2008 26th INTERNATIONAL CONFERENCE ON MICROELECTRONICS

PROCEEDINGS

Volume 2

NIŠ

MIEL 2008

Niš, Serbia
11-14 May 2008

organized by

IEEE Serbia and Montenegro Section - ED/SSC Chapter

under the co-sponsorship of

IEEE Electron Devices Society

with the cooperation of

IEEE Solid-State Circuits Society

IEEE Catalog No. CFP08432-PRT
ISBN 978-1-4244-1881-7

Error Significance Map for Bit-plane FIR Filtering Array

Jelena Kolokotronis, Vladimir Ćirić, Ivan Milentijević

Abstract— **Some applications require correct computation, while many others do not. Large domain where perfect functional performance is not always required is in multimedia and DSP systems. Relaxing the requirement of 100% correctness for devices and interconnections may dramatically reduce costs of manufacturing, verification, and testing. The goal of this paper is development of error significance map for bit-plane FIR filtering array. The map marks the part of the array that must be error-free in order to enable computing on the bit-plane array with acceptable results. In other words the array cells, out of the marked area, could produce errors, but without significant influence on the marked high order bits of the resulting word. The bit-plane array operates on a bit level and assumes accumulation throughout the array with sum and carry propagation. It means that derivation of the error significance map is not a trivial for design automation. In this paper we propose a rigorous mathematical path based on transitive closure that generates error significance map for the bit-plane array.**

I. Introduction

As scaling approaches the physical limits of devices and fabrication technology, designers will increasingly have to consider qualitative changes. Key concerns include increasing process variations, defect rates, and infant mortality rates [1]. As VLSI scaling continues along its traditional path, we will soon be in a situation where chips will have billions of devices and thousands of defects [2]. Relaxing the requirement of 100% correctness for devices and interconnections may dramatically reduce costs of manufacturing, verification, and testing. Such a paradigm shift is in any case forced by technology scaling, which leads to more transient and permanent failures of signals, logic values, devices, and interconnections [3].

Most multimedia devices process signals, such as images and sound, that are eventually consumed by people. Designers take advantage of the signal processing ability of people to convert the original source of signals to lower quality packets of information, since this usually provides acceptable performance to the end user, reduces bandwidth and hardware costs [4]. An interesting question is: if some signal processing device has a minor hardware defect, will it still produce results that are good enough for the end user? Then, maybe, they too could be sold to the public rather than being discarded [5]. Fault tolerance (FT) is the ability of a system to continue *correct* operation of its tasks after the occurrence of hardware or software faults [6]. On the other hand, circuit is error-tolerant (ET) with respect to an application, if (1) it contains defects that cause internal and may cause external *errors*, and (2) the system that incorporates this circuit produces *acceptable results* [1],[5]. There are passive and active measures of system degradation due to errors. Performance, capacity and throughput are said to be passive measures, while error Rate, Accumulation and Significance, known as RAS, are active measures [1].

The bit-plane (BP) FIR filter is semi-systolic architecture with bit-plane operations that provides regular connections with extensive pipelining and high computational throughput [7]-[9]. The BP architecture, due to regularity, is taken as a basis for synthesis of error tolerant bit-plane FIR filter.

In this paper we will investigate possibilities of error tolerant FIR filter design based on BP array. Allowing the array cells to produce errors, the requirement of 100% correctness in computation is relaxed, and the concept of *acceptable results* is employed. Thus, in order to mark the part of the BP array according to the margin of acceptable error in output word, it is important to define an error impact of each array cell to the output result - *error significance map.* The goal of this paper is the development of a error significance map for bit-plane FIR filtering array. The array cells out of the area marked by error significance map could produce errors, but without significant influence on high order bits of the resulting word. The bit-plane array operates on a bit level and assumes accumulation throughout the array with sum and carry propagation. It means that the formation of the error significance map is not trivial for design automation. In order to design error tolerant array in a systematic way, which is suitable for design automation, we will propose a rigorous mathematical path based on transitive closure that generates error significance map for the bit-plane array.

II. Bit-plane FIR Filter Architecture

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

$$y_i = c_0x_i + c_1x_{i-1} + \ldots + c_{k-1}x_{i-k+1}, \qquad (1)$$

Jelena Kolokotronis, Vladimir Ćirić and Ivan Milentijević are with the Department of Computer Science, Faculty of Electronic Engineering, University of Niš, Aleksandra Medvedeva 14, 18000 Niš, Serbia, E-mail: milentijevic@elfak.ni.ac.yu

978-1-4244-1882-4/08/$25.00 © 2008 IEEE

where c_0, c_1,..., c_{k-1} are coefficients while $\{x_i\}$ are input words. Computation (1) can be realized in different manners. When high performances are required systolic arrays are frequently used. Semi-systolic array share with systolic arrays desirable simplicity and regularity properties in addition to their pipelining and multiprocessing schemes of operation.

The BP is semi-systolic architecture with bit-plane operations that provides regular connections with extensive pipelining and high computational throughput [7], [9]. Functional block diagram of BP array is shown in Fig. 1.

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

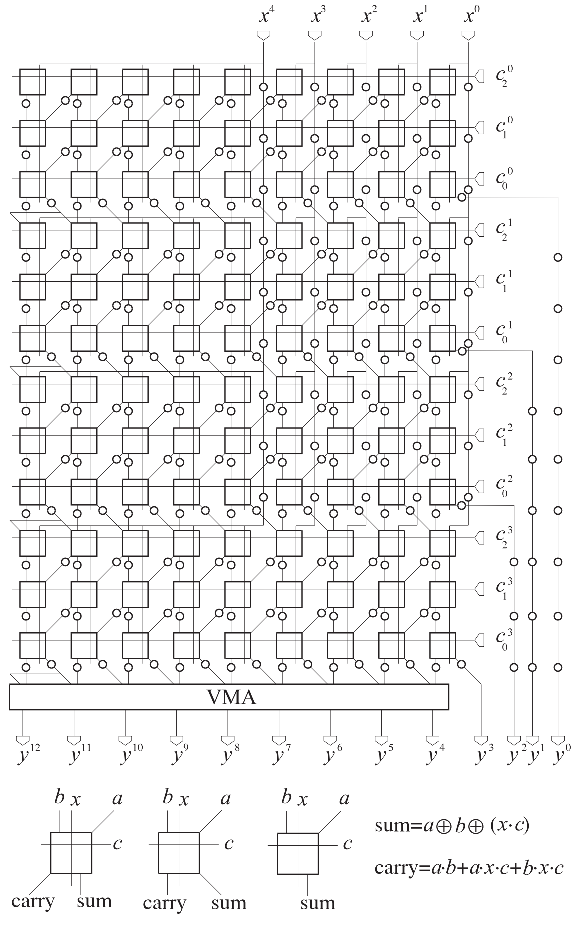


Fig. 1. The BP array for $k_C = 3$ and $m = 4$

There are m BP elements that form array shown in Fig. 1. Each BP (Fig. 1) is formed as a set of k_C rows. A row performs basic multiply-accumulate operation between intermediate result from previous row and product of input word and one coefficient bit. Delayed for one clock cycle per row, output word is available after $k_C \cdot m$ clock cycles.

III. Transitive Closure of Bit-plane Array

Error significance map marks the part of the array that must be error-free in order for BP array to produce acceptable results. The impact of error from particular cell on output result of BP array can be obtained using transitive closure, which gives information about all paths within the array. In order to clarify the error significance map development, we give a brief review of transitive closure.

Given a directed graph $\mathbf{G} = (\mathbf{V}, \mathbf{E})$, where $V = \{v_1, \dots, v_n\}$ is a finite set of vertices and E is a finite set of edges; An edge $e \in E$ is an ordered pair (v_i, v_j), where $v_i, v_j \in V$ and an edge (v_i, v_j) means that vertices v_i and v_j are connected. The transitive closure of $\mathbf{G}$ is defined as the Graph $\mathbf{G}^* = (\mathbf{V}, \mathbf{E}^*)$, where $E^* = \{(v_i, v_j) \mid \text{ there is a path from } v_i \text{ to } v_j \text{ in } \mathbf{G}\}$. The transitive closure of a graph is computed by computing the connectivity matrix $\mathbf{A}^*$. The connectivity matrix of $\mathbf{G}$ is a matrix $\mathbf{A}^* = (\mathbf{a}^*_{\mathbf{i,j}})$ such that $a^*_{i,j} = 1$ if there is a path from v_i to v_j or $i = j$, and $a^*_{i,j} = 0$ otherwise [10].

In order to develop a transitive closure for the BP array, both the directed graph $\mathbf{G}$ for the array from Fig. 1, and the connectivity matrix A have to be obtained. In the BP array shown in Fig. 1 there is a multiplication of partial product from each row within BP with factor 2, which is realized as a shift for one position to the left between rows. Between BPs there is a multiplication of a partial product by 1/2, i.e. shift for one position to the right (Fig. 1) [7]. In order to develop connectivity matrix for BP array in a general form, according to functional block diagram from Fig. 1, we obtained a directed graph $\mathbf{G}$ in a general form, shown in Fig. 2.

Graph $\mathbf{G}$ from Fig. 2 consists of two types of nodes: nodes that represent the BP cells from Fig. 1 (shaded nodes in Fig. 2), and fictive nodes that are added in order to obtain a graph with regular connections. In order to obtain connectivity matrix of graph $\mathbf{G}$ we form a directed graph $\mathbf{G_C}$ that shows the connectivity of columns of graph $\mathbf{G}$, and graph $\mathbf{G_R}$ that shows the connectivity of rows. Graphs $\mathbf{G_C}$ and $\mathbf{G_R}$ are shown in Figs. 3(a), and 3(b), respectively.

Graph from Fig. 3(a) can be represented by connectivity matrix as follows:

$$B = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 1 & 1 & 0 & \dots & 0 \\ 0 & 1 & 1 & \dots & 0 \\ & & & \vdots & \\ 0 & 0 & 0 & \dots & 1 \end{bmatrix}_{(l_0+m)\mathrm{x}(l_0+m)}$$

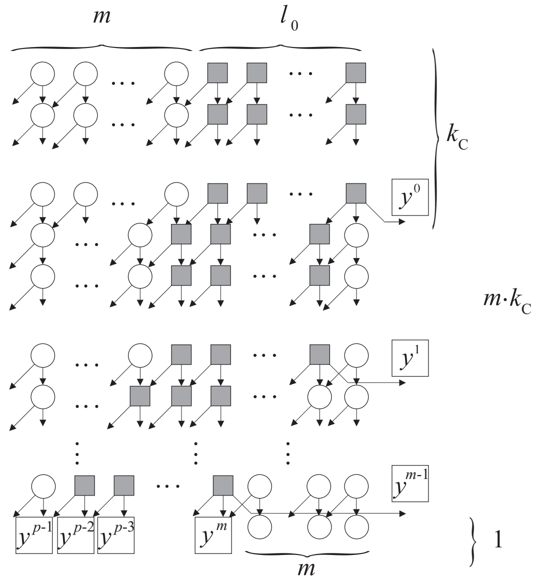


Fig. 2. Directed graph $\mathbf{G}$ for bit-plane array

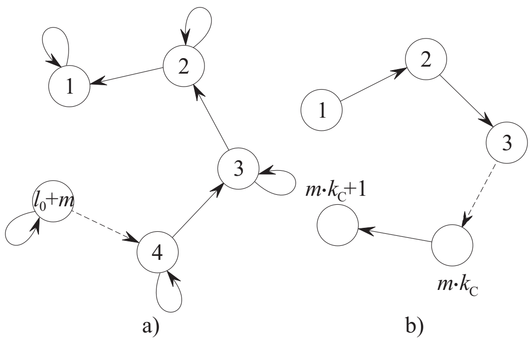


Fig. 3. Directed graph (a) $\mathbf{G_C}$ that shows the connectivity of columns of graph $\mathbf{G}$, and (b) $\mathbf{G_R}$ that shows connectivity of rows

Dimension of matrix B is $(l_0+m)\text{x}(l_0+m)$, where l_0+m is the number of columns of graph $\mathbf{G}$ (Fig. 2). The elements $b_{i,j}$ of matrix B are of the form:

$$b_{i,j} = \begin{cases} 1, & j+1 \geq i \geq j \\ 0, & \text{other} \end{cases}, \tag{2}$$

i.e., the elements on the main diagonal, and the elements below the main diagonal are equal to 1. Connectivity matrix A can be obtained from graph $\mathbf{G_R}$ (Fig. 3b), having in mind that each node in graph $\mathbf{G_R}$ has internal paths described with (2). Connectivity matrix A for directed graph $\mathbf{G}$ from Fig. 2 is:

$$A = \begin{bmatrix} 0 & B & 0 & \dots & 0 \\ 0 & 0 & B & \dots & 0 \\ & & & \vdots & \\ 0 & 0 & 0 & \dots & B \\ 0 & 0 & 0 & \dots & 0 \end{bmatrix}_{(m\cdot k_C+1)\text{X}(m\cdot k_C+1)} \tag{3}$$

Substituting sub-matrices B in (3) with (2), we derive a connectivity matrix A for graph $\mathbf{G}$ from Fig. 2. Transitive closure of matrix A is of the following form:

$$A^* = \begin{bmatrix} 0 & B & B^2 & \dots & B^{m\cdot k_C} \\ 0 & 0 & B & \dots & B^{m\cdot k_C-1} \\ & & & \vdots & \\ 0 & 0 & 0 & \dots & B \\ 0 & 0 & 0 & \dots & 0 \end{bmatrix}, \tag{4}$$

where $B^k = B^{k-1}\cdot B$. In order to obtain matrices B^k from transitive closure (4), we give the following lemma:

Lemma 1: Elements $(b^d_{i,j})$ of matrix B^d are of the form

$$b^d_{i,j} = \begin{cases} 1, & j+d \geq i \geq j \\ 0, & \text{other} \end{cases},$$

Proof Lemma 1 can be proven using mathematical induction. Elements $b^1_{i,j}$ of the matrix B^1 are of the form (2). From (2) it is straightforward to shown that Lemma 1 stands for B^2. In order to show that Lema 1 stands for B^{d+1}, we obtain $b^{d+1}_{i,j}$ as follows:

$$B^{d+1} = B^d \cdot B \Rightarrow b^{d+1}_{i,j} = \sum_{k=0}^{p} b^d_{i,k} \cdot b_{k,j} =$$

$$= \begin{cases} 1, & k+d \geq i \geq k \bigwedge j+1 \geq k \geq j \\ 0, & \text{other} \end{cases} =$$

$$= \begin{cases} 1, & j+d+1 \geq i \geq j \\ 0, & \text{other} \end{cases},$$

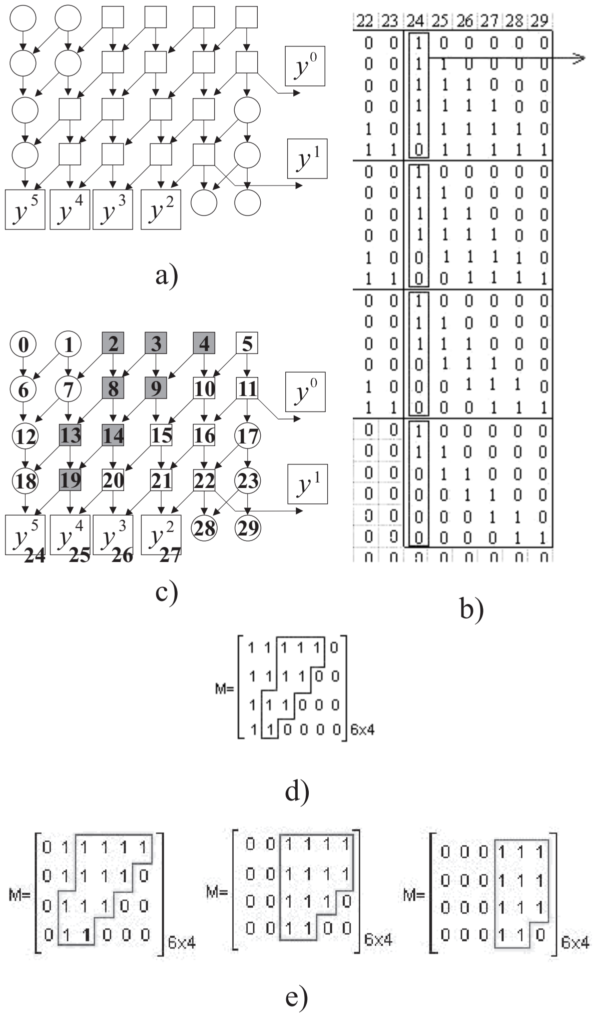
which proves the Lemma 1.

Transitive closure A^*, given with (4), is a square matrix with dimension $(m\cdot k_C+1)\cdot(m+l_0)\text{x}(\mathrm{m}\cdot\mathrm{k_C}+1)\cdot(\mathrm{m}+\mathrm{l_0})$, which shows the existence of the path between any two nodes of the graph $\mathbf{G}$ (Fig. 2). If element $a^*_{i,j}$ of matrix A^* is equal to 1, then path from node i to node j exists, while if it is equal to 0, the path does not exist.

IV. Example of Error Significance Map for Bit-plane Array

With the aim to illustrate error significance map development using a transitive closure of bit-plane array, we give an example of map development for BP array that has $k_C = 2$ coefficients, and coefficient length $m = 2$.

Directed graph $\mathbf{G}$ of bit-plane array with $k_C = 2$ coefficients, and coefficient length $m = 2$ is shown in Fig. 4(a). A transitive closure of the graph can be obtained according to (4) and Lemma 1. The dimensions of matrix A^* for the given example are 30x30 (enumerated as $0, 1, \dots, 29$), because there is a total of 30 nodes within the graph $\mathbf{G}$, including the output nodes y^5, y^4, y^3, and y^2. Fig. 4(b) shows the upper-right corner of transitive closure A^* of the graph $\mathbf{G}$ from Fig. 4(a).



22	23	24	25	26	27	28	29
0	0	1	0	0	0	0	0
0	0	1	1	0	0	0	0
0	0	1	1	1	0	0	0
0	0	1	1	1	1	0	0
1	0	1	1	1	1	1	0
1	1	0	1	1	1	1	1
0	0	1	0	0	0	0	0
0	0	1	1	0	0	0	0
0	0	1	1	1	0	0	0
0	0	1	1	1	1	0	0
1	0	0	1	1	1	1	0
1	1	0	0	1	1	1	1
0	0	1	0	0	0	0	0
0	0	1	1	0	0	0	0
0	0	1	1	1	0	0	0
0	0	0	1	1	1	0	0
1	0	0	0	1	1	1	0
1	1	0	0	0	1	1	1
0	0	1	0	0	0	0	0
0	0	1	1	0	0	0	0
0	0	0	1	1	0	0	0
0	0	0	0	1	1	0	0
0	0	0	0	0	1	1	0
0	0	0	0	0	0	1	1
0	0	0	0	0	0	0	0

$$M=\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}_{6x4}$$

$$M=\begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \end{bmatrix}_{6x4} \quad M=\begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \end{bmatrix}_{6x4} \quad M=\begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix}_{6x4}$$

Fig. 4. Implementation example: (a) directed graph **G** of bit-plane array with $k_C = 2$ coefficients, and coefficient length $m = 2$; (b) transitive closure of graph **G**; (c) error significance map of graph **G**; (d) error significance matrix for most significant bit of output result; (e) error significance matrices for different combinations of output word bits

A transitive closure, given in Fig. 4(b), shows the existence of all paths within the graph, thus, if we observe the 24^{th} column of transitive closure it will give the existence of path from each cell within the BP array to the most significant bit of output word y, (y^5). The nodes of the graph from Fig. 4(a) are enumerated as $0, 1, \ldots, 29$, as it is shown in Fig. 4(c). The path existence to 24^{th} node, i.e. the elements of the 24^{th} column, are rewritten in the form of a matrix, and mapped to the nodes of graph **G**, which is shown in Fig. 4(c). The nodes that have influence on the most significant bit of result are shaded. The fictive nodes, because of their nature, are not taken into consideration.

Fig. 4(d) shows the error significance map of bit-plane array, in respect to the most significant bit of the output result, in the form of matrix denoted as M. If acceptable results of BP array with $k_C = 2$, and $m = 2$ are defined as $Y_{acceptable} = Y_{correct} \pm (2^5 - 1)$, then matrix M from Fig. 4(d) shows the part of the bit-plane array which must be error-free in order for BP architecture to produce *acceptable results*.

Error significance maps concerning bits y^4, y^3, and y^2 of the output result are developed in the same manner, and are shown in Fig. 4(e).

V. Concluding Remarks

We derived the error significance map for the bit-plane array, which facilitates the design of error tolerant bit-plane FIR filter. Error tolerance is employed allowing the cells to produce errors. Thus, in order to mark the part of the BP array according to the margin of acceptable error in resulting word, we defined an error impact of each array cell to the result - *error significance map*. Starting the development with acceptable margins of error in output result, from transitive closure of the BP array we obtained the error significance map. The array cells out of the area marked by error significance map could produce errors, but without significant influence on high order bits of the resulting word. A rigorous mathematical path based on transitive closure that generates error significance map for the bit-plane array is proposed. General form of transitive closure of BP array is proven using mathematical induction. Obtaining the error significance map the error tolerant design automation of bit-plane array is disburdened.

References

[1] M. Breuer, S. Gupta, T. Mark, *Defect and Error Tolerance in -the Pressence of Massive Numbers of Defects*, IEEE Transactions on Design & Test of Computers, 2004, pp. 216-227.

[2] T.-Yu Hsieh, K.-Jong Lee, M. Breuer, *Reduction of detected acceptable faults for yield improvement via error-tolerance*, Proceedings of the conference on Design, automation and test in Europe, Nice, France, 2007, pp. 1599 - 1604.

[3] International Technology Roadmap for Semiconductors, 2001; http://public.itrs.net/.

[4] S. Momcilovic, N. Roma, L. Sousa, *Adaptive Motion Estimation Algorithm for H.264/AVC*, 15th International Conference on Digital Signal Processing (DSP) 2007, IEEE, Cardiff, Wales, UK, July 2007.

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

[6] B. W. Johnson, *Fault Tolerance*, The Electrical Engineering Handbook, CRC Press, 1993, Chapter 87.

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

[8] Y-C. Lin, F-C. Lin, *Classes of Systolic Arrays for Digital Filtering*, Int. J. Electronics, Vol. 70, No. 4, 1991, pp. 729-737.

[9] V. Ciric, I. Milentijevic,, *Configurable folded array for FIR filtering*, Journal of Systems Architecture, Elsevier, 2007, doi:10.1016/j.sysarc.2007.05.001

[10] S. Bawa, G.K. Sharma, *A Parallel Transitive Closure Computation Algorithm for VLSI Test Generation*, Lecture Notes in Computer Science, Volume 2367/2002, pp. 243-252.