# IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems

Proceedings

# IEEE International Symposium on

# DEFECT and FAULT TOLERANCE in VLSI SYSTEMS

**25-27 October 2000**

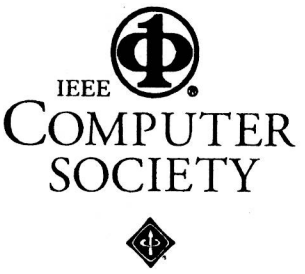
**Yamanashi, Japan**

*Sponsored by*

IEEE Computer Society
IEEE Computer Society Technical Committee on Fault-Tolerant Computing
IEEE Computer Society Test Technology Technical Committee

*In cooperation with*

Technical Group on Fault Tolerant Systems, IEICE, Japan





Los Alamitos, California
Washington Brussels Tokyo

IEEE Computer Society Order Number PR00719
ISBN 0-7695-0719-0
ISBN 0-7695-0720-4 (case)
ISBN 0-7695-0721-2 (microfiche)
ISSN 1063-6722

*Additional copies may be ordered from:*

IEEE Computer Society
Customer Service Center
10662 Los Vaqueros Circle
P.O. Box 3014
Los Alamitos, CA 90720-1314
Tel: + 1 714 821 8380
Fax: + 1 714 821 4641
http://computer.org/
csbooks@computer.org

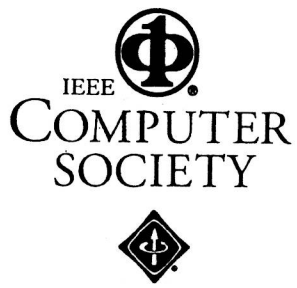
IEEE Service Center
445 Hoes Lane
P.O. Box 1331
Piscataway, NJ 08855-1331
Tel: + 1 732 981 0060
Fax: + 1 732 981 9667
http://shop.ieee.org/store/
customer-service@ieee.org

IEEE Computer Society
Asia/Pacific Office
Watanabe Bldg., 1-4-2
Minami-Aoyama
Minato-ku, Tokyo 107-0062
JAPAN
Tel: + 81 3 3408 3118
Fax: + 81 3 3408 3553
tokyo.ofc@computer.org

Editorial production by Danielle C. Young

Cover art production by Joe Daigle/Studio Productions

Printed in the United States of America by The Printing House



# DFT 2000

# Message from the Symposium Chairpersons

Welcome to the 2000 IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems! The millennium year marks the 15th of a long and productive series of technical meetings since 1986. At the begining, there was the IEEE Design for Yield Workshop, held in Oxford, UK in 1986. Then came the very successful series of the IEEE International Workshop on Defect and Fault Tolerance in VLSI Systems. Finally, in 1996, this meeting has been upgraded to Symposium. The changes are a mere reflection of the rapidly growing dependency of the society on the VLSI systems. In this 15th meeting, we vow to continue the tradition of bringing together academic and industry in the fields of defect and fault tolerance in VLSI systems and continue to be at the forefront of technical advancement.

The technical program consists of 12 single-track sessions with 45 contributed papers covering a wide range of current issues in defect and fault tolerance in VLSI systems. The papers in the 11 regular sessions covers yield analysis, modeling and enhancement, fault-tolerance interconnections and systems, reconfiguration and repair, error coding, online testing, testing and BIST techniques, and fault injection techniques and environment. A special session on "Wafer Scale/Large Area Systems" is organized by G. Chapman to bring forth the latest progress in the VLSI systems to our participants. The authors came from all over the world: USA, Japan, Italy, China, Taiwan, India, New Zealand, Italy, France, Germany, Spain, Austria, Netherlands, Poland, Sweden, Belarus, Estonia and Canada; 18 countries in all. We are particularly pleased to find that many collaborations included in this year's symposium were originated from the discussions at this Symposium previously.

We would like to thank the authors for their excellent contributions, the program committee and the external reviewers for their timely reviews and constructive feedbacks on the papers, and to the keynote speaker.

We hope you will find this Symposium to be technologically informative and stimulating. The exquisite location at the foot of Mt. Fuji will definite help encourage many insightful discussions. Welcome old friends and new. Welcome to Japan.

*General Chairs*

*Eiji Fujiwara*

*Vincenzo Piuri*

*Program Chairs*

*Susumu Horiguchi*

*Jien-Chung Lo*

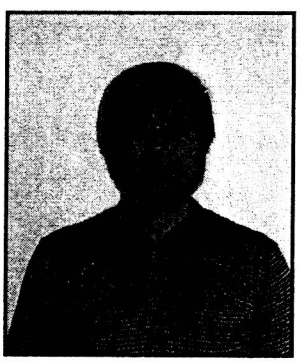
# Organizing Committee

## General Co-Chairs

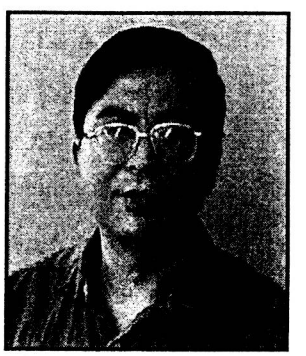
Eiji Fujiwara
*Tokyo Institute of Technology, Japan*

Vincenzo Piuri
*Politecnico di Milano, Italy*

## Program Co-Chairs



Susumu Horiguchi
*JAIST, Japan*



Jien-Chung Lo
*University of Rhode Island, USA*

## Special Session Chair

Glen Chapman, *Simon Fraser University, Canada*

# Table of Contents

## IEEE International Symposium on DEFECT and FAULT TOLERANCE in VLSI SYSTEMS

## Session 3: Wafer Scale/Large Area Systems



## Session 4: Fault-Tolerant Interconnections

## Session 5: Fault-Tolerant Systems



## Session 6: Error Coding



## Session 7: Reconfiguration and Repair

**Session 8: Online Testing**



**Session 9: Built-In Self-Test**



**Session 10: Testing Strategies**

**Session 11: $I_{DDQ}$ Testing**



**Session 12: Fault Injection**

# Session 1

## Yield Analysis and Modeling

# The Effect of Placement on Yield for Standard Cell Designs *

Rajnish K. Prasad and Israel Koren
Department of Electrical and Computer Engineering
University of Massachusetts, Amherst, MA 01003, USA

**Abstract**

*The ability to improve the yield of integrated circuits through layout modification has been recognized and several techniques for yield enhanced routing and compaction have been developed. Yield improvement during routing is however, limited by the predetermined placement. It is conceivable therefore, that different placements of the modules (e.g., standard or custom cells) may lead to very different yield enhanced routings with different projected yields. This is conceptually similar to the effect that the floorplanning of the entire chip has on the yield [2], but while chip floorplanning deals with the major building blocks, placement deals with the modules within an individual block. Yield enhanced placement of modules has not been attempted before mainly due to the difficulty of estimating the yield of the block before the routing is done. Recently, a technique for estimating the yield prior to the routing has been developed [1] making it possible to modify the placement in order to achieve higher yield. The goals of this paper are to investigate the effect that placement has on the projected yield and to modify a standard cell placement algorithm so that yield becomes a design objective.*

## 1: Introduction

The general placement problem is the problem of placing a set of circuit modules within a block such that a certain objective function is minimized. The ultimate goal is to minimize the total chip area occupied by the circuit modules and minimize the length of the interconnections between the modules. To make the placement problem computationally feasible, various simpler to calculate objective functions such as the area of the bounding rectangles, total interconnection wire length, or some other routing area estimates are commonly used. The yield of the circuit is normally not considered during placement.

Recently, it has been shown [2] that floorplanning may considerably affect the yield of the chip. We believe that the placement of modules within a block will have a similar impact. Yield enhancement has so far been attempted only during the detailed routing and compaction steps (e.g., [4, 5, 6, 7]).

* Supported in part by NSF under contract MIP-9710130.

0-7695-0719-0/00 $10.00 © 2000 IEEE

However, significant changes in wiring congestion cannot be performed during these steps as the circuits have already been placed. Since there is a direct relationship between the density of the routing and the yield, it is conceivable that by incorporating the expected yield into the objective function of the placement algorithm, improvements in yield can be achieved. This has not been attempted before, since until now the expected yield was calculated only after the layout (including routing) was completed.

It has recently been demonstrated in [1] that reasonably accurate estimates for the yield can be obtained prior to routing. Thus, we can use such estimates within the placement stage to enhance the yield of the final layout. In section 2 we describe the modified placement algorithm which incorporates the yield as a design objective. In section 3 we present some of our numerical results. Section 4 presents conclusions and future work.

## 2: The Modified Placement Algorithm

The placement problem can be classified according to the different types of design methodologies such as gate array, standard cell and macro/custom cell placement. We focus in this paper on the placement of standard cells with yield as a design objective and we use the standard cell placement algorithm TimberWolf [3] to illustrate our approach. This algorithm employs simulated annealing for minimizing the total wire length. The simulated annealing procedure randomizes the iterative improvement technique and also allows occasional "uphill moves" in an attempt to reduce the probability of getting stuck at a local optimal solution. These uphill moves are controlled probabilistically by the temperature T, and become less and less likely toward the end of the process, as the value of T decreases. TimberWolf allows placements with overlapping modules as intermediate solutions, to achieve fast update of the cost function. After each move to a neighboring solution, the overhead in displacing modules to remove overlap is not incurred. TimberWolf also allows modules to move to a new location without any swapping or width requirement, increasing this way the number of different placements examined. The cost function in TimberWolf consists of total wire length and a measure of the overlap between modules. The cost due to module overlap converges to zero, as the temperature T approaches zero guaranteeing in this way a feasible final placement.

We have modified the TimberWolf placement algorithm to include yield as a design objective. The pseudo-code for the modified simulated annealing placement algorithm is shown below.

```
SimulatedAnnealing(x,T){
/* Given an initial solution x and initial parameter T */
while("stopping criterion" is not satisfied ){
        generate T' < T;
        T = T';
        while("inner loop criterion" is not satisfied){
```

```
                generate a new solution x';
                /* estimate yield of new solution */
                YieldEstimate(x');
                /* c(x') is cost of the new solution */
                /* c(x)  is cost of the current solution */
                if(accept(c(x'),c(x)){
                        x = x';
                }
        }
}
}
```

To incorporate the yield objective into the cost function, we have to estimate the yield of intermediate placements. The routine "YieldEstimate" (shown below), estimates the yield of the new placement using *ybound* [1]. As reported in [1], fairly accurate yield estimates (with differences of 1.0 to 4.0% for short-circuit failures and 0.4 to 4.0% for open-circuit failures) can be obtained by the *ybound* algorithm in a fraction of the time required for actual yield estimation. This algorithm uses an approximation of the average length for the conductors in each wiring channel for estimating the short-circuit yield. If the current intermediate placement has overlapping cells, the overlap is removed temporarily before the yield is estimated. This is essential, since substantial overlap between adjacent modules will cause false net segment overlap, which in turn will result in a large number of track requirements and consequently, wrong yield estimation. Once overlaps are removed, a minimum spanning tree is constructed for each net. Then the left edge algorithm is used to assign tracks to all net segments. Finally, the channel information is decompiled for yield estimation. The pseudo-code for yield estimation is

```
YieldEstimate(original){
      /* original placement has module overlaps */
      /* make a copy of it before modifying it */
      newCopy = copyCurrentState(original);

      /* remove overlap from placement */
      removeOverlap(newCopy);

      /* build a Minimum Spanning Tree for each net */
      /* assign tracks using the Left Edge algorithm */
      globroute();
      /* obtain yield estimate using ybound  */
      yield = ybound();
}
```

The routine "accept" in the SimulatedAnnealing procedure, takes in the new cost c(x') and current cost c(x) and decides if the new solution should be accepted or rejected. The new solution is definitely accepted if the new cost is better than the current one, and is accepted with a probability determined by the annealing schedule if it is worse than the current one.

The new cost function for the modified algorithm is

$$Cost(x) = WireLength(x) + Overlap(x) - Yield(x) * ScaleFactor(x) * \beta \quad (1)$$

where $x$ denotes the index of the current iteration of the simulated annealing process. The parameters $ScaleFactor(x)$ and $\beta$ are explained below. Since the wire length and overlap costs are large integers and the yield is a fraction less than one, we introduced a scaling factor function so that changes in yield are not ignored completely. The scale factor is determined dynamically for each iteration, and is computed as shown below. We first define

$$Scale(x) = |(WireLength(x) - WireLength(x-1))/(Yield(x) - Yield(x-1))| \quad (2)$$

where $x-1$ denotes the index of the previous iteration. We then compute $ScaleFactor(x)$ as

$$ScaleFactor(x) = ScaleFactor(x-1) + Scale(x-1)/h - Scale(x-1-h)/h \quad (3)$$

where $h$ is the depth of history for variation in wire length with respect to yield. This $ScaleFactor(x)$ captures the information about the average variation in wire length with respect to the change in yield in the last few iterations. From the experiments we carried out, we found that a depth of history equal to 3 worked well. The parameter $\beta$ serves to assign a weight to the yield relative to the wire length and can be any real number greater than zero. As will become evident in the next section, the placement algorithm should be run for several values of $\beta$, and then a placement with acceptable wire length and yield should be selected.

## 3: Numerical Results

Ten benchmark circuits were selected from the iscas and lgsynth91 test suites. Table 1 shows the variation in yield for different placements of the ten circuits. It shows the possible range of yield for the different designs, when starting with any possible initial placement. A key observation is that for larger circuits (e.g., C5315 and C6288), the effect of placement on the yield is larger than for smaller circuits (e.g., C432, C499 and C1355). This is mainly due to the fact that a larger number of placements can be generated for a bigger circuit than for smaller circuits.

In practice however, such a choice of starting with any random initial placement (as shown in Table 1) is not available and various techniques to obtain an initial placement are used. Placement algorithms like those based on simulated annealing normally use an especially generated initial solution to obtain a near optimal placement. Table 2 compares the yield achieved by incorporating