Incorporating Yield Enhancement into the Floorplanning Process

Israel Koren, *Fellow*, *IEEE*, and Zahava Koren

Abstract—The traditional goals of the floorplanning process for a new integrated circuit have been minimizing the total chip area and reducing the routing cost, i.e., the total length of the interconnecting wires. Recently, it has been shown that, for certain types of chips, the floorplan can affect the yield of the chip as well. Consequently, it becomes desirable to consider the expected yield, in addition to the cost of routing, when selecting a floorplan. The goal of this paper is to investigate the two seemingly unrelated, and often conflicting, objectives of yield enhancement and routing complexity minimization. We analyze the possible trade-offs between the two and then present a constructive algorithm for incorporating yield enhancement as a secondary objective into the floorplanning process, with the main objective still being the minimization of the overall routing costs.

Index Terms— Floorplanning, memory ICs, microprocessor, redundancy, routing complexity, yield.

1 Introduction

FLOORPLANNING is the partitioning of the entire chip area into smaller rectangles which will be occupied by the various given building blocks. In the case of a microprocessor, these building blocks (also called *modules*) are instruction cache, data cache, instruction decode unit, integer arithmetic unit, and alike. When deciding on a floorplan, the designer takes into account the preferred mutual position of the modules, which is determined by the number of nets that connect them. Two modules which have a large number of common nets are placed as close to one another as possible in order to reduce the total wiring length and obtain a high performance design by reducing the communication delays.

The problem of finding the floorplan with the minimal wiring cost is, in the general case, NP-complete. Various heuristic algorithms for solving it have been proposed and are currently being used [4], [14], [15].

Until recently, VLSI designers rarely considered yield issues when selecting a floorplan for a newly designed chip. This is justified for small chips whose defect distribution can be accurately described by either the Poisson or the Compound Poisson yield models with large area clustering (i.e., the size of the defect clusters is much larger than the size of the chip) [8]. For such chips, selecting a different floorplan will not affect the projected yield.

This is no longer true for the new integrated circuits which have an area of $2cm^2$ and up, consist of components with different device densities, and may have some incorporated redundancy. It has been shown in [10] that if chips with these attributes are hit by medium-size defect clusters, then changes in the floorplan can affect their projected yield. It has been further shown that the optimal floorplan, from the yield point of view, depends on the fault densities of the different modules. Since the fault density of a module is determined by its sensitivity to manufacturing defects, there is no direct relationship between the module's fault density and its connectivity to the other modules. It is, therefore, very likely that the floorplans with the smallest wiring cost will not have the highest possible yield. Clearly, minimizing the total wiring length, which impacts the performance of the chip, will always be more important than increasing the expected yield. There is a need, however, to study the two objectives (total wiring cost and yield) and explore the possibility of optimizing the yield with the minimum wiring cost as a constraint. Also, some trade-off between the two may be considered.

There are two main types of ICs for which the projected yield may be affected by the floorplan. In the first type, the individual modules in the chip have substantially different device densities, leading to very different sensitivities to manufacturing defects and, consequently, to different fault densities. Most current microprocessors fall into this category since they include very dense modules, like data and instruction cache, on the one hand and very sparse modules, like instruction decoder, on the other. In the second type, the IC contains several spare modules to be activated upon the failure (due to manufacturing defects) of a primary module. Memory ICs, especially very large ones, tend to contain redundant modules and, thus, fall into this second category. Both types of ICs are analyzed in this paper.

The paper is organized as follows: In Section 2, we describe the underlying model and suggest approaches to the solution of the dual-objective optimization problem. In Section 3, we study the floorplan of a simple processor consisting of nine modules with different sensitivities to defects. In Section 4, a floorplan of a chip with redundancy is investigated. Similar analysis is then applied, in Section 5, to the floorplan of a 1Gbit memory IC which includes redundant circuits. In Section 6, we present an algorithm for incorporating yield considerations into the process of

- *The authors are with the Department of Electrical and Computer Engineering, University of Massachusetts, Amherst, MA 01003.*
E-mail: koren@euler.ecs.umass.edu.

Manuscript received 15 Apr. 1999; revised 1 Feb. 2000; accepted 15 Feb. 2000.
For information on obtaining reprints of this article, please send e-mail to: tc@computer.org, and reference IEEECS Log Number 111436.

0018-9340/00/$10.00 © 2000 IEEE

generating a floorplan with reduced wiring cost. In Section 7, we illustrate our algorithm using the floorplan of a 64-bit microprocessor. Section 8 concludes the paper.

2 Model Description

Our model deals with a chip consisting of n modules, where module i has dimensions $b_i \times c_i$ and an area of $a_i = b_i \cdot c_i$. A floorplan is a placement of the modules on the chip area, given by $(x_1, y_1), \ldots, (x_n, y_n)$, where (x_i, y_i) denote the coordinates of the center of module i as measured from the lower left corner of the chip. Since, at the floorplanning phase, the complete routing has not yet been attempted, only rough estimates of the wiring length can be taken into account. A commonly used estimator [4] for the wiring length of a given net is the rectilinear (Manhattan) distance, denoted by D_{ij}, between the centers of the two modules i and j connected by that net, i.e.,

$$D_{ij} = |x_i - x_j| + |y_i - y_j|.$$

Thus, the metric for the wiring length between two modules i and j, which have N_{ij} nets in common, is given by $w_{ij} N_{ij} D_{ij}$, where w_{ij} is the weight assigned to N_{ij} according to the criticality of the corresponding nets. A net on a critical path in the chip will be assigned a larger weight to guarantee shorter wires for better performance. In addition to the intermodule connections, we also consider the wires to the chip pins. For module i, we denote their number by $N_{i,out}$, their length by $D_{i,out}$, and the weight attached to them by $w_{i,out}$. Since the exact chip dimensions, as well as the exact position of the I/O pins, are not known at the floorplanning phase, we estimate the chip dimensions by $h \times h$ where

$$h = \sqrt{\sum_{i=1}^{n} a_i}$$

and then estimate $D_{i,out}$ by

$$D_{i,out} = min(x_i, y_i, h - x_i, h - y_i).$$

The objective is to minimize the total wiring cost W given by

$$W = \sum_{i=1}^{n} \sum_{j=1}^{i-1} w_{ij} N_{ij} D_{ij} + \sum_{i=1}^{n} w_{i,out} N_{i,out} D_{i,out}. \tag{1}$$

For the yield analysis of the different floorplans, we first describe the statistical spatial distribution of manufacturing *defects* over the chip area and then derive the distribution of *faults*, which are those defects that interfere with the proper operation of the chip and cause the actual yield losses. It has been known for many years now that manufacturing defects (and, consequently, faults) occur in clusters, rather than being randomly distributed over the chip area [12]. The large area negative binomial model has been successfully used in the past to describe the distribution of both defects and faults [8].

Since we are dealing with very large ICs for which the defect clusters are much smaller than the chip, we use in our analysis a generalization of the negative binomial distribution which allows for several independent defect clusters to occur on one chip. This model is the medium-size clustering negative binomial distribution presented in [5]. Under this yield model, the chip is divided into disjoint areas, called "blocks," representing the areas on the chip in which the defect clusters are enclosed. The defects in each block have a negative binomial distribution and the defects in different blocks are statistically independent. For greater model generality, the exact placement of the blocks on the chip is random, rather than fixed, better describing the actual behavior of defect clusters. The fit of the model to actual defect data has been demonstrated in [6]. The medium-size negative binomial defect distribution has as parameters $\lambda_b^{(D)}$—the average number of defects per block, α—the clustering parameter (which is smaller for more pronounced clustering) and the block size (which is about the size of a defect cluster). All three parameters can be estimated given wafer defect maps [5]. The parameter $\lambda_b^{(D)}$ is sometimes replaced by the parameter $d^{(D)}$—the defect density or the average number of defects per unit area. The two parameters are related through

$$\lambda_b^{(D)} = a_b d^{(D)},$$

where a_b is the block area. For this model, the probability that an area of size a located within a block has exactly k defects is given by

$$\begin{aligned} &Prob\,(area\ of\ size\ a\ has\ k\ defects) \\ &= \frac{\Gamma(\alpha + k)}{k!\,\Gamma(\alpha)} \frac{(ad^{(D)}/\alpha)^k}{(1 + ad^{(D)}/\alpha)^{\alpha+k}}. \end{aligned} \tag{2}$$

To be able to calculate the chip yield, we have to find the distribution of the operational *faults*, rather than that of defects. A defect becomes a fault if it disconnects a line or connects two adjacent lines which belong to different nets. Therefore, the probability that a defect will become a fault depends on the width of the lines and on the spacing between them and, thus, can differ from one module to the other. Denoting by p_i the probability that a defect occurring in module i will become a fault, by λ_i the average number of faults in module i, and by d_i the fault density for module i (i.e., the average number of faults per unit area) we obtain

$$d_i = p_i d^{(D)}$$

and

$$\lambda_i = a_i d_i = p_i \lambda_b^{(D)} \frac{a_i}{a_b} \tag{3}$$

and the probability that module i is fault-free is

$$Prob\,(module\ i\ is\ fault\text{-}free) = [1 + \lambda_i/\alpha]^{-\alpha}. \tag{4}$$

The yield of the chip as a whole depends on its exact structure and whether it has any incorporated redundancy.

The objective is to maximize the chip yield, Y, while still keeping the wiring cost W minimal. Multiobjective optimization problems, in which two or more possibly conflicting criteria appear simultaneously, occur frequently in engineering, economics, management, and manufacturing [2], [3]. There is no clear-cut solution to a multiobjective problem since improving one objective may have a negative impact on the others. The objectives of minimizing W and, at the same time, maximizing Y are usually conflicting objectives and, except for some very special cases, cannot be accomplished simultaneously. The two most frequently used approaches to attacking such a problem are weighting and Pareto optimization.

Definition. *A* solution *to the multiobjective optimization problem is the vector of values of the different objectives resulting from selecting one of the possible alternative actions. In our problem, a solution is a tuple (W, Y) resulting from selecting a specific floorplan.*

The first approach changes the optimization problem from multidimensional to one-dimensional. A weight is attached to each objective according to its importance. The weighted sum of the objectives is calculated for each solution and the solution with the highest weighted sum is then selected. In our problem, weights c_w and c_y are selected, and the optimal solution $S^* = (W^*, Y^*)$ is the one which minimizes the function $Z = c_w W - c_y Y$.

The second approach preserves the multidimensionality of the problem. Rather than selecting one solution, a set of "Pareto-optimal" solutions is found so that none of the solutions in the set "dominates" any of the others and all are considered equally good and equally "optimal."

To construct the Pareto-optimal set of solutions for our problem, we use the following definition

Definition. *Given two solutions, $S_1 = (W_1, Y_1)$ and $S_2 = (W_2, Y_2)$, we say that S_1 dominates S_2, denoted by $S_1 \succ S_2$, if $W_1 \leq W_2$ and $Y_1 \geq Y_2$.*

A set of solutions $\{S_1, \ldots, S_n\}$ is called Pareto-optimal if no solution in the set dominates any other solution, i.e., for any S_i, S_j in the set, $S_i \not\succ S_j$ and $S_j \not\succ S_i$.

In other words: A set of solutions is Pareto-optimal if, for every S_i and S_j in the set (where $S_i = (W_i, Y_i)$ and $S_j = (W_j, Y_j)$), either $W_i < W_j$ and $Y_i < Y_j$ or $W_i > W_j$ and $Y_i > Y_j$, i.e., each solution is better in one aspect but worse in the other.

We next illustrate these two approaches through several examples.

3 Example I—A Simple Processor Chip

In this example, we examine a simple processor chip consisting of nine modules, all of the same size (see Fig. 1). These nine units are a ROM (R), two static memory units (C_1 and C_2) (e.g., instruction cache and data cache), and six random logic units with different transistor densities (L_1, L_2, L_3, L_4, L_5 and L_6).

Denoting the vector $(R, C_1, C_2, L_1, \ldots, L_6)$ by $(1, 2, \ldots, 9)$, a floorplan of this chip can be represented as a permutation $(n_1, \ldots, n_9)$ of $(1, 2, \ldots, 9)$.

As the wiring cost for a given floorplan, we use W as defined in (1),

$$W = \sum_{i=1}^{9} \sum_{j=1}^{i-1} w_{n_i n_j} N_{n_i n_j} D_{ij} + \sum_{i=1}^{9} w_{n_i,out} N_{n_i,out} D_{i,out}.$$

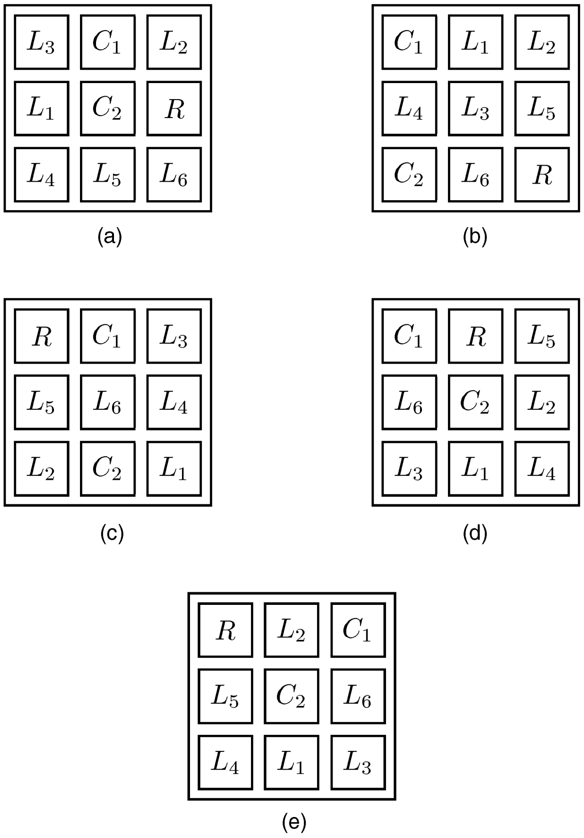


Fig. 1. Five floorplans of a simple processor.

As the interconnection matrix N, indicating the number of wires connecting any two modules, we select

$$N = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 24 & 0 \\ 0 & 0 & 0 & 0 & 0 & 20 & 0 & 0 & 20 \\ 0 & 0 & 0 & 30 & 30 & 0 & 0 & 30 & 30 \\ 0 & 0 & 30 & 0 & 0 & 0 & 15 & 0 & 0 \\ 0 & 0 & 30 & 0 & 0 & 0 & 15 & 0 & 0 \\ 0 & 20 & 0 & 0 & 0 & 0 & 15 & 0 & 0 \\ 0 & 0 & 0 & 15 & 15 & 15 & 0 & 15 & 10 \\ 24 & 0 & 30 & 0 & 0 & 0 & 15 & 0 & 0 \\ 0 & 20 & 30 & 0 & 0 & 0 & 10 & 0 & 0 \end{pmatrix}.$$

For the wires to the chip pins, we use $N_{2,out} = 24$ and $N_{7,out} = 32$. The total number of wires per module, $N_i = \sum_{j=1}^{9} N_{ij} + N_{i,out}$, was chosen so as to satisfy Rent's rule [1], i.e.,

$$N_i = K_p \cdot g^{\beta}, \tag{5}$$

where g is the number of gates in the module and K_p and β are constants whose values depend on the type of the module. For example, for static memories, $K_p = 6$ and $\beta = 0.12$, while, for microprocessor logic units, $K_p = 0.45$ and $\beta = 0.82$ [1].

For the weights w_{ij} we chose two sets of values: The first are uniform, $w_{ij} = 1$ for $i, j = 1, \ldots, 9$. The second are mostly uniform, except for $w_{3,8} = 2$ (i.e., the wires connecting the data cache with the integer arithmetic unit are critical).

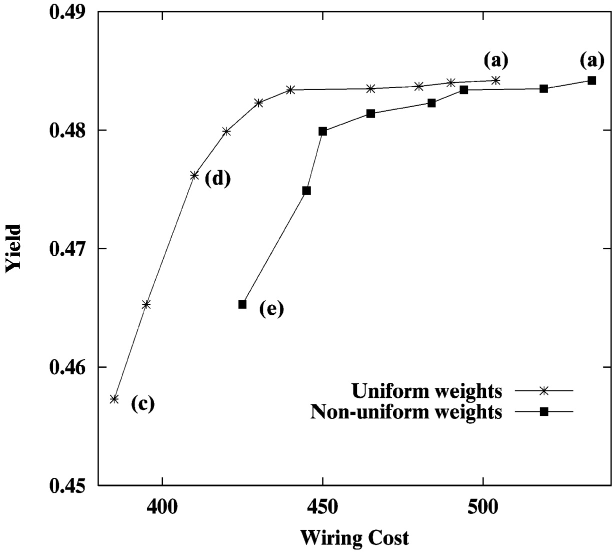


Fig. 2. Optimal combinations of yield and wiring cost.

The yield model used is the medium-size clustering negative binomial model, with a block size of 2 × 2 modules and a clustering parameter $\alpha = 0.3$. Since all modules have the same area, we can assume that the module fault averages, λ_i, are proportional to the module fault densities d_i, which in turn are approximately proportional to the number of transistors per module. Thus,

$$\begin{aligned}&\lambda_R : \lambda_{C_1} : \lambda_{C_2} : \lambda_{L_1} : \lambda_{L_2} : \lambda_{L_3} : \lambda_{L_4} : \lambda_{L_5} : \lambda_{L_6}\\ &= 0.24 : 0.30 : 0.36 : 0.06 : 0.09 : 0.06 : 0.03 : 0.06 : 0.03.\end{aligned} \tag{6}$$

For a given floorplan $(n_1, \ldots, n_9)$, the expected yield is obtained by [5]

$$\begin{aligned}&Y(n_1, \ldots, n_9)\\ &= \frac{1}{4}[y(\lambda_{n_1})y(\lambda_{n_2} + \lambda_{n_3})y(\lambda_{n_4} + \lambda_{n_7})y(\lambda_{n_5} + \lambda_{n_6} + \lambda_{n_8} + \lambda_{n_9})\\ &+ y(\lambda_{n_3})y(\lambda_{n_1} + \lambda_{n_2})y(\lambda_{n_6} + \lambda_{n_9})y(\lambda_{n_4} + \lambda_{n_5} + \lambda_{n_7} + \lambda_{n_8})\\ &+ y(\lambda_{n_7})y(\lambda_{n_1} + \lambda_{n_4})y(\lambda_{n_8} + \lambda_{n_9})y(\lambda_{n_2} + \lambda_{n_3} + \lambda_{n_5} + \lambda_{n_6})\\ &+ y(\lambda_{n_9})y(\lambda_{n_3} + \lambda_{n_6})y(\lambda_{n_7} + \lambda_{n_8})y(\lambda_{n_1} + \lambda_{n_2} + \lambda_{n_4} + \lambda_{n_5})],\end{aligned} \tag{7}$$

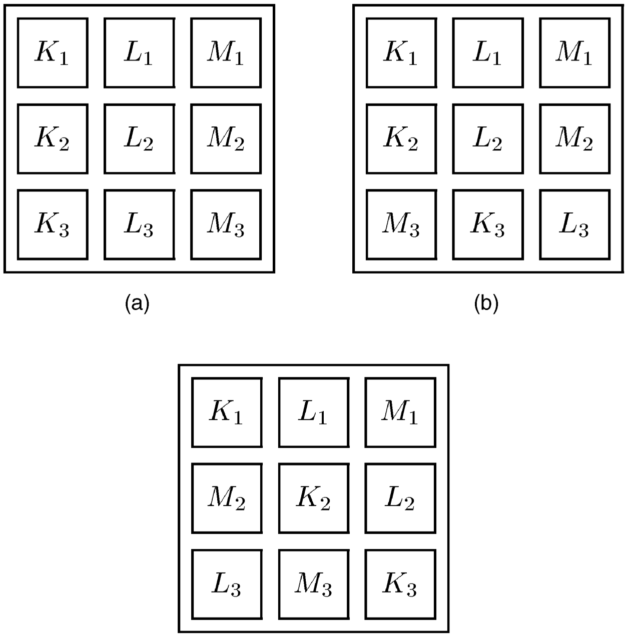
where

$$y(\lambda) = [1 + \lambda/\alpha]^{-\alpha}.$$

3.1 Numerical Results

We calculated the set of all Pareto-optimal floorplans, for both sets of weights, and they are depicted in Fig. 2. Given appropriate weights c_w and c_y, one of these points can be selected so as to minimize $Z = c_w W - c_y Y$. Clearly, if $c_w \geq c_y$, then the first point on the left, the one which minimizes the wiring length, will be chosen. If yield is of utmost importance, then the last point on the right, in which the yield is the highest, will be selected.

The floorplan with the highest overall yield (0.484) is shown in Fig. 1a. The floorplan with the lowest overall yield (0.437) is shown in Fig. 1b. These two floorplans demonstrate the benefits, with respect to yield, of following the rules presented in [10], i.e., for higher yield one should place the modules with the highest sensitivity to defects in the center of the chip and rearrange the other modules so that the lower the fault density, the farther away from the center is the module. The yield of the floorplan in Fig. 1a is larger by 10 percent than that of the floorplan in Fig. 1b.



K_1	L_1	M_1
K_2	L_2	M_2
K_3	L_3	M_3

(a)

K_1	L_1	M_1
K_2	L_2	M_2
M_3	K_3	L_3

(b)

K_1	L_1	M_1
M_2	K_2	L_2
L_3	M_3	K_3

(c)

Fig. 3. Three floorplans of a 3×3 array with redundancy.

Floorplan (c), shown in Fig. 1c, has the lowest wiring cost of 385, but a yield of only 0.457 (roughly half way between the highest and the lowest yields). Floorplan (d), shown in Fig. 1d, is an attempt to increase the yield with only a slight increase in the wiring cost. Its projected yield is 0.476 and its wiring cost is 410. In other words, for a 6.5 percent increase in wiring cost we can achieve a 4.2 percent increase in yield.

The effect of the weights w_{ij} can be seen by comparing floorplans (c) and (e), which have minimal wiring length for the two sets of weights. In (c), C_2 and L_5 are not adjacent, but they were forced to be adjacent in (e) by increasing the weight $w_{3,8}$ from 1 to 2.

4 Example II—A Design With Redundancy

To illustrate the possible trade-offs between the yield and the wiring cost for a chip with redundancy, consider the following example of a chip which consists of nine modules, K_1, K_2, K_3, L_1, L_2, L_3, M_1, M_2, and M_3, where two modules of each type (K_i, L_i, and M_i) are sufficient for the proper operation of the chip. Three topologically distinct floorplans for this chip are depicted in Fig. 3. The difference between these floorplans is the proximity of the modules of the same type to one another. Floorplan (a) is one extreme in the sense that all modules of the same type are adjacent, while floorplan (c) is the opposite extreme since modules of the same type are the farthest apart. Floorplan (b) is situated between these two.

To estimate the expected yield of this chip, we again use the negative binomial distribution with medium-size clusters. Clearly, if the blocks in this yield model are very small or very large relative to the chip size, all three floorplans will have the same yield. Similarly, if the blocks are horizontal, the projected yield is the same for all floorplans. If, on the other hand, the blocks are vertical, the different floorplans will have significantly different yields, as shown below.

Suppose the chip consists of three vertical blocks of size three modules each, with an average of λ_m faults per module and a clustering parameter of α. For each of the three floorplans, the yield of the chip is calculated as the probability that at least two of the K_i, at least two of the L_i, and at least two of the M_i modules are fault-free. Using the negative binomial distribution, the probability that k modules located in the same block will be fault-free, denoted by y_k, is given by

$$y_k = [1 + k\lambda_m/\alpha]^{-\alpha}. \qquad (8)$$

Based on combinatorial considerations (i.e., the inclusion and exclusion formula), the yield of floorplan (a), $Y(a)$, is equal to [8]

$$Y(a) = (3y_2 - 2y_3)^3.$$

By going over all the combinations of faulty and nonfaulty modules such that the chip is operational, we obtain as the yield of floorplan (b), $Y(b)$, the expression

$$\begin{aligned} Y(b) = {} & y_3[y_2(2y_1 - y_2) + 2(y_1 - y_2)(2y_2 - y_3)] \\ & + [y_2 - y_3]\left[y_2^2 + 2(y_1 - y_2)y_3\right] \\ & + 2[y_2 - y_3][y_3(2y_1 - y_2) + 2(y_2 - y_3)(2y_2 - y_3)] \\ & + 2[y_1 - 2y_2 + y_3][y_3y_2 + 2(y_2 - y_3)y_3], \end{aligned}$$

which, after algebraic simplification, becomes

$$Y(b) = 18y_1y_2y_3 - 12y_1y_3^2 - 42y_2^2y_3 + 36y_2y_3^2 + 9y_2^3 - 8y_3^3.$$

Similarly, for the yield of floorplan (c), $Y(c)$, we obtain

$$\begin{aligned} & Y(c) = y_3 \cdot \\ & [y_3 + 3(y_2 - y_3)y_1 + 3(y_1 - 2y_2 + y_3)y_2 + (1 - 3y_1 + 3y_2 - y_3)y_3] \\ & + 3[y_2 - y_3][y_1y_3 + 2(y_2 - y_3)y_2 + (y_1 - 2y_2 + y_3)y_3] \\ & + 3[y_1 - 2y_2 + y_3][y_2y_3 + (y_2 - y_3)y_3] \\ & + [1 - 3y_1 + 3y_2 - y_3]y_3^2 \end{aligned}$$

which can be simplified to

$$\begin{aligned} Y(c) = {} & 18y_1y_2y_3 - 18y_1y_3^2 - 36y_2^2y_3 + 36y_2y_3^2 + 6y_2^3 + 3y_3^2 \\ & - 8y_3^3. \end{aligned}$$

It can be easily shown that for any values of λ_m and α, $Y(a) \leq Y(b) \leq Y(c)$. To demonstrate this point, we calculated the yield of the three floorplans, using the negative binomial model with $\alpha = 0.5$, several values of the module fault average λ_m, and vertical fault clusters. The results are depicted in Fig. 4 and the advantage of floorplan (c) is evident.

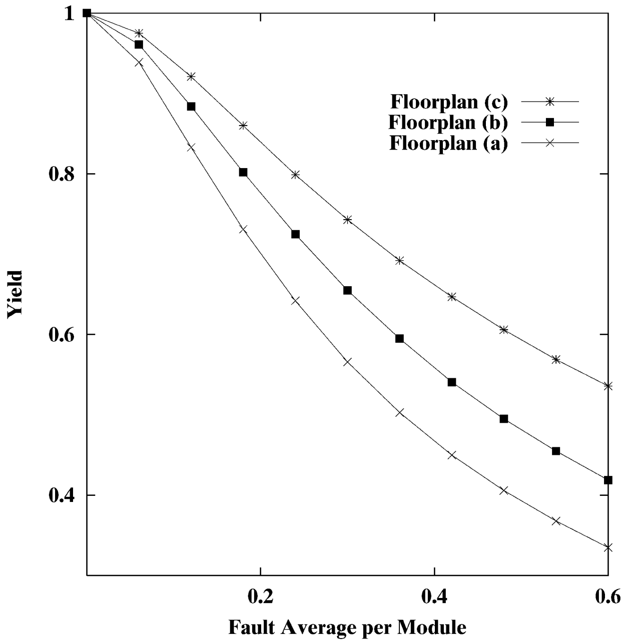


Fig. 4. The yield of the three floorplans.

An intuitive justification for floorplan (c) having the highest yield is that it guarantees the separation between modules of the same type for almost any fault cluster, making it less likely that the same cluster will hit two of them. In floorplan (a), on the other hand, it is more likely that a fault cluster will hit two modules of the same type thus rendering the chip faulty. Therefore, if maximizing the yield is the main consideration, floorplan (c) should be preferred over (a) and (b).

However, the three floorplans have different wiring lengths and this criterion should be taken into account when selecting a floorplan.

Suppose the interconnect matrix N is given by

$$N^{(1)} = \begin{pmatrix} 0 & 16 & 16 & 32 & 16 & 16 & 32 & 16 & 16 \\ 16 & 0 & 16 & 16 & 32 & 16 & 16 & 32 & 16 \\ 16 & 16 & 0 & 16 & 16 & 32 & 16 & 16 & 32 \\ 32 & 16 & 16 & 0 & 16 & 16 & 32 & 16 & 16 \\ 16 & 32 & 16 & 16 & 0 & 16 & 16 & 32 & 16 \\ 16 & 16 & 32 & 16 & 16 & 0 & 16 & 16 & 32 \\ 32 & 16 & 16 & 32 & 16 & 16 & 0 & 16 & 16 \\ 16 & 32 & 16 & 16 & 32 & 16 & 16 & 0 & 16 \\ 16 & 16 & 32 & 16 & 16 & 32 & 16 & 16 & 0 \end{pmatrix}$$

and $w_{ij} = 1$ for $i, j = 1, \ldots, 9$, then

$$W(a) = 1,344, \qquad W(b) = 1,472, \qquad W(c) = 1,536.$$

Floorplan (a) has the lowest wiring cost, (b) is second, and (c) is the worst with regard to this criterion. Fig. 5 depicts the three combinations of the yield (for $\lambda_m = 0.3$ and $\alpha = 0.5$) and wiring cost and there is clearly a conflict between the two. One of the points can be selected according to the relative importance of the two criteria.

If, on the other hand, the matrix N is given by

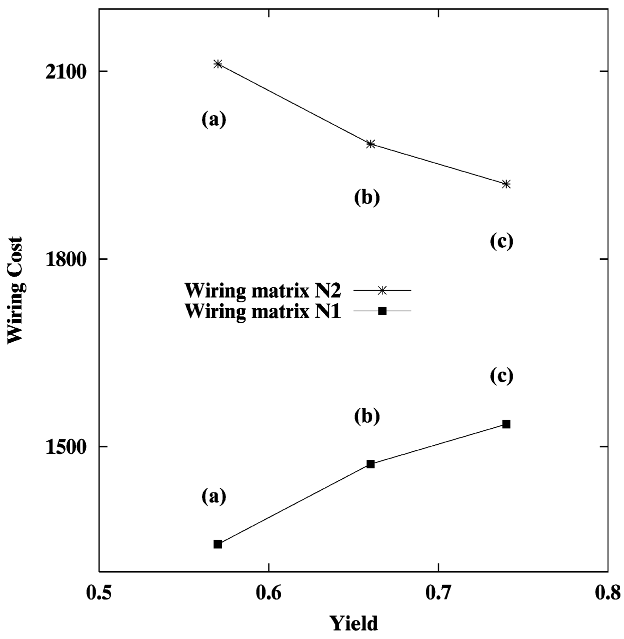


Fig. 5. Wiring cost vs. yield for the three floorplans.

$$N^{(2)} = \begin{pmatrix} 0 & 32 & 32 & 16 & 32 & 32 & 16 & 32 & 32 \\ 32 & 0 & 32 & 32 & 16 & 32 & 32 & 16 & 32 \\ 32 & 32 & 0 & 32 & 32 & 16 & 32 & 32 & 16 \\ 16 & 32 & 32 & 0 & 32 & 32 & 16 & 32 & 32 \\ 32 & 16 & 32 & 32 & 0 & 32 & 32 & 16 & 32 \\ 32 & 32 & 16 & 32 & 32 & 0 & 32 & 32 & 16 \\ 16 & 32 & 32 & 16 & 32 & 32 & 0 & 32 & 32 \\ 32 & 16 & 32 & 32 & 16 & 32 & 32 & 0 & 32 \\ 32 & 32 & 16 & 32 & 32 & 16 & 32 & 32 & 0 \end{pmatrix}.$$

Then,

$$W(a) = 2,112, \qquad W(b) = 1,984, \qquad W(c) = 1,920.$$

As can be seen in Fig. 5, both criteria agree and floorplan (c) is the one to be selected.

5 Example III—A Memory IC

Several new approaches for incorporating defect-tolerance into memory ICs have been proposed and implemented [13], [16], [17]. In [17], a hybrid design which combines the traditional row and column redundancy with several redundant subarrays is presented. The main reason for this hybrid design is that the higher density of the new submicron memory ICs drastically increases the yield loss due to chip-kill defects, e.g., defects in core circuits like sense amplifiers and line drivers [13], [16], [17]. The conventional technique using spare rows and columns is incapable of dealing with such defects and the entire subarray must be replaced. The purpose of the redundant subarrays in the Samsung memory IC [17] is to replace those primary memory subarrays hit by chip-kill faults.

The designed chip is a 1 Gbit memory which includes eight mats of size 128 Mbit each and eight redundant blocks of size 1 Mbit each (see Fig. 6). Each redundant block consists of four basic 256 Kbit arrays and has additional eight spare rows and four spare columns (see Fig. 7). The purpose of the spare rows and columns is to increase the probability that the redundant block is operational and can be used for replacing a block with chip-kill faults.

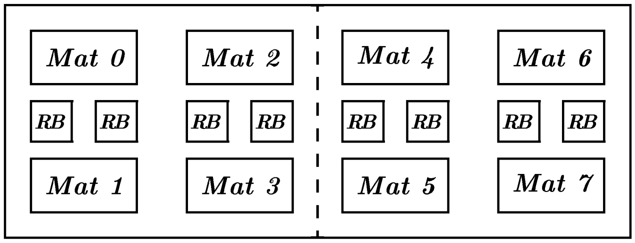


Fig. 6. A 1 Gbit chip with eight mats of size 128Mbit each and eight redundant blocks (***RB***) of size 1Mbit each.

Each mat consists of 512 basic arrays of size 256 Kbit and has 32 spare rows and 32 spare columns. However, these spare rows and columns cannot be used to replace every defective row or column in the entire mat. Four spare rows are allocated to a 16 Mbit portion of the mat and eight spare columns are allocated to a 32 Mbit portion of the mat.

The analysis in this paper concentrates on chip-kill faults and, as such, deals only with the placement of the redundant blocks. In the current design, the chip is divided into two halves, each with four mats and four redundant blocks (RBs), where an RB can replace any faulty block in the same half-chip. In an alternative floorplan, all four RBs in one half of the chip serve as spares for the mats in the second half. This is expected to increase the yield since the mat and its spares will not be hit by the same cluster of defects, but the wiring length for this floorplan is clearly higher. Another possible solution which will increase both the yield and the wiring length by a smaller amount is to have two out of the four RBs as spares for one half-chip and the other two as spares for the other half. Other floorplans which can be considered are for a design with 16 RBs. In this case, the alternatives are to identify 0, 2, 4, or 8 as spares for the same half in which they are located and the rest as spares for the mats in the second half.

To compare the different floorplans, we used as yield model the negative binomial model, with a fault density of d faults per 1 Mbit, a clustering parameter α, and fault clusters with approximately the size of one eighth of a chip (128 Mbits). We compared the yield of the three different floorplans with eight RBs and the five different floorplans with 16 RBs. The percentages of yield improvement, relative to the yield of the original design, are depicted in Fig. 8. As this figure shows, the more spares are separated from the modules that they are replacing, the higher the yield. To determine whether the increase in yield is worth the increase in wiring cost, we estimated the length of the wires connecting the redundant blocks to the mats for which they act as spares. Fig. 9 shows the percentage of

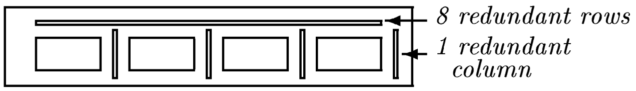


Fig. 7. A redundant block including four 256 Kbit arrays, eight redundant rows, and four redundant columns.

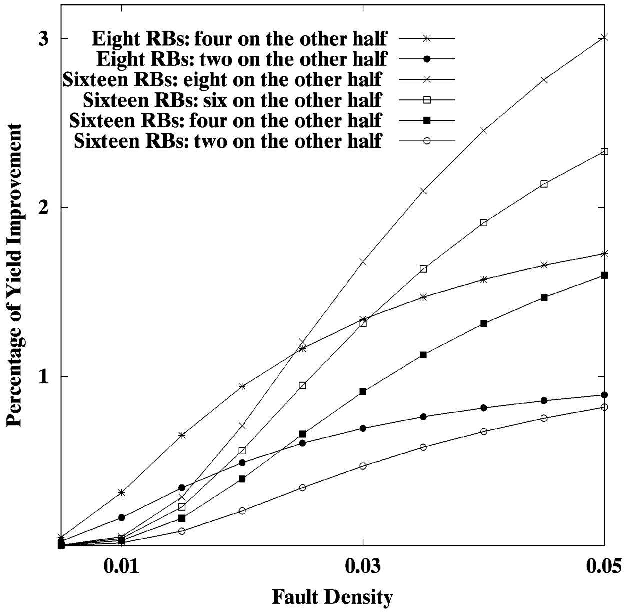


Fig. 8. Percentage of yield improvement.

yield increase versus the additional wiring cost for the redundant blocks (which is only a small percentage of the total wiring cost for the entire chip). This figure shows by how much the yield can increase if the designer is willing to allow the wiring cost to increase slightly. Note that the small increase in wiring length is expected to have a negligible effect on the overall yield, which is mainly determined by the sensitivity to defects of the high-density memory arrays (the mats and redundant blocks). Also notice that the expected yield improvement is quite limited and, as a result, the above proposed modifications in the placement of the redundant blocks are worthwhile only for high manufacturing volume memory ICs.

6 A Yield-Oriented Placement Algorithm

We next present an algorithm which incorporates yield enhancement into the floorplanning process. Our algorithm is based on a constructive floorplanning algorithm [15], adding yield enhancement as a secondary consideration in addition to wiring minimization. The algorithm places one module at a time and this is done in three steps:

1. Selecting the next module to be placed out of the yet unplaced modules according to its area and its connections to the already placed modules.
2. Placing the selected module in the floorplan with the aim of minimizing the wiring length to the already placed modules.
3. If more than one placement is found in Step 2, then the one maximizing the yield is selected.

Steps 1 and 2 are based on [15], but Step 3 is a new contribution of this paper and will be explained next.

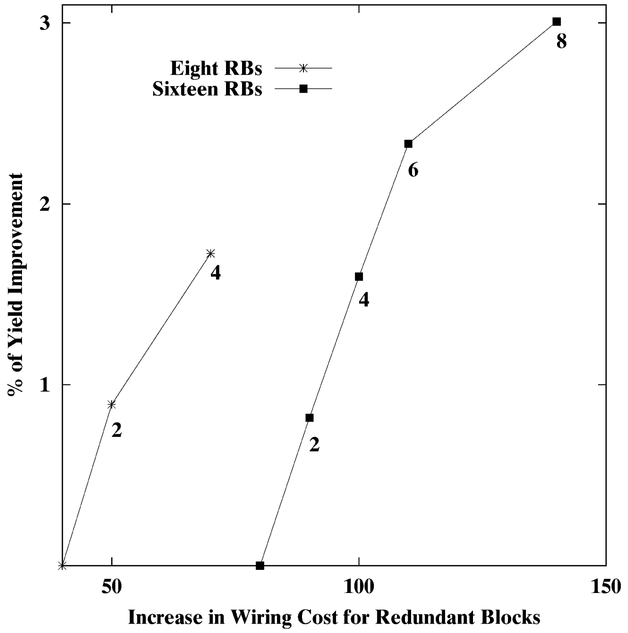


Fig. 9. Yield improvement vs. additional wiring cost for the different floorplans (the number of RBs serving as spares for the other half of the chip is marked).

As shown in [10] and exemplified in Section 2 of this paper, the yield of a chip is enhanced when the modules are placed at a distance from the center which is inversely related to their fault densities d_i. To incorporate this into the placement process, let us assume that the chip is a square of size $h \times h$ and denote by (x, y) the coordinates of the center of an about to be placed module, as measured from the lower left corner of the chip. The distance of (x, y) from the center of the chip is given by

$$\sqrt{(x-h/2)^2+(y-h/2)^2}=\sqrt{x(x-h)+y(y-h)+h^2/2}. \qquad (9)$$

To determine whether the module is placed at a yield favorable location, we define a "yield fitness" function

$$F(x,y) = x(h-x) + y(h-y),$$

which decreases as the distance in (9) increases.

To maximize the yield, the modules should be placed so that the higher their fault density d_i, the higher the value of $F(x, y)$. We divide the chip into $l \times l$ equal-area squares, for some odd number l (this is done for a square chip, while a division into $l_1 \times l_2$ would be more suitable for a rectangular chip), and calculate the value of the function F for each of their centers. Due to the symmetry of F around the center of the chip, each four symmetric squares have the same value of F. This results in $M = 1 + (l^2 - 1)/4$ different values, which can be denoted by $F_1 < F_2 < \cdots < F_M$. F_1 is the value of F in the four corner squares, while F_M is its value in the center square. The set of modules $\{1, \ldots, n\}$ is now arranged according to an increasing order of the fault densities d_i and divided, in ascending order, into M groups $G_1, \ldots, G_M$ so that

$$\sum_{i \in G_j} a_i \approx 4h^2/l^2 \;\; for \;\; j = 1, \ldots, M-1$$

and

$$\sum_{i \in G_M} a_i \approx h^2/l^2,$$

where a_i is the area of module i.

In case of a tie in Step 2 (two or more placements have wiring lengths which differ by no more than some predetermined ϵ), then, if the module to be placed is in group G_j, it is placed at the location (x, y), which minimizes $|F(x, y) - F_j|$.

The number l is selected so that h^2/l^2 is roughly the average area of a module. Hence, the larger the number of modules in the chip, the higher will l be. The selection of ϵ, the difference in wiring costs which determines a tie in Step 2, is determined by the relative importance of the yield vs. wiring cost objectives. ϵ will be higher as more emphasis is placed on yield enhancement at the expense of increasing the wiring length.

Denote:
$S = \{1, \dots, n\}$—the set of all modules.
$S^{(P)}$—the set of already placed modules, $S^{(U)}$—the set of yet unplaced modules.

The yield-oriented floorplanning algorithm is:

1. Select values for the parameters l, ϵ.
2. Arrange the modules $1, \dots, n$ so that $d_1 \leq \dots \leq d_n$.
3. Calculate $F_1, \dots, F_M$. Divide $S = \{1, \dots, n\}$ into groups $G_1, \dots, G_M$.
4. Set $S^{(P)} = \Phi$; $S^{(U)} = S$.
5. Select a module $k \in S^{(U)}$ such that

   $$a_k \left(\sum_{i \in S^{(P)}} N_{ki} + N_{k,out} \right)$$

   is maximized.
6. Select for module k a location (x, y) on the chip such that $W^{(x,y)}$ is minimized, where $W^{(x,y)}$

   $$= \sum_{i \in S^{(P)}} w_{ki} N_{ki} D(x, y, x_i, y_i) + w_{k,out} N_{k,out} D(x, y, out)$$

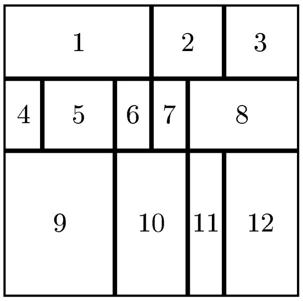
   and $D(x, y, x_i, y_i) = |x - x_i| + |y - y_i|$;

   $$D(x, y, out) = min(x, h - x, y, h - y).$$

   If more than one location (x, y) exists such that the values of $W^{(x,y)}$ differ by less than ϵ, go to 7, otherwise go to 8.
7. If $k \in G_j$, calculate $|F(x, y) - F_j|$ for every minimal location (x, y) found in 6. Select the location for which the difference is minimized.
8. Place k in selected location (x, y) and set $(x_k, y_k) = (x, y)$ $S^{(P)} = S^{(P)} + \{k\}$ $S^{(U)} = S^{(U)} - \{k\}$.
9. If $S^{(U)} \neq \Phi$, go to 5.

7 Example—Floorplanning of a Microprocessor

We used the algorithm described in Section 6 to construct the floorplan of Matsushita's ADENART microprocessor [11]. This microprocessor has a 64-bit RISC superscalar architecture containing a data cache (DCU) and an instruction cache (ICU). It has been implemented in a 0.8μm CMOS technology and it contains 1,300K transistors in a total area of 14.7 × 15.3 mm^2. A simplified diagram of the chip's floorplan is depicted in Fig. 10. The microprocessor includes two register files (FR—floating-point registers and PR—pointer registers), an instruction decode unit (IDU), a data bus control unit (DBC), a ROM, and five execution units: a floating-point add and subtract unit (FAU), a floating-point multiply and divide unit (FMU), a load address add unit (LDU), a pointer arithmetic and logic unit (PNU), and a flow control unit (FCU). The 12 modules have six different transistor densities with the ROM having the highest density and the FCU and IDU, the lowest density. Assuming that the fault densities are linearly proportional to the transistor densities we defined six fault densities which satisfy



Unit	Name	d
1	FCU	d_{lg_1}
2	IDU	d_{lg_1}
3	ICU	d_{cache}
4	PR	d_{lg_4}
5	PNU	d_{lg_2}
6	LDU	d_{lg_2}
7	ROM	d_{rom}
8	FMU	d_{lg_3}
9	DCU	d_{cache}
10	DBC	d_{lg_5}
11	FR	d_{lg_4}
12	FAU	d_{lg_3}

Fig. 10. The original floorplan for the ADENART chip.

$$d_{rom} > d_{cache} > d_{lg\_5} > d_{lg\_4} > d_{lg\_3} > d_{lg\_2} > d_{lg\_1}.$$

These fault densities were assigned to the individual modules, as shown in Fig. 10. Based on the transistor densities reported in [11], the approximate fault densities satisfy

$$\begin{aligned} & d_{rom} : d_{cache} : d_{lg\_5} : d_{lg\_4} : d_{lg\_3} : d_{lg\_2} : d_{lg\_1} \\ & = 8.88 : 7.69 : 3.27 : 2.42 : 2.27 : 1.69 : 1. \end{aligned}$$

As an interconnect matrix, we used the matrix

$$N = \begin{pmatrix} 0 & 0 & 32 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 64 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 32 & 64 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 96 & 96 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 96 & 0 & 0 & 0 & 0 & 32 & 0 & 0 & 0 \\ 0 & 0 & 0 & 96 & 0 & 0 & 0 & 0 & 32 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 32 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 32 & 0 & 0 & 0 & 192 & 0 \\ 0 & 0 & 0 & 0 & 32 & 32 & 0 & 0 & 0 & 160 & 192 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 160 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 192 & 192 & 0 & 0 & 192 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 192 & 0 \end{pmatrix}.$$

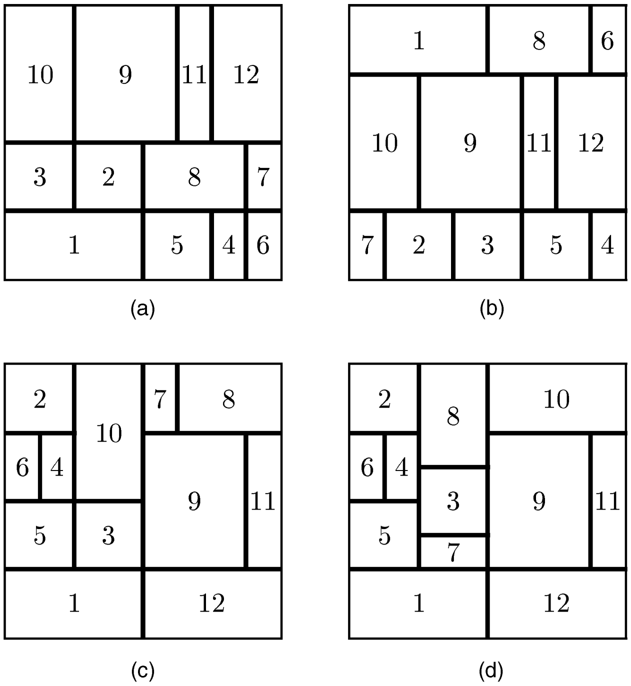


Fig. 11. Four alternative floorplans for the ADENART chip.

As the number of wires to the chip pins, we used $N_{3,out} = 64$ and $N_{10,out} = 160$ and assumed $w_{ij} = 1$ for every i, j. A floorplan of this chip can be represented by $(n_1, \ldots, n_{12})$, where $n_1, \ldots, n_{12}$ is a permutation of $1, \ldots, 12$, although not all permutations can serve as floorplans due to the different sizes of the modules. As the yield model, we used the medium-size clustering negative binomial distribution, with $\lambda_i = a_i d_i$, $\alpha = 0.3$, and a block size of 6×6 (where the chip is assumed to be of size $h \times h = 8 \times 8$).

We applied the algorithm described in Section 6 with $l = 3$, i.e., the chip area is divided into nine squares, and the modules are divided into three groups, according to whether they should be placed in the center, on the sides, or in the corners. We selected four values of ϵ: 0, 100, 200, and 300, resulting in four floorplans in which the yield increases at the expense of increasing wiring length. The four floorplans are depicted in Fig. 11 and their corresponding yields and wiring lengths are shown in Fig. 12. The characteristics of the original floorplan shown in Fig. 12 are for reference only since we had to use approximate values for the entries of the interconnection matrix N.

Fig. 11 and Fig. 12 illustrate the usefulness of the algorithm presented in Section 6, allowing the chip designer to consider the trade-off between wiring cost and yield, and generate the most desirable floorplan according to the relative significance of these two objectives. Notice, however, that we do not claim that the generated floorplans are optimal in any sense. This example is meant to demonstrate the incorporation of yield considerations into a given floorplanning algorithm and, for this purpose, we selected a greedy constructive algorithm. In principle, other floorplanning algorithms can be modified in a similar manner to include yield as a secondary objective.

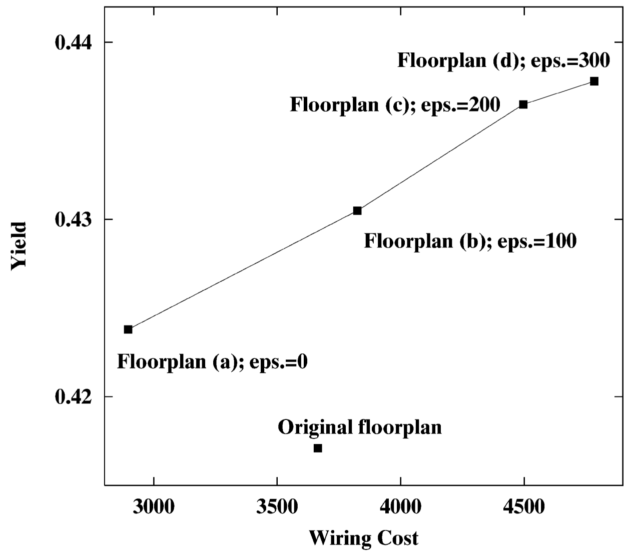


Fig. 12. The yield and wiring cost of the generated floorplans.

8 Conclusion

We have studied in this paper two distinct objectives in floorplanning, namely, total wiring length minimization and yield maximization. We focused on two types of chips for which the floorplan has a great impact on the yield. The first type includes chips consisting of modules with very different defect sensitivities and the second type includes chips which contain redundant modules. Most current microprocessors are of the first type, while several very large memory ICs designed recently belong to the second type.

We have demonstrated that, for these two types of ICs, the yield and wiring cost objectives are often conflicting. We therefore introduced a set of Pareto-optimal solutions, allowing consideration of the trade-off between the two objectives and the selection of one of the solutions according to the relative importance of each of the objectives.

Finally, we introduced, and exemplified through a 64-bit microprocessor, a variation on a constructive floorplanning algorithm which incorporates the yield enhancement goal into the floorplanning process. This variation calculates a specially designed "yield fitness" function and, by varying a parameter ϵ, generates a series of floorplans in which the yield increases at the cost of increasing wiring length, allowing the designer to decide on the most desirable one. Although this variation has been implemented for a greedy constructive algorithm, it can be incorporated into any other constructive floorplanning algorithm.

Acknowledgments

This work was supported in part by the Jet Propulsion Laboratory under contract 961294 and by the U.S. National Science Foundation under contract MIP-9710130. Parts of this work were presented at the 1998 IEEE Symposium on Defect and Fault Tolerance in VLSI Systems, Austin, Texas [9] and the 1999 Symposium on Microelectronic Manufacturing Technologies—Yield, Reliability, and Failure Analysis, Edinburgh, United Kingdom [7].

REFERENCES

[1] H.B. Bakoglu, *Circuits, Interconnections, and Packaging for VLSI* Reading, Mass.: Addison-Wesley, 1990.

[2] V. Chankong and Y.Y. Haimes, *Multiobjective Decision Making: Theory and Methodology.* New York: North-Holland, 1983.

[3] G. Evans, "Overview of Techniques for Solving Multiobjective Mathematical Programs," *Management Science,* vol. 30, no. 11, pp. 1,268-1,282, Nov. 1984.

[4] T. Lengauer, *Combinatorial Algorithms for Integrated Circuit Layout.* West Essex, U.K.: John Wiley & Sons, 1990.

[5] I. Koren, Z. Koren, and C.H. Stapper, "A Unified Negative Binomial Distribution for Yield Analysis of Defect Tolerant Circuits," *IEEE Trans. Computers,* vol. 42, no. 6, pp. 724-737, June 1993.

[6] I. Koren, Z. Koren, and C.H. Stapper, "A Statistical Study of Defect Maps of Large Area VLSI ICs," *IEEE Trans. VLSI Systems,* vol. 2, pp. 249-256, June 1994.

[7] I. Koren and Z. Koren, "Floorplanning of Memory ICs: Routing Complexity vs. Yield," *Proc. Int'l Symp. Microelectronic Manufacturing Technologies—Yield, Reliability, and Failure Analysis (MMT04),* May 1999.

[8] I. Koren and Z. Koren, "Defect Tolerant VLSI Circuits: Techniques and Yield Analysis," *Proc. IEEE,* vol. 86, pp. 1,817-1,836, Sept. 1998.

[9] I. Koren and Z. Koren, "Yield and Routing Objectives in Floorplanning," *Proc. 1998 IEEE Int'l Symp. Defect and Fault Tolerance in VLSI Systems,* pp. 28-36, Nov. 1998.

[10] Z. Koren and I. Koren, "On the Effect of Floorplanning on the Yield of Large Area Integrated Circuits," *IEEE Trans. VLSI Systems,* vol. 5, pp. 3-14, Mar. 1997.

[11] H. Nakano, M. Nakajima, Y. Nakakura, and T. Yoshida, "An 80-MFLOPs (Peak) 64-b Microprocessor for Parallel Computer," *IEEE J. Solid-State Circuits,* vol. 27, pp. 365-371, Mar. 1992.

[12] C.H. Stapper, "Defect Density Distribution for LSI Yield Calculations," *IEEE Trans. Electron Devices,* vol. 20, pp. 655-657, July 1973.

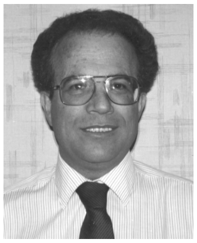
[13] T. Sugibayashi et al., "A 1-Gb DRAM for File Applications," *IEEE J. Solid-State Circuits,* vol. 30, pp. 1,277-1,280, Nov. 1995.

[14] S. Wimer and I. Koren, "Analysis of Strategies for Constructive General Block Placement," *IEEE Trans. Computer-Aided Design,* vol. 7, pp. 371-377, Mar. 1988.

[15] S. Wimer, I. Koren, and I. Cederbaum, "Optimal Aspect Ratios of Building Blocks in VLSI," *IEEE Trans. Computer-Aided Design,* vol. 8, pp. 139-145, Feb. 1989.

[16] T. Yamagata et al., "A Distributed Globally Replaceable Redundancy Scheme for Sub-Half-Micron ULSI Memories and Beyond," *IEEE J. Solid-State Circuits,* vol. 31, pp. 195-201, Feb. 1996.

[17] J.-H. Yoo et al., "A 32-Bank 1Gb Self-Strobing Synchronous DRAM with 1GB/s Bandwidth," *IEEE J. Solid-State Circuits,* vol. 31, pp. 1,635-1,643, Nov. 1996.



Israel Koren (S'72-M'76-SM'87-F'91) received the BSc, MSc, and DSc degrees from the Technion-Israel Institute of Technology, Haifa, in 1967, 1970, and 1975, respectively, all in electrical engineering. He is currently a professor of electrical and computer engineering at the University of Massachusetts, Amherst. Previously, he was with the Technion-Israel Institute of Technology. He also held visiting positions with the University of California at Berkeley, University of Southern California, Los Angeles and University of California, Santa Barbara. He has been a consultant to several companies, including IBM, Intel, Analog Devices, AMD, Digital Equipment Corp., National Semiconductor, and Tolerant Systems. Dr. Koren's current research interests are fault tolerance techniques, models for yield and performance, and computer arithmetic. He published extensively in several IEEE transactions and has more than 130 publications in refereed journals and conferences. He has been a co-guest editor for the *IEEE Transactions on Computers* special issue on high yield VLSI systems, April 1989, and the special issue on computer arithmetic, July 2000. He served on the editorial board of the *IEEE Transactions on Computers* between 1992 and 1997. He also served as program chair and general chair for several conferences and as a program committee member for numerous conferences. He has edited and coauthored the book *Defect and Fault-Tolerance in VLSI Systems*, volume 1 (Plenum, 1989). He is the author of the textbook *Computer Arithmetic Algorithms* (Brookside Court Publishers, 1999).

Zahava Koren received the BA and MA degrees in mathematics and statistics from the Hebrew University in Jerusalem in 1967 and 1969, respectively, and the DSc degree in operations research from the Technion-Israel Institute of Technology in 1976. She is currently a senior research fellow in the Department of Electrical and Computer Engineering, University of Massachusetts at Amherst. Previously, she has held positions with the Department of Industrial Engineering at the University of Massachusetts, Amherst, the Department of Statistics, University of Haifa, Departments of Industrial Engineering and Computer Science at the Technion, and the Department of Business and Economics, California State University at Los Angeles. Her main interests are stochastic analysis of computer networks, yield of integrated circuits, and reliability of computer systems.