that A^{n+1} is the matrix with elements equal to the weights of the shortest path with length equal to $(n + 1)$. ■

From Lemma 1 it is straightforward to derive that the following lemma stands.

Lemma 2. *The shortest paths matrix,* $S(\mathbf{G})$*, of graph* $\mathbf{G}$ *can be obtained as*

$$S(\mathbf{G}) = \bigoplus_{i=1}^{\infty} A^i = \bigoplus_{i=1}^{T} A^i, \tag{11}$$

where T *is the maximum-length path within the graph* $\mathbf{G}$.

With aim to construct the analytical model that can indicate the overall magnitude of error in the output result, caused by possible existence of defect, we will define the system error, in the respect to the Euclidean metrics.

Definition 1 (System Error)**.** *Let* y *be the output of the system without errors, and let* y_{err} *be the output of the system with errors. The absolute error of the system is*

$$\Delta_E(y, y_{err}) = |y - y_{err}|, \tag{12}$$

and the relative error of the system is

$$r_E(y, y_{err}) = \frac{\Delta_E(y, y_{err})}{y}. \tag{13}$$

The number of significant bits in y_{err} *is defined as*

$$z(y, y_{err}) = -\log_2\left(r_E\left(y, y_{err}\right)\right). \tag{14}$$

Loss of significance in the output result is an undesirable effect in calculations that occurs when relative error, i.e. the number of significant bits in y_{err}, decreases. For example, if 6-bit output word is $y = 2^6$, and the absolute error is relatively small, for instance $\Delta_E = |y - y_{err}| = 2^1$, there are $z = -\log_2\left(2^1/2^6\right) = 5$ significant bits in the output result y_{err} with the presence of the error. However, if the absolute error increases to $\Delta_E = 2^5$, there will be only $z = 1$ significant bit in the output result y_{err}, while the rest of the bits can't be taken as correct. Thus, in that case we can say that there is a major loss in significance in the output.

This gives us the opportunity to define the system failure using the concept of significate bits in the output result.

Definition 2 (System Failure)**.** *Let α be a threshold value. The state of the system failure for error tolerant architecture is the state where the number of significant bits in y_{err} is*

$$z(y, y_{err}) < \alpha. \tag{15}$$

We will develop the significant bit propagation model by analyzing the change of the number of significant bits, which occurs due to the transition of intermediate results from one vertex to another, under the presence of errors. One basic cell of the BPA from Fig. 1b is represented by one vertex in the error propagation graph $\mathbf{G}$, as it is shown in Fig. 2a.

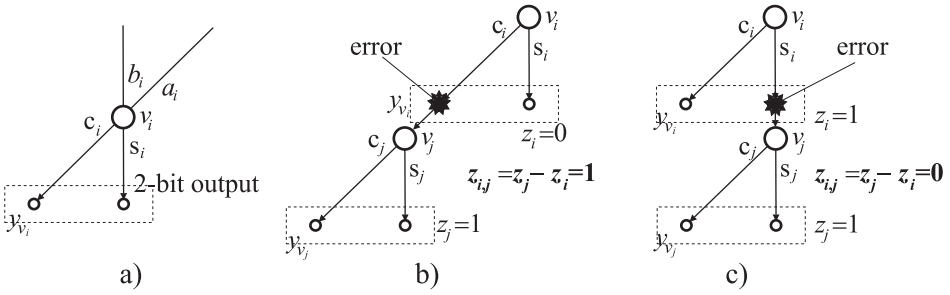

Fig 2.: The error model of the cell: a) two-bit output of the cell, b) the propagation of significant bits through the carry path, c) the propagation through the sum path.

The wiring errors on the paths that feed x and c into the cell are neglected (Fig. 1b), due to the fact that they represent broadcast lines (Fig. 1a). Thus, we assume that the defects might exist in the logic circuits of a cell, and that those defects can cause erroneous outputs for sum s and carry c.

Each basic cell from Fig. 1b, which is represented as a vertex v_i in Fig. 2a, gives two-bit output

$$y_{v_i} = c_i \cdot 2^1 + s_i \cdot 2^0. \tag{16}$$

If the logic circuits that computes the sum output s is defective, then the absolute error on the cell's output from Fig. 2a becomes

$$\Delta_E(y_{v_i}, y_{v_i}^{err}) = \left|y_{v_i} - y_{v_i}^{err}\right| = \left|\left(c_i \cdot 2^1 + s_i \cdot 2^0\right) - \left(c_i \cdot 2^1 + \overline{s_i} \cdot 2^0\right)\right| = 2^0. \tag{17}$$

Using (13) and (14), we have that $r_E = 2^0/2^1 = 1/2$, and the number of significant bits in this case is $z_i = 1$. Similarly, if the logic circuits that computes the carry output is defective, the number of significant bits in the cell's output is $z_i = 0$.

Let us consider the propagation of the number of significant bits through the system. For that purpose, we will assume that the inputs a_i and b_i of the vertex v_i from Fig. 2a might be defective. From the cell's transfer function (Fig. 1b) it is straightforward to show that a single error in either a_i or b_i input will cause that the number of significant bits on the cell's output becomes $z_i = 1$ (Fig. 2a). For example, if both inputs a_i and b_i are 0, and there is an error in a_i that causes $a_i = 1$, then the output $c_i = 0$ is correct, while the output $s_i = 1$ contains the error. Thus, one of the bits is still correct, i.e. $z_i = 1$. The same conclusion can be made for the error in the input b_i.

Figs. 2b and 2c show two possible error propagation paths. Fig. 2b presents the case when the previous cell (v_i in Fig. 2b) has an erroneous carry output c_i. As the error is in the carry bit of the vertex v_i, the number of significant bits in that particular vertex is $z_i = 0$. However, as the error comes to the vertex v_j via its input a_j (Fig. 2b), the vertex v_j will have $z_j = 1$ significant bits. Thus, the change of the number of significant bits, due to the propagation of the error from v_i to v_j in Fig. 2b, is $z_{i,j} = z_j - z_i = 1$.

The change of the number of significant bits $z_{i,j} = 1$ means that the number of significant bits increased for 1, i.e. the magnitude of the error is deceased if the error propagates through carry path. Similarly, we obtain that the change of the number of significant bits, due to the transition of the error through sum path s from Fig. 2a, as $z_{i,j} = 0$, as it is shown in Fig. 2c. In other words, the magnitude of the error doesn't change if the error propagates through the sum path.

Let us note that due to the properties of $\log_2$ function in (14), the number of significant bits can be accumulated along the path. The change in the number of significant bits $z_{i,j}$ from the vertex v_i to v_j, in the model from Fig. 2 represents the influence of error that appear in v_i on the output of the vertex v_j, where $z_{i,j} = T, T \geq 0$ means that the output of the vertex v_j has T significant bits more than v_i.

5. The design and implementation of PDF BPA with Euclidean error metric

We will illustrate the significant bit propagation model from Fig. 2 on the example of PDT BPA design. The Error propagation graph $\mathbf{G}$ of the BPA from Fig. 1a is shown in Fig. 3a. The graph from Fig. 3a is obtained by direct mapping of the BPA topology from Fig. 1a on the error propagation graph $\mathbf{G}$, using the error propagation model of the basic cell from Fig. 2.

In order to avoid irregularities between bit-planes, and obtain more convenient analytical model of error propagation, the graph $\mathbf{G}$ is extended with "dummy" cells, which are shown as white circles in Fig. 3b. There are $p = l_0 + (m-1)$ basic cells in each row, and $q = m \cdot k_C + 1$ basic cells in each column of the graph $\mathbf{G}$ from Fig. 3b, including the cells of the adder at the bottom of the array. The total number of cells is $t = p \cdot q$.

The adjacency matrix of the error propagation graph from Fig. 3b is of the following form:

$$A = \begin{bmatrix} O & A_R & O & \cdots & O \\ O & O & A_R & \cdots & O \\ \vdots & & & \ddots & \\ O & O & O & \cdots & A_R \\ O & O & O & \cdots & A_{add} \end{bmatrix}_{q \times q}, \tag{18}$$

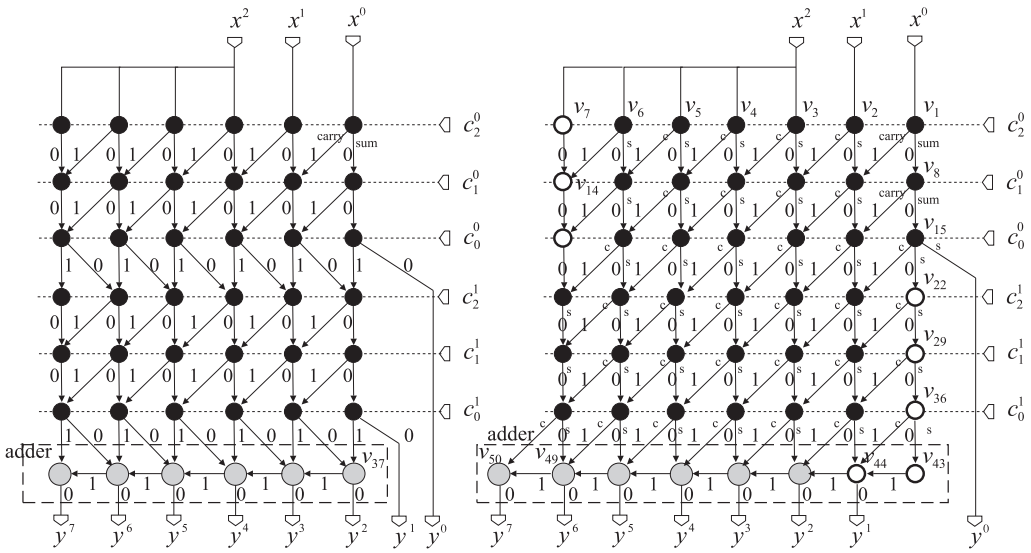

Fig 3.: The Error Propagation Graph for BPA with $k_C = 3$ and $m = 2$: a) direct representation, b) modified graph with removed irregularities

where matrices A_R and A_{add} are

$$A_R = \begin{bmatrix} 0 & 1 & \infty & \cdots & \infty & \infty \\ \infty & 0 & 1 & \cdots & \infty & \infty \\ \infty & \infty & 0 & \cdots & \infty & \infty \\ \vdots & & & \ddots & & \\ \infty & \infty & \infty & \cdots & 0 & 1 \\ \infty & \infty & \infty & \cdots & \infty & 0 \end{bmatrix}_{p \times p} \quad A_{add} = \begin{bmatrix} 0 & 1 & \infty & \cdots & \infty & \infty \\ \infty & 0 & 1 & \cdots & \infty & \infty \\ \infty & \infty & 0 & \cdots & \infty & \infty \\ \vdots & & & \ddots & & \\ \infty & \infty & \infty & \cdots & 0 & 1 \\ \infty & \infty & \infty & \cdots & \infty & 0 \end{bmatrix}_{p \times p}. \tag{19}$$

Due to the position of submatrices A_R and A_{add} in the matrix A from (18), it can be noticed that the submatrix A_R describes the connectivity of two neighboring rows within the graph $\mathbf{G}$, while the submatrix A_{add} corresponds to the edges in the last row of the graph, i.e. the adder. For example from Fig. 3b, the order of submatrices A_R and A_{add} from (19) is $p = 7$, and according to (18), the element $a_{1,8}$ of the adjacency matrix A of the graph $\mathbf{G}$ is $a_{1,8} = 0$, due to the fact that the weight of the edge that connects vertices v_1 and v_8 is equal to 0. The element $a_{1,9} = 1$, etc.

We will give the error propagation model, i.e. the change of the number of significant bits from each basic cell to all other basic cells of the semi-systolic BPA in the form of the following theorem.

Theorem 1. *The significant bits propagation matrix $S(\mathbf{G})$ of the error propagation graph $\mathbf{G}$ of*

the BPA from Fig. 3b can be obtained as

$$S(\mathbf{G}) = \bigoplus_{i=1}^{\infty} A^i = \begin{bmatrix} O & A_R & A_R^2 & A_R^3 & \cdots & \left(\bigoplus_{i=1}^{\infty} A_{add}^i\right) A_R^{q-1} \\ O & O & A_R & A_R^2 & \cdots & \left(\bigoplus_{i=1}^{\infty} A_{add}^i\right) A_R^{q-2} \\ \vdots & & & & \ddots & \\ O & O & O & O & \cdots & \left(\bigoplus_{i=1}^{\infty} A_{add}^i\right) A_R \\ O & O & O & O & \cdots & \left(\bigoplus_{i=1}^{\infty} A_{add}^i\right) \end{bmatrix}_{q\text{x}q}. \tag{20}$$

PROOF The matrix A from (18) can be written as a tropical sum of two matrices

$$A = A_1 \oplus A_2, \tag{21}$$

where

$$A_1 = \begin{bmatrix} O & A_R & O & \cdots & O & O \\ O & O & A_R & \cdots & O & O \\ O & O & O & \cdots & O & O \\ \vdots & & & \ddots & & \\ O & O & O & \cdots & O & A_R \\ O & O & O & \cdots & O & O \end{bmatrix}_{p\text{x}p} \quad A_2 = \begin{bmatrix} O & O & O & \cdots & O & O \\ O & O & O & \cdots & O & O \\ O & O & O & \cdots & O & O \\ \vdots & & & \ddots & & \\ O & O & O & \cdots & O & O \\ O & O & O & \cdots & O & A_{add} \end{bmatrix}_{p\text{x}p}, \tag{22}$$

i.e. the elements at the position (i, j) in the matrices A_1 and A_2 are

$$(A_1)_{i,j} = \begin{cases} A_R, & i+1 = j \\ O, & otherwise \end{cases}, \quad (A_2)_{i,j} = \begin{cases} A_{add}, & i = j = q \\ O, & otherwise \end{cases}. \tag{23}$$

Then, the significant bits propagation matrix $S(\mathbf{G})$ of the error propagation graph $\mathbf{G}$ of the BPA from Fig. 3b, according to lemma 2 and binomial expansion (7), can be obtained as the shortest path matrix

$$S(\mathbf{G}) = \bigoplus_{i=1}^{\infty} A^i = \bigoplus_{i=1}^{\infty} (A_1 \oplus A_2)^i = \bigoplus_{i=1}^{\infty} \left(\bigoplus_{u=0}^{i} A_1^u A_2^{i-u}\right) = \bigoplus_{i=1}^{\infty} \left(A_1^i \oplus \left(\bigoplus_{u=1}^{i-1} A_1^u A_2^{i-u}\right) \oplus A_2^i\right) \tag{24}$$

Using (22) and (23) it is straightforward to develop exponentials of the matrices A_1^n, A_2^n, and $A_1^n A_2^{k-n}$, which by substituting into (24) directly proves (20). ■

The significant bits propagation matrix $S(\mathbf{G})$ from theorem 1 gives the differences in the number of significant bits through graph $\mathbf{G}$ from Fig. 3b for all vertex combinations. If we select only the values from (20) that correspond to the output y, and rewrite them in the form which topologically corresponds to the graph $\mathbf{G}$, we obtain Euclidean matrix D of the graph $\mathbf{G}$ from Fig. 3b as

$$D = \begin{bmatrix} 0 & 1 & 2 & 3\cdots & p-2 & p-1 \\ 0 & 1 & 2 & 3\cdots & p-2 & p-1 \\ 0 & 1 & 2 & 3\cdots & p-2 & p-1 \\ 0 & 1 & 2 & 3\cdots & p-2 & p-1 \\ 0 & 1 & 2 & 3\cdots & p-2 & p-1 \\ 0 & 1 & 2 & 3\cdots & p-2 & p-1 \end{bmatrix}_{p\text{x}q}. \tag{25}$$

The matrix D has elements $d_{i,j}$ equal to the difference of the number of significant bits z in the result y, and the number of significant bits in the vertex that corresponds to the basic cell in i-th column and j-th row of the error propagation graph $\mathbf{G}$.

The most significant partition of the BPA can be obtained from matrix D. For example, if the error threshold (15) is defined with $\alpha = 1$, meaning that the system can tolerate errors if there is at least 1 significant bit in the output (errors up to 2^{p-2}), then, according to definition 2, the system will not be in the failure state if $z \geq 1$. In this case, the system will have failure only if $z = 0$. The most significant partition, as can be seen from (25) is a set of cells from leftmost column of the graph $\mathbf{G}$ from Fig. 3b. This is due to the fact that in that case all other cells might produce error equal or lower then 2^{p-2}.

If the system can tolerate errors up to 2^{p-3}, the threshold from (15) is $\alpha = 2$. Thus, the system can tolerate the errors only if the number of significant bits in the output is $z \geq 2$. The most significant array partition in that case contains the cells that corresponds to two leftmost columns of the matrix D, where $z < 2$, etc.

However, it must be taken into the account that the graph $\mathbf{G}$ from Fig. 3b contains "dummy" cells, as well. Thus, the most significant partition in the case of $\alpha = 1$ contains only cells from the leftmost column of the last bit-plane, as all other bit-planes have "dummy" cells in the leftmost column.

We will evaluate hardware requirements of PDT BPA with Euclidean metric and Triple Modular Redundancy (TMR) as a method of choice for FT basic cells [7, 8]. The most significant partition of the BPA in respect to (25) is shown in Fig. 5a. The Fig. 5a shows BPA for $k_C = 3$ and $m = 3$, and fault tolerant partition filled with dark-lines pattern for error threshold $\alpha = 1$. For the sake of comparison, the BPA partition for error threshold that protects the most significant bit using Hamming's metric is given with the cells shaded in light-gray in Fig. 5a [8]. The structure of TMR cell is shown in Fig. 5b, where "V" stands for majority voter.

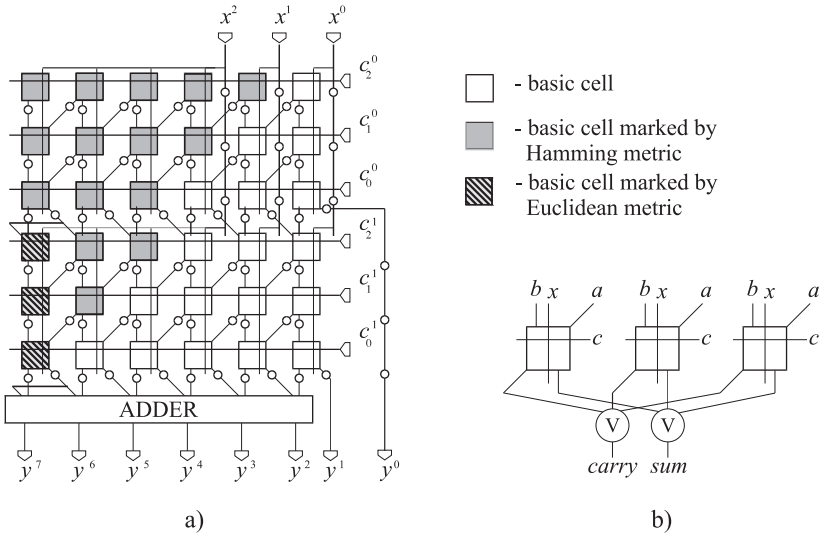

Fig 4.: The PDT BPA that can tolerate defects which can produce errors with the magnitude up to 2^{p-2}

It should be mentioned that the most significant partition obtained using Euclidean metric is a subset of the partition selected using Hamming's metric. Thus, the Hamming metric protects the architecture from defects equally well, or even better, because it involves additional redundancy. However, it requires more hardware. Euclidean metric, in the other hand, gives more details about the error propagation, which can lead to the reduction of chip resources occupation in the cases when it is not needed to obtain exact values of the resulting bits, but rather to limit the overall magnitude of possible error. For example, if the correct output result is "10000000", and the error is produced by the second cell from the right in the first row of the BPA from Fig. 5a, the erroneous result will be "01111111". The PDT BPA with Hamming metric marks this cell as important, because it causes the transition of all output bits, but the PDT BPA with Euclidean metric will not mark this cell as important, because it produces a relatively small magnitude of error equal to 2^1, according to (25).

Hardware requirements of three different array sizes for both Hamming's and Euclidean metric are given in Table 1. The table shows evaluation results for three BPAs with $k_C = 4$ and $m = 8$, and the input word $\{x\}$ of the length $n = 8$ in the case of the BPA labeled as "Arr1", $n = 16$ labeled as "Arr2", and $n = 24$ as "Arr3". The input word length n directly affects the overall width of the bit-plane array [11], thus the BPA "Arr1" has $l_0 = 16$, "Arr2" has $l_0 = 24$, while the BPA "Arr3" is of the width $l_0 = 32$.

For each of the three mentioned BPAs, two columns are given in Table 1: the total number of required basic cells for the PDT BPA with Euclidean metric (PDT BPA$_E$), and the total number of cells for the PDT BPA with Hamming metric (PDT BPA$_H$).

The threshold α is given in table 1 in the range from 0 to 16. For $\alpha = 0$ we have that the BPA is completely tolerant to errors, with no need to involve FT methods on any cell, while for $\alpha = p$ all cells are important and they require application of TMR FT method. The total number of required cells in both BPA$_E$ and BPA$_H$ cases is obtained as the total number of cells in the basic BPA, plus the redundant basic cells added in the most important partition of the array. For example, for $\alpha = 0$ BPA "Arr1" consists of $(m \cdot k_C) \cdot l_0 = 8 \cdot 4 \cdot 16 = 512$ basic cells. For $\alpha = 1$ the most important partition of BPA "Arr1" consists of $k_C = 4$ cells, thus $512 - 4 = 508$ cells in their basic form, while only 4 cells are triplicated by TMR technique as shown in Fig. 5b. The total number of required basic cells for "Arr1" and $\alpha = 1$ is $(512 - 4) \cdot 1 + 4 \cdot 3 = 520$ (Table 1).

Table 1.: Number of basic cells required for the implementation of arrays with different sizes with α as a parameter

	Arr1		Arr2		Arr3	
α	BPA$_E$	BPA$_H$	BPA$_E$	BPA$_H$	BPA$_E$	BPA$_H$
0	512	512	768	768	1024	1024
1	520	1236	776	1598	1032	1856
2	536	1274	792	1658	1048	1920
4	592	1344	848	1770	1104	2048
8	800	1450	1056	1962	1312	2304
16	1312	1536	1568	2218	1824	2730

The results from Table 1 are illustrated in Fig. 5. From Table 1 and Fig. 5 it can be seen that PDT BPAs with both Euclidean and Hamming's metric for $\alpha > 0$ have hardware overhead, due

to unavoidable triplication caused by application of TMR. However, the number of triplicated cells in the PDT BPA with Euclidean metric is lower than the number of triplicated cells in the PDT BPA with Hamming's metric, especially for lower values of α. For maximum α, where all output bits are important, for both Euclidean and Hamming's metrics the number of triplicated cells will be maximal and equal. For lower values of threshold α significant hardware reductions can be achieved.

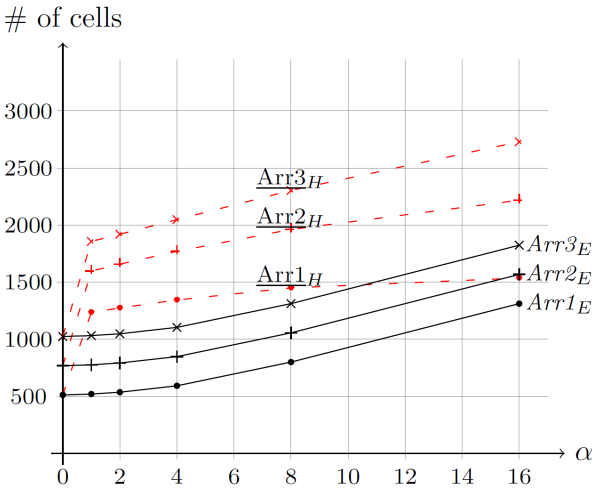

Fig 5.: The PDT BPA that can tolerate defects which can produce errors with the magnitude up to 2^{p-2}

6. Concluding remarks

In this paper a novel technique for mathematical modeling of significant digits propagation through the system under the presence of errors is proposed. The model incorporates Euclidean metric, and it can be used to obtain the difference in the number of significant digits between any two given intermediate results within the system. It is shown that the model is suitable for fault tolerant system design if the application doesn't require 100% correctness of the output. Min-Plus algebra is chosen to formalize structural dependencies within the system. The proposed model is evaluated on the example of semi-systolic bit-plane array for FIR filtering, with error defined by Euclidean distance. The most significant partition obtained using Euclidean metric is a subset of the partition selected using Hamming's metric. From that perspective, the Hamming metric protects the architecture from defects equally well, or even better, because it involves additional redundancy. However, it introduces the significant hardware overhead. Euclidean metric, used in this paper, gives more details about the error propagation and enables the limitation of the overall error magnitude in the presence of errors, while the overall hardware overhead is reduced.

Acknowledgement. The research was supported in part by the Serbian Ministry of Education, Science and Technological Development (Project TR32012).

References

[1] MOORE G.E., *Cramming more components into integrated circuits*, Electronics, **38**(8), 1965.

[2] HASELMAN Michael, HAUCK Scott, *The Future of Integrated Circuits: A Survey of Nanoelectronics*, Proceedings of the IEEE, **98**(1), pp. 11–38, 2010.

[3] ITRS, *International Technology Roadmap for Semiconductors*, http://www.itrs.net/links/2010itrs/home2010.htm, 2010.

[4] MISHRA M., GOLDSTEIN S.C., *Defect Tolerance at the End of the Roadmap*, Intl Test Conference Proceedings, **1**, pp. 1201–1210, 2003.

[5] MIJATOVIC D., EIJKEL J., VAN DEN BERG A., *Technologies for nanofluidic systems: top-down vs. bottom-upa review.*, Lab on a Chip, **5**(5), pp. 492–500, 2005.

[6] RIVERS J.A., PRABHAKAR Kudva, *Reliability challenges and system performance at the architecture level.* IEEE Design & Test of Computers, No. 6, pp. 62–73, 2009.

[7] ĆIRIĆ Vladimir, CVETKOVIĆ Aleksandar, MILENTIJEVIĆ Ivan, *Yield analysis of partial defect tolerant bit-plane array*, Elsevier, Int.J., Computers and Mathematics with Applications, **59**(1), pp. 98–107, 2010.

[8] ĆIRIĆ Vladimir, KOLOKOTRONIS Jelena, MILENTIJEVIĆ Ivan, *Partial Error-Tolerance for Bit-Plane FIR Filter Architecture*, International Journal of Electronics and Communication, Elsevier Science (AEU), **63**, 2009.

[9] ĆIRIĆ Vladimir, CVETKOVIĆ Aleksandar, SIMIC Vladimir, MILENTIJEVIC Ivan, *Tropical algebra based framework for error propagation analysis in systolic arrays*, Elsevier Applied Mathematics and Computation, **225**, pp. 512–525, 2013.

[10] SPEYER David, STURMFELS Bernd, *Tropical mathematics*, arXiv preprint math/0408099, 2004.

[11] CIRIC Vladimir, MILENTIJEVIC Ivan, *Configurable folded array for FIR filtering*, Journal of Systems Architecture, An Internation Journal, Elsevier Science, **54**(1–2), pp. 177–196, 2008.

[12] NOLL T., *Semi-systolic Maximum Rate Transversal Filters with Programmable coefficients*, Workshop of Systolic Architectures, Oxford, U.K., pp. 103-112, 1986.

[13] DEHON A., *Reconfigurable Computing: The Theory and Practice of FPGA-Based Computing*, Defect and Fault Tolerance (Ed. S. Hauck and A. DeHon), Morgan Kaufmann, San Francisco, US, 2008.

[14] BREUER M., *Multimedia Applications and Imprecise Computation*, Proceedings on the 8th Euromicro conference on Digital System Design, Euromicro, Porto, Portugal, 0-7695-2433-8/05, 2005.

[15] SIMON Imre, *Limited Subsets of a Free Monoid*, in Proc. of 19th Annual Symposium on Foundations of Computer Science, Piscataway, N.J., pp. 143–150, 1978.

[16] LEUNG Hing, *On the topological structure of a finitely generated semigroup of matrices*, Semigroup Forum, **37**(1), Springer, pp. 273–287, 1988.

[17] IZHAKIAN Zur, *Tropical Arithmetic and Matrix Algebra*, Communications in Algebra, **37**(4), pp. 1445–1468, 2009.

[18] IZHAKIAN Zur, ROWEN Louis *The Tropical Rank of a Tropical Matrix*, Communications in Algebra, **37**(11), pp. 3912–3927, 2009.