Applied Mathematics and Computation 225 (2013) 512–525

Contents lists available at ScienceDirect

Applied Mathematics and Computation

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

Tropical algebra based framework for error propagation analysis in systolic arrays

Vladimir Ciric [a,*], Aleksandar Cvetković [b], Vladimir Simić [a], Ivan Milentijević [a]

[a] University of Niš, Faculty of Electronic Engineering, Computer Science Department, Aleksandra Medvedeva 14, P.O. Box 73, Niš, Serbia
[b] University of Belgrade, Faculty of Mechanical Engineering, Kraljice Marije 16, Belgrade, Serbia

ARTICLE INFO

Keywords:
Nanoarchitectures
Fault tolerant systems
Error propagation analysis
Partial error tolerance
Tropical algebra

ABSTRACT

Nanotechnology is yet to come, but even now, in early stage of development it is clear that defect and fault levels will be much higher than current CMOS technology. The exact level of defect densities is unknown, but it is assumed that 1—15% on-chip resources will be defective. Novel techniques and architectures have to be devised in order for nanoelectronics to become a viable replacement for current VLSI processes. With defect rates for current VLSI processes in the range of 1 part per billion, manufacturers can afford to discard any chip that is found to be defective. However, in order to increase fabrication yield, nanotechnology requires extensive and computationally demanding analysis of defect significance. In order to simplify the analysis, in this paper we propose a mathematical framework based on tropical algebra for circuit analysis. It is more descriptive and convenient to use in graph analysis than traditional algebra. In tropical algebra, we will derive a simple iterative algorithm for error propagation analysis of systolic arrays. It will be shown that the computational complexity of the proposed algorithm is reduced from $\mathcal{O}(T^3)$ to $\mathcal{O}(T^2)$, where T is the number of array cells. An example of tropical algebra analysis and design of partially defect tolerant hexagonal systolic multiplier will be given, too.

1. Introduction

In 1975 Gordon Moore, cofounder of Intel, predicted that the number of transistors that could be placed on a chip would double every two years [1]. Chip manufacturers have relied on the continued scaling down of the transistor size to achieve the exponential growth in transistor counts, but the scaling will be ended soon. Moore's law cannot hold forever. When the size of a transistor reaches the limit of 1–2 atoms, the scaling will have to cease and a new technology have to be adopted [2].

The current projections by the International Technology Roadmap for Semiconductors (ITRS) say that the end of the road on MOSFET scaling will arrive sometime around 2018 with a 22 nm process [3]. Even getting to 22 nm presents some major unsolved hurdles. Among these are increasing power consumption, particularly through leakage currents, less tolerance for process variation, and increasing cost [2]. Given the history of the semiconductor industry, most of these issues can probably be solved with current processes. However, there are two significant exceptions. Physical size limitations and astounding costs may require a shift in the fundamental way integrated circuits are fabricated. Many researchers believe that the shift will be to nanoelectronics. It is estimated that nanoelectronics will be able to integrate 10^{12} devices per cm^2, while the ITRS

* Corresponding author.
E-mail addresses: vladimir.ciric@elfak.ni.ac.rs (V. Ciric), acvetkovic@mas.bg.ac.rs (A. Cvetković), vladimir.simic@elfak.ni.ac.rs (V. Simić), ivan.milentijevic@elfak.ni.ac.rs (I. Milentijević).

0096-3003/$ - see front matter © 2013 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.amc.2013.09.059

[3] estimates that at the end of the roadmap in 2018 manufacturers will only be able to achieve 10^{10} MOSFET transistors per cm^2 [2,4].

With current transistors and lithography, essentially any circuit can be created and manufactured with high reliability. This level of control is unlikely to be possible for nanoelectronics. Because of their small size, nanoelectronic devices will likely not be able to be deterministically placed [5]. Researchers have been able to manipulate components with atomic force microscopes, but this will be impractical for full chips. Even if advances in manufacturing allow other ways to manipulate at this scale, the tolerances required will likely make the costs prohibitive. One of the approaches is to let the circuits to assemble themselves [2,5]. The consequence is that defects are inevitable and must be systematically handled [6].

Even though nanoelectronics device fabrication is in its infancy, it is clear that defect and fault levels will be much higher than current CMOS technology. The exact level of defect densities is unknown, but it is assumed that 1–15% of the resources on a chip (wires, switches FETs, etc.) will be defective [4,7]. Current research on fault tolerance has followed three tracks to solve this problem. These are configuring around the defects [8], masking faults with redundancy or designs that are inherently fault tolerant [2,9].

A balanced combination of circuit and logic level innovations, and architecture and software level solutions is necessary to achieve the required resiliency. In particular, a comprehensive understanding of the vulnerabilities of the architecture is necessary [6,10]. When such information is available, appropriate and cost-effective approaches can facilitate efficient error resiliency. Such a toolset and methodology would help derive vulnerability maps of the chip, which can be optionally visualized as color-coded chip floorplans indicating hotspot regions of the chip [6], and facilitate in finding the most vulnerable partition of the architecture to apply some of the defect tolerance methods [10]. Architecture which has the most vulnerable partition designed as defect tolerant is called Partially Defect Tolerant (PDT) architecture [10]. A careful analysis of defect rates and error propagation paths through an architecture can lead to fabrication yield improvement [11].

The error propagation analysis relies on a selected error metrics, and even for relatively simple metrics the analysis can be a complex task. Furthermore, the analysis steps usually depend on a given topology [10]. Finding a suitable algebraic structure as a mathematical framework could be a powerful tool for problem complexity reduction in circuit design [12]. In this paper we propose a framework for error propagation analysis, which is based on relatively new mathematical field called tropical algebra [13,14]. In tropical algebra, we will derive simple iterative algorithm for error propagation analysis of systolic arrays in general. It will be shown that the computational complexity of the proposed algorithm is reduced from $\mathcal{O}(T^3)$ to $\mathcal{O}(T^2)$, where T is the number of array cells. An example of tropical algebra analysis and design of partially defect tolerant hexagonal systolic multiplier will be given.

The paper is organized as follows. Section 2 gives a brief background on error propagation, error significant maps, and partial defect tolerance. Section 3 is devoted to the tropical algebra as a basis for error propagation analysis. Section 4 is the main section and presents the proposed analysis method in formal mathematical manner. The complexity reduction analysis will be given in Section 5. A design example of the PDT systolic multiplier will be presented in Section 6. Section 7 is devoted to FPGA implementation, while in Section 8 concluding remarks are given.

2. Partial defect tolerance and error analysis

With aim to clarify the analysis of error propagation, we give a brief review of Error Significance Map (ESM) development, and its role in yield improvement through the application of partial defect tolerance [10,11].

Fault tolerance (FT) is the ability of a system to continue correct operation of its tasks after hardware or software faults occur (see [15]). Correct operation typically implies that no errors occur at any system output. Relaxing the requirement of 100% correctness for devices and interconnections may dramatically reduce costs of manufacturing, verification, and testing (see [3]). If a signal processing device has a minor hardware defect, it can still produce results that are good enough for the end user. If so, they could also be sold rather than be discarded (see [16]).

The PDT assumes that the application of the architecture is, mainly, tolerant to errors. In order to improve the yield, the PDT applies FT methods only to the most critical parts of the architecture. Taking into the account the probability of having a defect, this approach introduces a design compromise that can lead to yield improvement [11].

PDT design process has the following steps [10]:

1. Define a metric and the state of the system fault - The metric defines the distance between the correct and erroneous result. The system is not operational if the distance is above the threshold that application can tolerate.
2. Error Propagation Graph (EPG) – The EPG is obtained for a given architecture, taking into account possible vulnerability points and the adopted metric. The edges of the graph indicate the paths between architecture's nodes where errors can propagate. The graph edges are weighted by the defined metric.
3. Adjacency matrix – The matrix is obtained from the EPG in such a manner that its elements indicate existence of edges between the corresponding nodes.
4. ESM – computed from the adjacency matrix, and interpreted as a matrix with the structure that visualize critical nodes for the correct operation of the architecture. Allowed error threshold is a parameter used in computation of the map. The

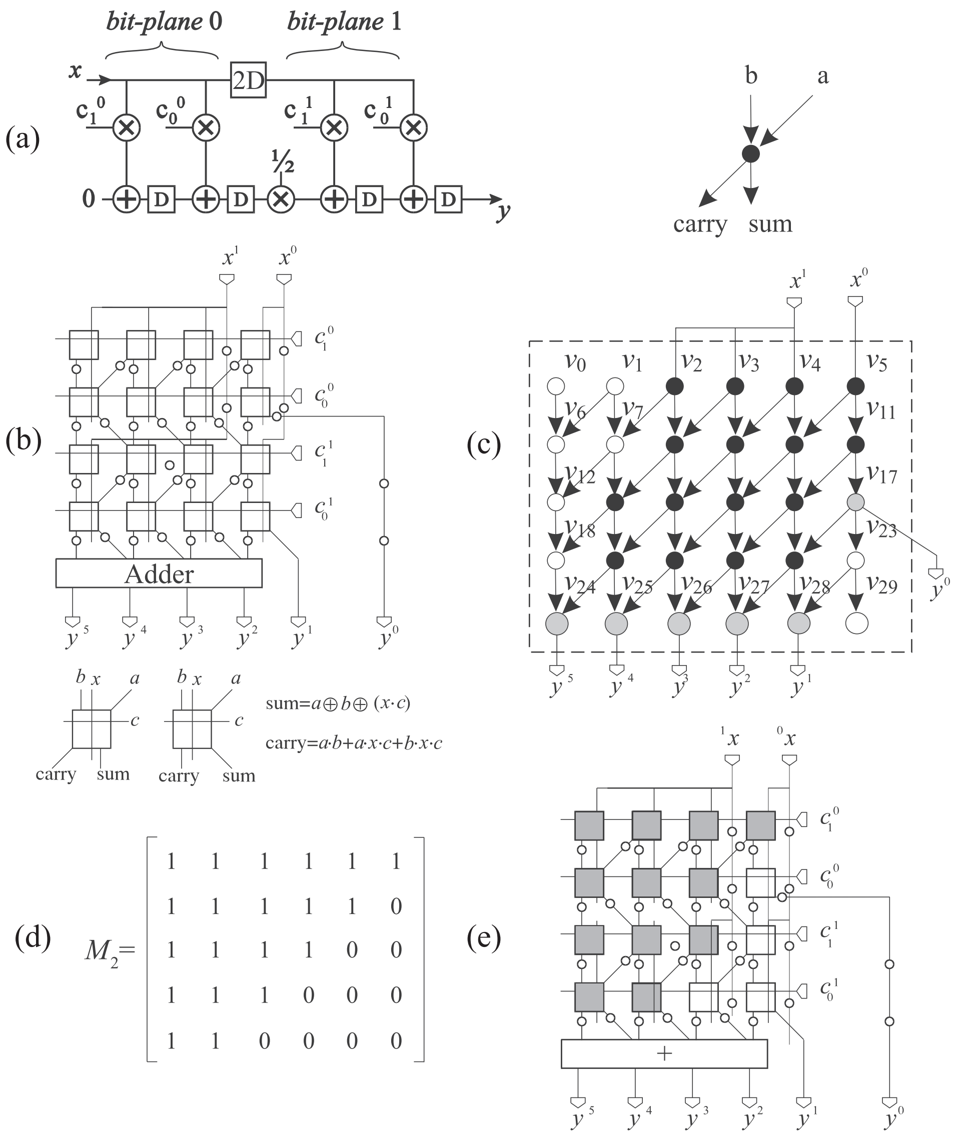
lower the error threshold, the more of the nodes will be declared as important. The ESM development technique, proposed in [10], suggests using a transitive closure on the adjacency matrix.

5. The PDT architecture is obtained applying a suitable FT method to the part of the architecture which has been marked by ESM.

We will illustrate PDT design process on the example of PDT bit-plane FIR filter design, proposed in [10] (Fig. 1). The FIR filtering is a processes of transforming the input sequence $\{x_i\}, i = 0, 1, 2, \ldots$, into the output sequence $\{y_i\}, i = 0, 1, 2, \ldots$, using the coefficients $c_i, i = 0, 1, \ldots, k-1$, as

$$y_i = x_i \cdot c_0 + x_{i-1} \cdot c_1 + \cdots + x_{i-(k-1)} \cdot c_{k-1}. \qquad (1)$$

One possible hardware implementation of FIR filtering is on the bit-plane array. The bit-plane array cells operate with single bits, multiplying one bit from the input word with weight 2^j, denoted as x^j, and one bit from the coefficient c_i with weight 2^j, denoted as c_i^j, and accumulating the previous partial product to the computed product [10,17]. The transfer function of the bit-plane FIR filtering array is a z-domain representation of the filtering Eq. (1), where all operations are bit-operations. The transfer function is



$$M_2 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Fig. 1. PDT semi-systolic bit-plane FIR filter array design steps: (a) Data Flow Graph of the semi-systolic bit-plane FIR filter, (b) the basic array, (c) error propagation graph, (d) the ESM, (e) PDT semi-systolic array.

$$\begin{aligned}G(z) = \frac{Y(z)}{X(z)} &= z^{-1}(c_0^{m-1}2^{m-1}z^{-(m-1)k} + z^{-1}(c_1^{m-1}2^{m-1}z^{-(m-1)k} + \ldots \\ &+ z^{-1}(c_{k-1}^{m-1}2^{m-1}z^{-(m-1)k} \\ &+ z^{-1}(c_0^{m-2}2^{m-2}z^{-(m-2)k} + z^{-1}(c_1^{m-2}2^{m-2}z^{-(m-2)k} + \ldots \\ &+ z^{-1}(c_{k-1}^{m-2}2^{m-2}z^{-(m-2)k} + \\ &\ldots \\ &+ z^{-1}(c_0^0 2^0 + z^{-1}(c_1^0 2^0 + z^{-1}(\cdots + z^{-1}(c_{k-1}^0 2^0))\ldots). \end{aligned} \tag{2}$$

The bit-operations that use the coefficients bits with same weight are grouped into the so called bit-planes (Eq. 2). For the sake of illustration, Fig. 1a shows an algorithmic abstraction of hardware realization of relatively small bit-plane array using Data Flow Graph (DFG). This array performs filtering of input sequence x_i using $k = 2$ coefficients with length $m = 2$. As the coefficient length is $m = 2$ there are two bit-planes in Fig. 1a. Hardware realization is direct application of Eq. (2), and it is shown in Fig. 1b. The input word length of the array from Fig. 1b is $n = 2$. The basic cells are represented using squares, while the delays are denoted with circles (Fig. 1b). Each basic cell performs basic bit operations, as it is shown in Fig. 1b. The output word is formed by the adder, which adds the carries to partially formed sums.

PDT proposed in [10] is illustrated on the example of the bit-plane FIR filter from Fig. 1b. In the respect to the previously described PDT steps 1 to 5, the chosen metric within the first step in the example published in [10] is Hamming's metric. Based on the proposed metric, it is assumed that the system is operational if

$$d_H(A, A_{err}) = \Delta_H = \sum_{i=L-\alpha}^{L-1} (a^i \veebar a^i_{err}) = 0, \tag{3}$$

where $d_H(A, A_{err})$ is a distance between the correct and erroneous result (A and A_{err}, respectively), a^i and a^i_{err} are bits of A and A_{err}, respectively, with weights 2^i, L is the total number of architecture outputs, and α is the number of the most significant output bits that cannot tolerate errors, which represents allowed error threshold. In other words, according to (3), system is operational if α most significant bits of obtained result are identical to the bits of the correct result.

The EPG of the array from Fig. 1b is shown in Fig. 1c. While forming the EPG in the step 2 of the PDT design process from the architecture shown in Fig. 1b, new fictive nodes are introduced in order to obtain highly regular graph structure, which is more convenient for further manipulation [10]. The nodes that exist in the implementation in Fig. 1b are colored in black in Fig. 1c, fictive nodes that introduce high-regularity into the graph are shown in white, while the grey color denotes output nodes. Having in mind the metric defined by (3), the weights of all existing edges in the graph from Fig. 1c are equal to $w = 1$. In other words, this is the simplest form where the error that appears in one node propagates with the same weight to the other.

Due to the size of the adjacency matrix, the matrix obtained in the step 3 of the PDT design process is omitted from Fig. 1 (see [10]). The ESM is obtained in the fourth step of the PDT process using the transitive closure of the adjacency matrix, and it is shown in Fig. 1d, for $\alpha = 2$. The ESM is a matrix $M_\alpha = m^\alpha_{i,j}$, whose element $m^\alpha_{i,j} = 1$ if there is a path from a node on the position (i,j) in the EPG to some of the $\alpha = 2$ most significant outputs (Fig. 1) [10].

The ESM visualizes the vulnerabilities of the array. Using the map, the PDT architecture is obtained in the fifth step of the PDT design process, and shown in Fig. 1e. The FT partition of the array is shown in gray in Fig 1e.

It is shown in [11] that fabrication yield can be increased by application of PDT, using FT techniques to increase the tolerance of the most critical part of the array, having in mind defect rates. The defect rate, for which the PDT architecture increases the yield, depends on the size of the array. For the scope of the problem typical in nanotechnology PDT can introduce yield improvement [11].

However, the computations performed using the PDT method proposed in [10], due to the simplified metric that is used, do not take into the consideration the potential change of the error weight after the propagation. It has the side effect that can be illustrated on the following example. If the magnitude of the correct output signal $\{y_i\}$ is $011\ldots111$, and the error within a signal is small, e.g. $000\ldots001$, the output result will be $100\ldots000$. This seems to be an error in all bits of the output result, including the most significant one. Due to this fact, the PDT method with the metric given with (3) will mark particular cell (upper-right in Fig. 1b) as important, regardless the fact that the error produced within the cell is very small. Thus, if the more accurate metrics are used in the PDT design the yield could be improved even further then it is given in [11].

If the chosen error metric was Euclidian, instead of Hamming's, the elements of ESM from Fig. 1d will be rather $m_{i,j} \in \mathbf{N}_0$, instead of $m_{i,j} \in \{0,1\}$, and the elements $m_{i,j}$ will be proportional to the weighted path length. This requires more robust mathematical expressions, and more complex computations. Tropical algebra [13] can be used as a tool that can help in reducing the complexity of such operations. In the next section we'll briefly introduce the tropical algebra, and develop required tropical matrix forms.

3. Tropical algebra

Rather than coloring the architecture vulnerabilities in "black and white", as illustrated in previous example ($0-1$ coding in Fig. 1), in this paper we are proposing a toolset for ESM development, which will include much more details of the architecture vulnerabilities. We will refer in further text to this maps as the Color-Coded ESM (CCESM). The prefix *Color-Coded* is given with the respect to the vision of "future toolsets and technologies, required for the success of nanotechnology", proposed in [6].

The development of CCESM can be a complex task. Hence, more accurate metrics should be used. In order to reduce development complexity, in this paper we propose a novel method for error propagation analysis of systolic arrays based on tropical algebra.

3.1. Tropical semiring

The tropical algebra is an algebraic structure $(\mathbb{R} \cup \{\infty\}, \oplus, \odot)$. This algebraic structure is known as the tropical semiring or as the min-plus algebra. The adjective "tropical" was coined by French mathematicians, notably Jean-Eric Pin [13], to honor their Brazilian colleague Imre Simon [18], who pioneered the use of min-plus algebra.

In the tropical semiring, the basic arithmetic operations of addition and multiplication of real numbers are redefined as

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

In words, the tropical sum of two numbers is their minimum, and the tropical product of two numbers is their usual sum. 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 mathematics [14]. For instance, both addition and multiplication are commutative:

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

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:

$$x \odot (y \oplus z) = x \odot y \oplus x \odot z.$$

Both arithmetic operations have a neutral element. Infinity is the neutral element for addition and zero is the neutral element for multiplication:

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

Analysis of error propagation through systolic arrays requires matrix manipulation. We will briefly investigate the properties of matrix operations in tropical algebra.

3.2. Properties of basic matrix operations in tropical algebra

Let $A = [a_{i,j}]$ and $B = [b_{i,j}]$ be matrices with NxK and KxM elements, respectively.

Definition 1 (*Matrix multiplication*). Tropical product of matrices A and B is matrix $C = A \odot B$ with NxM elements of the following form:

$$c_{i,j} = \bigoplus_{k=0}^{K-1} (a_{i,k} \odot b_{k,j}), \tag{4}$$

where $i = 0, 1, \ldots, N-1, j = 0, 1, \ldots, M-1$, and the tropical sum $\bigoplus_{k=0}^{K-1} v_k$ represents the sum $v_0 \oplus v_1 \oplus \cdots \oplus v_{K-1}$.

Matrix multiplication is associative operation, i.e.

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

A neutral element for tropical matrix multiplication is identity matrix $I = [u_{i,j}]$, with elements

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

Let matrix $A = [a_{i,j}]$, with NxK elements, be represented by its submatrices $A_{p,q}$ as

$$A = \begin{bmatrix} A_{0,0} & A_{0,1} & \cdots & A_{0,k-1} \\ A_{1,0} & A_{1,1} & & A_{1,k-1} \\ \vdots & & \ddots & \\ A_{n-1,0} & A_{n-1,1} & & A_{n-1,k-1} \end{bmatrix}_{NxK}$$

and let matrix $B = [b_{i,j}]$, with KxM elements, be represented by its submatrices $B_{p,q}$ in the same manner. Directly from the definition of tropical matrix multiplication (Def. 1) it can be shown that the following lemma holds.

Lemma 1. *Submatrix $C_{p,q}$ of matrix $C = A \odot B$ is of the following form*

$$C_{p,q} = \bigoplus_{i=0}^{k-1} A_{p,i} \odot B_{i,q}. \tag{5}$$

It should be noted that Lemma 1 stands for any semiring $(\mathbb{G}, +, \cdot)$. Let a tropical kth power of the adjacency matrix A of graph $\mathbf{G}$ be represented as a matrix $A^{ⓚ}$ with elements $a_{i,j}^{ⓚ}$.

Lemma 2. *Elements $a_{i,j}^{ⓚ}$ of the matrix $A^{ⓚ}$ are equal to the length of the shortest path that pass 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 (4), square of the matrix A is matrix $A^{②}$ with elements

$$a_{i,j}^{②} = \bigoplus_{k=0}^{K-1} (a_{i,k} \odot a_{k,j}), \tag{6}$$

or, in usual $(\mathbb{R} \cup \{\infty\}, +, \cdot)$ algebra Eq. (6) becomes

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

Having in mind that the element $a_{i,k}$ of the adjacency matrix represents the weight of the edge between ith and kth node of the graph $\mathbf{G}$, and that the element $a_{k,j}$ represents the weight of the edge between kth and jth node, expression (6) is the minimum weight path between ith and jth node that has exactly 2 edges.

If there is n, such that $A^{ⓝ}$ is matrix whose elements are equal to the weights of the shortest n-length paths, then, in the same manner, from (4) it can be shown that $A^{\textcircled{n+1}}$ is the matrix with elements equal to the weights of the shortest $(n+1)$ length path. □

From Lemma 2 it is straightforward to derive that the following lemma stands.

Lemma 3. *The shortest paths matrix, $S(\mathbf{G})$, of graph $\mathbf{G}$ can be obtained as*

$$S(\mathbf{G}) = \bigoplus_{i=1}^{T} A^{ⓘ} = \bigoplus_{i=1}^{\infty} A^{ⓘ}, \tag{7}$$

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

4. Error propagation through systolic arrays

Using the toolset adopted in the previous section, in this section we will derive an algebraic form of the error propagation of systolic arrays, which will be further used in the design of partially error tolerant arrays with much more details of array vulnerabilities.

Error propagation can be defined as an erroneous computation in the architecture's node v_i, which is not caused by a defect within that particular node, but rather by the erroneous data on the node's inputs, caused and propagated from some of the defective predecessors nodes [10,15]. Let the error propagation through an architecture be represented by the error flow graph $\mathbf{G}$. Given a directed graph $\mathbf{G} = (V, E)$, where $\mathbf{V} = \{v_1, \ldots, v_n\}$ is a finite set of vertices, where each vertex stands for one architecture node in which error can occur, and $\mathbf{E}$ is a finite set of edges; An edge $e_{i,j} \in \mathbf{E}$ 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 error that appears within vertex v_i can propagate to the vertex v_j. Each edge $e_{i,j} \in \mathbf{E}$ has a weight $w_{i,j}$ equal to the influence of the error that originates from the vertex v_i into the computation in the vertex v_j.

Let each processing element of an array be represented by one vertex in the graph $\mathbf{G}$. We will consider error propagation graphs that have regular topologies with only neighboring rows and columns connected, as the most common topologies [2]. Therefore, the adjacency matrix of the graph $\mathbf{G}$ of regular array is of the following form

$$A(\mathbf{G}_E) = \begin{bmatrix} A_R & A_C & \infty & \cdots & \infty \\ \infty & A_R & A_C & & \infty \\ \infty & \infty & A_R & & \infty \\ \vdots & & & \ddots & \\ \infty & \infty & \infty & & A_R \end{bmatrix}_{nxn}, \tag{8}$$

where A_R is row connectivity matrix, with elements $a_{i,j}^{R}$ equal to the weight of the edge between nodes i and j in neighboring rows, and A_C is the column connectivity matrix, with elements $a_{i,j}^{C}$ equal to the weight of the edge between nodes i and j in neighboring columns.

Lemma 4. *Submatrix $A_{p,q}^{(n)}$, on position (p,q) within the nth tropical power of systolic array's adjacency matrix $A^{ⓝ}$ can be obtained using the following iterative formula:*

$$A_{p,q}^{(n)} = A_{p,q}^{(n-1)} A_R \oplus A_{p,q-1}^{(n-1)} A_C, \tag{9}$$

where $A_{p,q}^{(1)} = A_{p,q}$, and $A_k A_l$ denotes a tropical product $A_k A_l = A_k \odot A_l$.

Recurrence relation holds true for $n = 1$ if we choose $A_{p,p}^{(0)} = I$ and $A_{p,q} = \infty$ for $p \neq q$.

Proof. We will prove the lemma using mathematical induction. Substituting $n = 2$ in (9), using the properties given in (8), we obtain following submatrices

$$\begin{aligned}
A_{1,1}^{②} &= A_{1,1}A_R \oplus A_{1,0}A_C = A_R^{②},\\
A_{1,2}^{②} &= A_{1,2}A_R \oplus A_{1,1}A_C = A_C A_R \oplus A_R A_C,\\
A_{1,3}^{②} &= A_{1,3}A_R \oplus A_{1,2}A_C = A_C^{②},\\
&\dots
\end{aligned}$$

The same equalities are obtained substituting (8) into the definition of the tropical matrix multiplication (5), which proves the Eq. (9) for $n = 2$.

Let (9) holds for $n = k$, i.e.

$$A_{p,q}^{(k)} = A_{p,q}^{(k-1)} A_R \oplus A_{p,q-1}^{(k-1)} A_C.$$

Let the number of submatrices in a column of the matrix given with (8) be N. According to Lemma 1, we obtain $(k+1)$th iteration as

$$A_{p,q}^{(k+1)} = \bigoplus_{i=1}^{N} \left(A_{p,i}^{(k)} \odot A_{i,q} \right),$$

where only two sum members are not equal to ∞, namely $A_{i,q} = A_R$ for $i = q$, and $A_{i,q-1} = A_C$ for $i = q-1$ (8), thus

$$A_{p,q}^{(k+1)} = A_{p,q}^{(k)} A_{q,q} \oplus A_{p,q-1}^{(k)} A_{q-1,q} = A_{p,q}^{(k)} A_R \oplus A_{p,q-1}^{(k)} A_C,$$

which proves the lemma for $n > 1$.

For $n = 1$ we have

$$\begin{aligned}
A_{p,p}^{(1)} &= A_{p,p}^{(0)} A_R \oplus A_{p,p-1}^{(0)} A_C = I \odot A_R \oplus \infty \odot A_C = A_R,\\
A_{p,p+1}^{(1)} &= A_{p,p+1}^{(0)} A_R \oplus A_{p,p}^{(0)} A_C = \infty \odot A_R \oplus I \odot A_C = A_C,\\
A_{p,q}^{(1)} &= A_{p,q}^{(0)} A_R \oplus A_{p,q-1}^{(0)} A_C = \infty \odot A_R \oplus \infty \odot A_C = \infty,
\end{aligned}$$

which proves the lemma. □

According to the structure of the matrix $A^{(0)}$ we can interpret it as a matrix of the paths of the length 0.

Lemma 4 gives an iterative algorithm for computation of nth tropical power of adjacency matrix. In order to obtain the shortest path matrix $S(\mathbf{G}_E)$, in the respect to Lemma 3, we will obtain the form of particular submatrices $A_{p,q}^{(n)}$ on the typical positions within the nth tropical power $A^{ⓝ}$ of the matrix A given with Eq. (8).

Lemma 5

(i) *For $n \in \mathbb{N}$, matrix $A^{ⓝ} = \left[A_{p,q}^{(n)}\right]$ has the following property, if $p - q = p_1 - q_1$ then $A_{p,q}^{(n)} = A_{p_1,q_1}^{(n)}$.*
(ii) *If $p > q$ we have $A_{p,q}^{(n)} = \infty$.*
(iii) *For $p = 1, \dots, N$, we have*

$$A_{p,p}^{(n)} = A_R^{ⓝ}. \tag{10}$$

Proof

(i) We will prove that matrix $A^{ⓝ} = \left[A_{p,q}^{(n)}\right]$ has the given structure using mathematical induction. The statement is obviously true for $n = 1$. Assume that it is true for $n = k$. Then for $p - q = p_1 - q_1$ we have

$$A_{p,q}^{(k)} = A_{p_1,q_1}^{(k)}.$$

Assume that $p - q = p_1 - q_1$. According to Lemma 4 we have

$$A^{(k+1)}_{p,q} = A^{(k)}_{p,q}A_R \oplus A^{(k)}_{p,q-1}A_C$$
$$A^{(k+1)}_{p_1,q_1} = A^{(k)}_{p_1,q_1}A_R \oplus A^{(k)}_{p_1,q_1-1}A_C.$$

According to our hypothesis $p - q = p_1 - q_1$, and induction hypothesis that $p - q = p_1 - q_1$ implies $A^{(k)}_{p,q} = A^{(k)}_{p_1,q_1}, p - q + 1 = p_1 - q_1 + 1$ implies $A^{(k)}_{p,q-1} = A^{(k)}_{p_1,q_1-1}$, we have

$$A^{(k+1)}_{p,q} = A^{(k)}_{p,q}A_R \oplus A^{(k)}_{p,q-1}A_C = A^{(k)}_{p_1,q_1}A_R \oplus A^{(k)}_{p_1,q_1-1}A_C = A^{(k+1)}_{p_1,q_1}$$

which finishes inductive proof.

(ii) The statement is obviously true for $n = 1$. If $p > q$, using (9), and the fact that $x \odot \infty = \infty$, the submatrices right below the main diagonal are

$$A^{(n)}_{p,q} = A^{(n-1)}_{p,q}A_R \oplus A^{(n-1)}_{p,q-1}A_C = (\infty \odot A_R) \oplus (\infty \odot A_C) = \infty.$$

(iii) The statement is obviously true for $n = 1$. Using previous statement, we have

$$A^{(n)}_{p,p} = A^{(n-1)}_{p,p}A_R \oplus A^{(n-1)}_{p,p-1}A_C = \left(A_R^{(n-1)} \odot A_R\right) \oplus (\infty \odot A_C) = A_R^{ⓝ} . \qquad \square$$

Lemma 6. *The shortest path matrix $S(\mathbf{G})$, of graph $\mathbf{G}$ whose adjacency matrix $A(\mathbf{G}_E)$ is given with (8) is of the following form*

$$S(\mathbf{G}_E) = \begin{bmatrix} S_0 & S_1 & S_2 & \cdots & S_{n-1} \\ \infty & S_0 & S_1 & & S_{n-2} \\ \infty & \infty & S_0 & & S_{n-3} \\ \vdots & & & \ddots & \\ \infty & \infty & \infty & & S_0 \end{bmatrix}_{nxn} . \tag{11}$$

Proof. According to Lemma 5, we can represent matrix $A^{ⓝ}$ in the following form $A^{ⓝ} = \left[A^{(n)}_{q-p}\right]$, where we used notation $A^{(n)}_{q-p} = A^{(n)}_{p,q}$ since all $A^{(n)}_{p,q}$ are the same for the same difference $q - p$, and the same n. Using Lemma 3, we have

$$S(\mathbf{G}_E) = \bigoplus_{n=1}^{+\infty} A^{ⓝ} = \bigoplus_{n=1}^{+\infty}\left[A^{(n)}_{q-p}\right] = \left[\bigoplus_{n=1}^{+\infty} A^{(n)}_{q-p}\right] = \left[S_{q-p}\right].$$

For $p > q$ we have $A^{(n)}_{q-p} = \infty$, and accordingly

$$S_{q-p} = \bigoplus_{n=1}^{\infty} A^{(n)}_{q-p} = \bigoplus_{n=1}^{\infty} \infty = \infty.$$

$\square$

Theorem 1. *The submatrix S_m of the shortest paths matrix $S(\mathbf{G})$ can be computed using an iterative formula*

$$S_m = S_{m-1}A_C, \tag{12}$$

where starting matrix for $m = 0$ is the shortest paths submatrix that lays on the diagonal of the matrix S, i.e. for $p = q$

$$S_0 = \bigoplus_{i=1}^{\infty} A_R^{ⓘ} . \tag{13}$$

Proof. Elements of the submatrix S_m, according to Lemma 3, are

$$S_m = \bigoplus_{i=1}^{\infty} A^{(i)}_m. \tag{14}$$

For $m = 0$ we have

$$S_0 = \bigoplus_{i=1}^{\infty} A^{(i)}_0.$$

As $A^{(1)}_0$ has zeros on the main diagonal, it follows that S_0 has zeros on the main diagonal as well. Hence, $S_0 \oplus I = S_0$.

Substituting (9) into (14) we obtain

$$\begin{aligned} S_m &= \bigoplus_{i=1}^{\infty}\left(A_m^{(i-1)}A_R \oplus A_{m-1}^{(i-1)}A_C\right) \\ &= \bigoplus_{i=1}^{\infty}A_m^{(i-1)}A_R \oplus \bigoplus_{i=1}^{\infty}A_{m-1}^{(i-1)}A_C \\ &= \bigoplus_{i=0}^{\infty}A_m^{(i)}A_R \oplus \bigoplus_{i=0}^{\infty}A_{m-1}^{(i)}A_C \\ &= \left(A_m^{(0)}A_R \oplus \bigoplus_{i=1}^{\infty}A_m^{(i)}A_R\right) \\ &\oplus \left(A_{m-1}^{(0)}A_C \oplus \bigoplus_{i=1}^{\infty}A_{m-1}^{(i)}A_C\right) \end{aligned}$$

For $m \geqslant 2$ we have $A_m^{(0)} = A_{m-1}^{(0)} = \infty$, so that previous equation becomes

$$S_m = S_m A_R \oplus S_{m-1}A_C. \tag{15}$$

Let $m = 1$. We have

$$S_1 = S_1A_R \oplus S_0A_C \oplus A_C = S_1A_R \oplus (S_0 \oplus I)A_C = S_1A_R \oplus S_0A_C,$$

so that we obtain previous equation for $m = 1$. Let $m = 0$. We have

$$S_0 = S_0A_R \oplus S_{-1}A_C \oplus A_R = S_{-1}A_C \oplus (S_0 \oplus I)A_R = S_0A_R \oplus S_{-1}A_C,$$

so that we obtain previous equation for $m = 0$.

As shortest paths matrix remains the shortest paths matrix after tropical matrix product, thus Eq. (15) can be rewritten as

$$S_m = S_{m-1}A_C. \tag{16}$$

which proves Eq. (12).

The submatrix S_0 from (13) can be obtained from $A_{p,p}^{(n)} = A_R^{\textcircled{n}}$, substituting (10) into (14). □

Theorem 1 gives a simple iterative formula for the shortest paths matrix computation of a given regular systolic array, which presents the error distances between nodes in the architecture. Color-coded ESM of regular systolic array can be simply obtained by rewriting the corresponding error distances of the matrix $S(\mathbf{G}_E)$ from (11) into a matrix form that fits the topology of the array. The computation of CCESM will be illustrated in the next section on the example of PDT hexagonal systolic multiplier design.

5. Example of PDT hexagonal systolic multiplier design

A hexagonal 4-bit integer systolic multiplier is shown in Fig. 2. Fig. 2a shows the multiplication algorithm. The systolic array is shown in Fig. 2b, while 2c illustrates functions of a node.

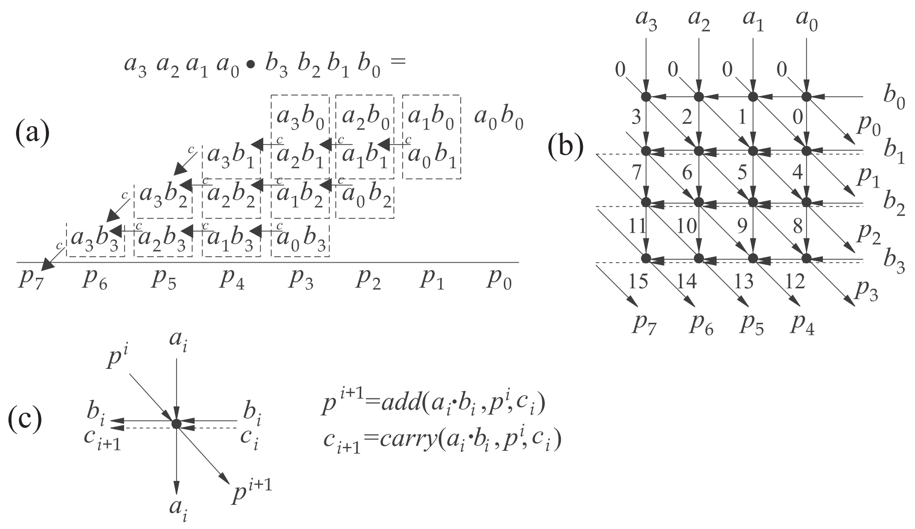


Fig. 2. Hexagonal 4-bit integer systolic multiplier: (a) multiplication algorithm, (b) hexagonal systolic array for multiplication, (c) functions of a node of the systolic array.

A suitable metric for the design of the PDT hexagonal systolic multiplier is Euclidian metric, in which the system fault can be defined as

$$d_E(A_t, A_{err}) > \Delta_E = 2^{L-\alpha} - 1, \tag{17}$$

where L is a total bits in the system output, and α represents allowed error threshold. In other words, if $\alpha = L$, then all bits must be correct, and if $\alpha = 0$, then the system can tolerate any error magnitude.

Let the potential error source be the combinatoric logic of the node. Hence, the erroneous product p^{i+1} can propagate and cause the error of the same magnitude in the node within the next row, i.e. $\Delta = 2^0$. If error occurs in the carry bit, then the error will cause an error in the partial product with $\Delta = 2^1$ greater weight than the weight of the node that produced the error. The EPG **G** is shown in Fig. 3. The edge labels in Fig. 3 represent the exponents of the errors, i.e. $\Delta = 2^{w_{i,j}}$.

We will give the general form of the submatrices S_m of the EPG from Fig. 3, in the respect to the Eq. (11), within the following lemma.

Lemma 7. *The submatrix S_m of the shortest paths matrix $S(\mathbf{G_E})$ of graph $\mathbf{G_E}$ from Fig. 3, has the following form*

$$S_m = \begin{bmatrix} m & m+1 & \cdots & m+N-1 \\ \vdots & & & \\ 2 & 3 & & 2+N-1 \\ 1 & 2 & & 1+N-1 \\ 0 & 1 & & N-1 \\ \vdots & & & \\ \infty & \infty & \cdots & m \end{bmatrix}_{NxN}, \tag{18}$$

i.e., the elements of the matrix S_m are

$$s^m_{i,j} = \begin{cases} j - i + m, & i \leqslant j + m \\ \infty, & other \end{cases}. \tag{19}$$

Proof. Adjacency matrix $A(\mathbf{G}_E)$ of the graph $\mathbf{G}_E$ shown in Fig. 3 is

$$A = \begin{array}{c|cccc|cccc|cccc|cccc|}
 & v_0 & v_1 & v_2 & v_3 & v_4 & v_5 & v_6 & v_7 & v_8 & v_9 & v_{10} & v_{11} & v_{12} & v_{13} & v_{14} & v_{15} \\
v_0 & 0 & 1 & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty \\
v_1 & \infty & 0 & 1 & \infty & 0 & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty \\
v_2 & \infty & \infty & 0 & 1 & \infty & 0 & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty \\
v_3 & \infty & \infty & \infty & 0 & \infty & \infty & 0 & 1 & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty \\
\hline
v_4 & \infty & \infty & \infty & \infty & 0 & 1 & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty \\
v_5 & \infty & \infty & \infty & \infty & \infty & 0 & 1 & \infty & 0 & \infty & \infty & \infty & \infty & \infty & \infty & \infty \\
v_6 & \infty & \infty & \infty & \infty & \infty & \infty & 0 & 1 & \infty & 0 & \infty & \infty & \infty & \infty & \infty & \infty \\
v_7 & \infty & \infty & \infty & \infty & \infty & \infty & \infty & 0 & \infty & \infty & 0 & 1 & \infty & \infty & \infty & \infty \\
\hline
v_8 & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & 0 & 1 & \infty & \infty & \infty & \infty & \infty & \infty \\
v_9 & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & 0 & 1 & \infty & 0 & \infty & \infty & \infty \\
v_{10} & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & 0 & 1 & \infty & 0 & \infty & \infty \\
v_{11} & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & 0 & \infty & \infty & 0 & 1 \\
\hline
v_{12} & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & 0 & 1 & \infty & \infty \\
v_{13} & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & 0 & 1 & \infty \\
v_{14} & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & 0 & 1 \\
v_{15} & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & 0
\end{array} \tag{20}$$

According to (8) the matrices A_R and A_C are

$$A_R = \begin{bmatrix} 0 & 1 & \infty & \cdots & \infty & \infty \\ \infty & 0 & 1 & & \infty & \infty \\ \infty & \infty & 0 & & \infty & \infty \\ \vdots & & & \ddots & & \\ \infty & \infty & \infty & & 0 & 1 \\ \infty & \infty & \infty & & \infty & 0 \end{bmatrix}_{NxN} \quad A_C = \begin{bmatrix} \infty & \infty & \infty & \cdots & \infty & \infty \\ 0 & \infty & \infty & & \infty & \infty \\ \infty & 0 & \infty & & \infty & \infty \\ \vdots & & & \ddots & & \\ \infty & \infty & \infty & & \infty & \infty \\ \infty & \infty & \infty & & 0 & 1 \end{bmatrix}_{NxN}. \tag{21}$$

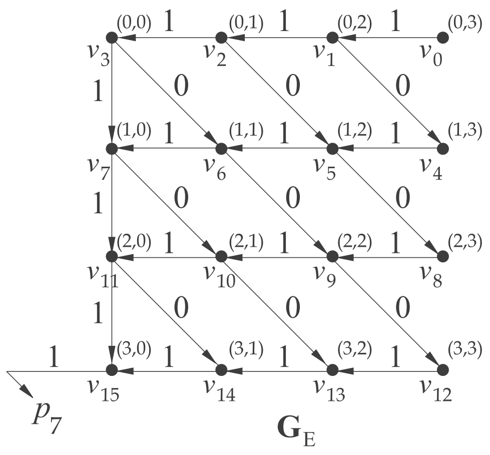


Fig. 3. Error propagation graph of hexagonal 4-bit integer multiplier array for Euclidian metric.

Using the definition of tropical matrix product (4) and the form of the matrix A_R given in Eq. (21), the elements of the matrix $A_R^{ⓝ} = [r_{i,j}^{(n)}]$ are

$$r_{i,j}^{(n)} = \begin{cases} j-i, & i \leqslant j \\ \infty, & \text{other} \end{cases}. \tag{22}$$

We will prove the lemma by mathematical induction. Substituting (22) into (13) we obtain

$$s_{i,j}^0 = \begin{cases} j-i, & i \leqslant j \\ \infty, & \text{other} \end{cases}, \tag{23}$$

which is equal to (19) for $m = 0$.

Let the following expression stands

$$s_{i,j}^k = \begin{cases} j-i+k, & i \leqslant j+k \\ \infty, & \text{other} \end{cases}. \tag{24}$$

Using Theorem 1 we obtain $S_{k+1} = S_k A_C$, i.e, from the definition of matrix multiplication in the tropical algebra (4) we obtain

$$s_{i,j}^{k+1} = \bigoplus_{l=1}^{N} s_{i,l}^k \odot c_{l,j}, \tag{25}$$

where $c_{i,j}$ is an element of the matrix A_C on the position (i,j). The element $c_{l,j} = 0$ for $l = j+1$, in each column of the matrix A_C except the last, the tropical sum (25) can be reduced to only one sum member, namely

$$s_{i,j}^{k+1} = s_{i,j+1}^k \odot 0 = s_{i,j+1}^k = \begin{cases} (j+1)-i+k, & i \leqslant (j+1)+k \\ \infty, & \text{other} \end{cases}.$$

For the last column, where $c_{N,N} = 1$, the sum (25) becomes

$$s_{i,N}^{k+1} = \bigoplus_{l=1}^{N} s_{i,l}^k \odot c_{l,N} = s_{i,N}^k \odot 1 = \begin{cases} (N-i+k)+1, & i \leqslant N+k \\ \infty, & \text{other} \end{cases}.$$

As index i is always less than N, and $k \geqslant 0$, none of the elements from the last column of $s_{i,N}^{k+1}$ is equal to ∞. Thus, the last equation can be rewritten as

$$s_{i,N}^{k+1} = \begin{cases} (N-i+k)+1, & i \leqslant N+k+1 \\ \infty, & \text{other} \end{cases} = (N-i+k)+1,$$

which proves the lemma □

The shortest paths matrix contains all shortest paths within the graph **G** from Fig. 3. Only the paths that end to the most significant output vertex, which is denoted as v_{15} in Fig. 3, are of the interest in forming the CCESM. The weights of those paths will indicate the error influence of particular node into the overall system output. Let CCESM be a matrix $M_{CC} = [m_{i,j}^{CC}]$, where the value of the element $m_{i,j}^{CC}$ represents the influence of node (i,j) on the output word, in the case of error. From (18) it can be shown that for the architecture from Fig. 2 the following lemma stands.

Lemma 8. *The elements of the matrix $M_{CC} = [m^{CC}_{i,j}]$ of graph $\mathbf{G}_E$ from Fig. 2 are of the following form*

$$m^{CC}_{i,j} = N + i - j, \tag{26}$$

where $(i,j) \in \{0,1,\ldots,N-1\}^2$.

For example, the CCESM of the architecture from Fig. 2b is

$$M_{CC} = \begin{bmatrix} 4 & 3 & 2 & 1 \\ 5 & 4 & 3 & 2 \\ 6 & 5 & 4 & 3 \\ 7 & 6 & 5 & 4 \end{bmatrix}. \tag{27}$$

In words, $m^{CC}_{0,3} = 1$ means that defective array node on corresponding position will cause an error $\Delta_E = 2^1$ in output word, while defective node v_7 (Fig. 3), where $m^{CC}_{2,1} = 5$ will cause the error $\Delta_E = 2^5$ in output word.

Information contained within the CCESM can be used to visualize vulnerable part of the array in the form of ESM for given error threshold. Let the PDT partition of the architecture be described using regular ESM matrix $M_\alpha = m_{i,j}$, where $m_{i,j} = 1$ if corresponding array node belongs to the vulnerable part of the architecture, and $m_{i,j} = 0$ if not. From (17) and (26) we obtain

$$m_{i,j} = \begin{cases} 1, & \alpha \geqslant N - i + j \\ 0, & \text{other} \end{cases}. \tag{28}$$

For example, if the systolic array from Fig. 2b is tolerant only to errors less than 2^6, then $\alpha = 2$, and the vulnerable part of the architecture is obtained from (27) as

$$M_2 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix}. \tag{29}$$

The PDT architecture that is tolerant to errors less than 2^6 is shown in Fig. 4. Gray nodes $11, 14$, and 15 are FT nodes.

6. Complexity reduction analysis

The CCESM given in (27) could be obtained using traditional algorithms, as well. One of well-known algorithms is Floyd–Warshall algorithm [19]. Let $s^k_{i,j}$ be the cost of the path from the node v_i to the node v_j that passes through exactly k other nodes. Floyd–Warshall algorithm is a recursive algorithm of the following form

$$s^k_{i,j} = \begin{cases} w_{i,j}, k = 0, \\ \min\left(s^{k-1}_{i,j}, s^{k-1}_{i,k} + s^{k-1}_{k,j}\right), \text{other} \end{cases}.$$

Let T be the total number of array cells ($T = N^2$). The Floyd–Warshall algorithm compares all possible paths through the graph between each pair of vertices. It is able to do this with $\mathcal{O}(T^3)$ comparisons.

Using tropical iterative algorithm proposed in Theorem 1, the number of required comparisons can be significantly reduced.

Lemma 9. *The number of required comparisons for iterative algorithm given by Eqs. (12) and (13) is $\mathcal{O}(T^2)$.*

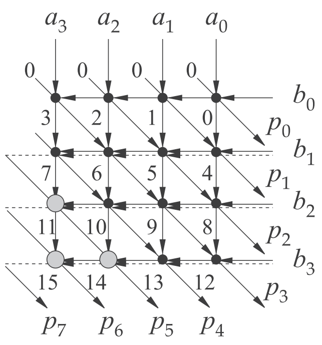


Fig. 4. PDT hexagonal systolic multiplier tolerant to errors less than 2^6.

Proof. First, we will obtain the number of comparisons required to compute the matrix S_0, given in (13). The total number of rows and columns in matrix A_R, according to (21), is N. The number of required comparisons for computation of one element of the matrix $A_R^{②}$, according to (4), is N. The total number of elements within the matrix is N^2, thus, computation of $A_R^{②}$ requires $N \cdot N^2 = N^3$ comparisons.

In order to obtain S_0, according to (13), the Nth tropical power of the matrix A_R is required, so previously analyzed computation should be repeated $N-1$ times, which gives the total of $(N-1) \cdot N \cdot N^2$ comparisons.

The matrix S_1 can be obtained, according to (12), as tropical product of matrices S_0 and A_C, which requires N^3 comparisons. The matrices $S_i, i = 2, 3, \ldots, N$, can be obtained in the same manner, requiring $(N-1) \cdot N^3$ comparisons.

The total number of comparisons is the number of comparisons required for computation of S_0, plus the number of comparisons required for computation of the matrices $S_i, i = 1, 2, \ldots N$, i.e.

$$(N-1)N^3 + (N-1)N^3 = 2(N-1)N^3,$$

which proves the lemma. □

The lemma shows that complexity of the proposed algorithm is slightly more than T times reduced, which can be significant, especially in nanoarchitectures with 10^8 and more components [2].

The computational complexity in obtaining error significance maps is reduced in comparison to traditional algorithms by customizing the topology of starting graph in tropical algebra according to the typical topology of systolic arrays. The fact that systolic arrays have connected cells in neighboring rows and columns only is used in (8), and built in (10), which led to the simple iterative algorithm (12).

We implemented both Floyd–Warshall and proposed algorithm in C programing language, using *gcc* 4.6 compiler on Linux 2.6 kernel, and executed them on Intel P4@3.0 GHz with 512 MB of RAM, for different values of parameter N, while measuring the execution time. The obtained results are given in Table 1.

7. FPGA implementation results

In order to illustrate the tradeoffs enabled by partial application of fault tolerance on the array presented in Section 5, we choose a TMR as a FT scheme to be applied in PDT hexagonal systolic multiplier implementation.

For the sake of comparison, the PDT multiplier (Fig. 4), is described in VHDL as parameterized core and implemented on Virtex4 FPGA. Equations (12) and (13) are implemented as VHDL functions, which in conjunction with parameter α return ESM (28). Using the design automation described in VHDL, the only parameter which has to be set, regardless the array dimensions, is the error threshold (α).

We implemented three arrays with different parameter sets. The parameter sets for the implemented arrays are given in Table 2. The arrays differ in number of columns/rows (N), and in error threshold. For each implemented array three parameters are given: the number of cells (#), FPGA resources usage, given in kilogates (kG), and the percentage of cells that belong to the vulnerable part of the array (%). For example, if the array consists of $N = 8$ rows, then the total number of the result bits is $L = 16$. If the error threshold is $\alpha = 0$, then the total number of required cells is $8 \cdot 8 = 64$ (Table 2). If the error threshold is $\alpha = 4$, then 0.16% of the array belongs to the vulnerable part of the array. Thus, if we apply triple modular redundancy to the vulnerable part of the array, we obtain the array with $(0.16 \cdot 64) \cdot 3 + (1 - 0.16) \cdot 64 \cdot 1 = 84$ cells (Table 2).

Let the probability of having a defective cell within the multiplier arrays from Table 2 be denoted as p, and let T be the total number of the array cells ($T = N^2$). In the same manner as proposed in [11] we can obtain function that shows probabilities for different array sizes where PDT design can introduce the yield improvement (Y_{PDT}) in comparison to the regular array, for the same error tolerance (Y_{ET}). The function is given in Fig. 5. From Fig. 5 it can be noticed that PDT design is preferable as the number of array cells is increased.

Table 1
Comparison of times required for execution of the well-known Floydv–Warshall and the proposed iterative algorithm.

N [ms]	4	16	256	1024
F-W alg	0.005	0.079	387.195	22,641.609
Proposed	0.002	0.009	1.676	25.916

Table 2
Implementation results of hexagonal multipliers with different setting of α.

N	$\alpha/L = 0$		0.25			0.50			0.75			1.0	
	#	Res. [kG]	%	#	Res. [kG]	%	#	Res. [kG]	%	#	Res. [kG]	#	Res. [kG]
4	16	0.35	0.19	22	0.40	0.62	36	0.53	0.93	46	0.59	48	0.60
8	64	1.36	0.16	84	1.93	0.56	136	2.17	0.90	180	2.33	192	2.40
16	256	5.19	0.14	328	5.75	0.53	528	7.46	0.89	712	9.06	768	9.55

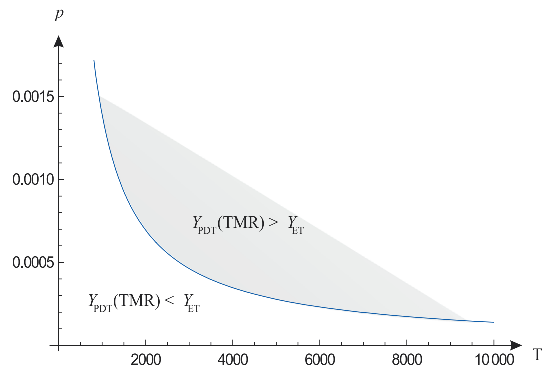


Fig. 5. A function which gives the defect probability for which the PDT design can introduce yield improvement.

8. Conclusions

In this paper we proposed the application of tropical algebra in systolic array analysis, due to the fact that it is more descriptive and convenient to use in graph analysis than traditional algebra. We investigated the properties of tropical algebra as a mathematical framework in error propagation analysis in systolic arrays. Using the properties of systolic arrays in tropical algebra, we developed simple iterative algorithm for shortest path computation of regular systolic arrays. It is shown that the computational complexity of the proposed algorithm is reduced from $\mathcal{O}(T^3)$ to $\mathcal{O}(T^2)$, where T is the number of array cells. The results are used to form color-coded error significance map of the array, which contains all information about defects influence in output result. The proposed technique gives a general solution for computation of defect influence in output result of array topologies. The proposed method was defined and described in details through the set of definitions and lemmas. An example of tropical algebra analysis and design of partially defect tolerant hexagonal systolic multiplier were given. The execution speed of the proposed algorithm was given, as well as VHDL/FPGA implementation of basic multiplier and partially defect tolerant multiplier, designed using the proposed method.

Acknowledgements

The research was supported in part by the Serbian Ministry of Education and Science (Projects TR32012, 174015 and III44006).

References

[1] G.E. Moore, Cramming more components into integrated circuits, Electronics 38 (8) (1965).
[2] Michael Haselman, Scott Hauck, The future of integrated circuits: a survey of nanoelectronics, Proceedings of the IEEE 98 (1) (2010) 11–38.
[3] ITRS, International Technology Roadmap for Semiconductors, 2010. <http://www.itrs.net/links/2010itrs/home2010.htm>.
[4] M. Mishra, S.C. Goldstein, Defect tolerance at the end of the roadmap, International Test Conference Proceedings 1 (2003) 1201–1210.
[5] Christopher Brown, Ulrich Jonas, Jon Preece, Helmut Ringsdorf, Markus Seitz, Fraser Stoddart, Introduction of Catenanes into Langmuir Film and Langmuir-Blodgett Multilayers, A Possible Strategy for Molecular Information Storage Materials, vol. 16, no. 4, 2000, pp. 1924–1930.
[6] Jude Rivers, Prabhakar Kudva, Reliability challenges and system performance at the architecture level, in: Design & Test of Computers, 6, vol. 26, IEEE, 2009, pp. 62–73.
[7] Y. Chen, Gun-Young Jung, Douglas Ohlberg, Xuema Li, Duncan Stewart, Jan Jeppesen, Kent Nielsen, Fraser Stoddart, Stanley Williams, Nanoscale molecular-switch crossbar circuits, Nanotechnology 14 (4) (2003) 462–468.
[8] A. Al-Yamani, S. Ramsundar, D. Pradhan, A defect tolerance scheme for nanotechnology circuits, IEEE Transactions on Circuits and Systems I: Regular Papers 54 (11) (2007) 2402–2409.
[9] C. Moritz, Teng Wang, P. Narayanan, M. Leuchtenburg, Yao Guo, C. Dezan, M. Bennaser, Fault-tolerant nanoscale processors on semiconductor nanowire grids, IEEE Transactions on Circuits and Systems I: Regular Papers 54 (11) (2007) 2422–2437.
[10] Vladimir Ćirić, Jelena Kolokotronis, Ivan Milentijević, Partial error-tolerance for bit-plane FIR filter architecture, International Journal of Electronics and Communication 63 (2009) 398–405.
[11] Vladimir Ćirić, Aleksandar Cvetković, Ivan Milentijević, Yield analysis of partial defect tolerant bit-plane array, International Journal of Computers and Mathematics with Applications 59 (1) (2010) 98–107.
[12] A. Menti, T. Zacharias, J. Milias-Argitis, Geometric algebra: a powerful tool for representing power under nonsinusoidal conditions, IEEE Transactions on Circuits and Systems I: Regular Papers 54 (3) (2007) 601–609.
[13] Jean-Eric Pin, Tropical Semirings, Idempotency, Publ. Newton Inst., vol. 11, Cambridge Univ. Press, Cambridge, 1998. pp. 50–69.
[14] Diane Maclagan, Bernd Sturmfels, Introduction to Tropical Geometry, University of Warwick, U.K., Nov. 2009.
[15] A. DeHon, Reconfigurable computing: the theory and practice of fpga-based computing, in: S. Hauck, A. DeHon (Eds.), Defect and Fault Tolerance, Morgan Kaufmann, San Francisco, US, 2008.
[16] M. Breuer, Multimedia applications and imprecise computation, in: Proceedings on the 8th Euromicro Conference on Digital System Design, Euromicro, Porto, Portugal, September 2005, 0–7695-2433-8/05.
[17] Vladimir Ćirić, Ivan Milentijević, Configurable folded array for FIR filtering, Journal of Systems Architecture, An International Journal 54 (1–2) (2008) 177–196.
[18] Imre Simon, Recognizable sets with multiplicities in the tropical semiring, Mathematical Foundations of Computer Science, Lecture Notes in Comput. Sci., vol. 324, 1998, pp. 50–69.
[19] Robert Floyd, Algorithm 97: shortest path, Communications of the ACM 5 (6) (June 1962) 345.