ABSTRACT. The DIOGENES methodology produces designs for fault-tolerant VLSI processor arrays in two stages: First, the desired array is viewed as an undirected graph and is embedded in a book; then, the book embedding is converted to an efficient fault-tolerant layout of the array. We survey here work on both stages of the methodology, highlighting recent progress and pointing out remaining challenges.

1. INTRODUCTION

DIOGENES is a methodology for designing fault-tolerant VLSI arrays of identical processing elements (PEs, for short). The methodology operates in two stages: It takes a design problem and converts it to the problem of embedding graphs in books; it then converts the resulting graph embedding to a fault-tolerant layout. The methodology appeared first in [14], where its philosophy and basic configuring tools were developed. The methodology was abstracted to the book-embedding problem in [3], where an embedding heuristic is suggested and applied to obtain a variety of (near-)optimal embeddings.

The graph-embedding stage of the methodology is of interest in its own right. The problem of embedding graphs in books was studied first in [1], where one finds several rudimentary results and suggested directions for further research. [4] represents the first major effort to study the DIOGENES methodology in terms of the book-embedding problem. The first goal of this paper is to survey the major results on book embeddings, that are relevant to the DIOGENES methodology, highlighting recent breakthroughs and noting remaining challenges.

The embedding-to-layout stage of the DIOGENES methodology has received less attention thus far, but this is beginning to change [16]. The second goal of this paper is to describe a three-part project devoted to devising algorithms for converting a book embedding of a graph to an efficient fault-tolerant layout of the processor array represented by the graph. The subtasks of the project are: (1) to assign the $m$ vertices of the undirected graph to the $n$ fault-free PEs that have been fabricated; (2) to determine the extent to which strategically placed "shortcuts" in the physical arrangement of the fabricated PEs enhance the run-time efficiency of DIOGENES layouts; (3) to produce automatically, from the given book embedding, logic designs of the switching networks that configure the fault-free array.

2. THE DIOGENES DESIGN METHODOLOGY

2.1. The Approach Exemplified

We excerpt from [14]. The DIOGENES design methodology achieves tolerance to faults via the following scenario. One lays out his PEs in a (logical, but not necessarily physical) row, with some number of "bundles" of wires running above the row of PEs; all PEs are hooked into the bundles in the same format. One scans along the row of PEs testing which are faulty and which are fault-free. As each good PE is encountered, it is hooked into the bundles of wires through a network of switches, thereby connecting it to the fault-free PEs that have already...
been found and preparing it to connect to those that will be found. For illustration, one cell of a DIOGENES layout of the depth-4 complete binary tree is depicted in Fig. 1(a).

The lines above the PEs are the single bundle needed for the layout. The switches are controlled by two externally set variables, \( G_i \) which is high when PE\(_i\) is good and low when it is faulty, and \( L_i \) which is high when PE\(_i\) is to be a leaf of the tree and low otherwise.

The layout's single bundle has wires numbered 1 to 4. As one encounters a good PE that is to be a leaf of the tree, the PE is connected to line 1, thereby preparing it to connect to its father in the tree; simultaneously lines 1, 2, and 3 "shift up" to "become" lines 2, 3, and 4, respectively; switches disconnect the left parts of the lines from the right parts so node-to-node connectivity remains correct; see Fig. 1(c). The bundle has thus behaved like a stack being PUSHed. A good PE that is to be a nonleaf of the tree is connected to the bundle in two stages. First, it is connected to lines 1 and 2 of the bundle, thereby connecting it to its sons in the tree; simultaneously, lines 3 and 4 "shifts down" to "become" lines 1 and 2, respectively; again switches maintain proper node-to-node connectivity; see Fig. 1(d). The bundle has here behaved like a stack being POPped. Second, the PE PUSHes a connection onto the stack, to prepare for eventual connection to its father in the tree.

As in this example, the DIOGENES methodology attempts to simplify both the machinery and the process required to configure the wire bundles in the face of faults, by organizing all bundles as stacks (or as queues, cf. [11, 13]). Such organization minimizes the number of control bits needed to set the switches that configure the array.

We emphasize that the logical linearization of the PEs demanded by the methodology need not be realized by physical linearization: In [15] we noted that the run-time efficiency of DIOGENES designs could be enhanced by adding shortcuts to the row of PEs, as in Fig. 2, so that signals need not pass over every long stretch of faulty PEs.

2.2. The Approach Generalized via Graph Embedding

Chung, Leighton, and Rosenberg [3] generalized the embryonic version of DIOGENES in [14] to a methodology that applies to arrays of arbitrary structure by partitioning the fault-tolerant design problem into two tasks:

1. Translate the design problem into the book-embedding problem.
2. Convert the resulting book embeddings to efficient layouts.

The process of embedding a graph \( G \) in a book can be described as follows.

* the vertices of \( G \) lie along the spine of the book;
* each edge of \( G \) lies on a single page;
* no two edges on the same page cross.

The central insight that establishes the equivalence of the book-embedding problem and the stack-layout problem is that edges that are laid out via the same stack do not cross; thus, each page of the book corresponds to a stack in the layout.

One further level of abstraction makes explicit the degrees of freedom one has when translating book embeddings to physical layouts. Splice the ends of a book's spine together: The book-embedding problem becomes the problem of embedding the graph \( G \) in a circle:

* the vertices of \( G \) lie on the circle;
* each edge of \( G \) is a chord of the circle.

One must color the chords of the circle so that like-colored chords do not cross.

There are three relevant measures of the quality of a book/circle embedding:

1. the number of pages employed (= the number of edge-colors used);

---

1 Since the \( G_i \) variables allow one, as in Fig. 1, to bypass faulty PEs with no conceptual difficulty, one needs concentrate only on the issue of configuring the good PEs into the desired structure using stacks.