The Hyeti Defect Tolerant Microprocessor: A Practical Experiment and its Cost-Effectiveness Analysis

R. Leveugle, *Member, IEEE*, Z. Koren, I. Koren, *Fellow, IEEE*, G. Saucier, *Fellow, IEEE*, and N. Wehn

***Abstract*— This paper summarizes a practical experiment in designing a defect tolerant microprocessor and presents the underlying principles. Unlike memory integrated circuits, microprocessors have an irregular structure which complicates both the task of incorporating redundancy for defect tolerance in the design and the task of analyzing the resulting yield increase. The main goal of this paper is to present the detailed yield analysis of a defect tolerant microprocessor with an irregular structure which has been successfully fabricated.**

The approaches employed for achieving the goal of yield enhancement in the data path and the control part of the microprocessor are described first. Then, the yield enhancement due to the incorporated redundancy is analyzed. Finally, some practical and theoretical conclusions are drawn.

***Index Terms*—Application-specific IC, defect tolerance, equivalent yield, microprocessor, redundancy, yield enhancement.**

I. Introduction

THE Hyeti (High yield and error tolerant integration) microprocessor is a 16-bit defect tolerant microprocessor that was designed and fabricated as part of the European ESPRIT project. The goal of the Hyeti project was to demonstrate the feasibility of a high yield defect tolerant microprocessor. This microprocessor may be used as the core of an application-specific microprocessor-based system that is integrated on a single chip. The large silicon area consumed by such a system will most certainly result in a low yield, unless some defect tolerance in the form of redundancy is incorporated in the design.

Defect tolerance techniques have been applied in the past to several regular designs, most notably memory IC's. The high regularity of memory arrays greatly simplifies the task of incorporating redundancy into their design. A variety of defect tolerance techniques have been exploited in memory designs, from the simple technique using spare rows and columns through the use of error correcting codes [5]. These techniques have been successfully employed by many semiconductors manufacturers, resulting in significant yield improvements ranging from 30-fold increases in the yield of early prototypes to 1.5-to-3-fold yield increases in mature processes.

In contrast, only a few logic IC's have been designed with some built-in defect tolerance (e.g., [8]). Some regularity in the design is necessary if a low overhead for redundancy inclusion is desired. (For completely irregular designs, duplication or even triplication with 100% to 200% overhead are currently the only available redundancy techniques.) Consequently, defect tolerant designs of programmable logic arrays (PLAs) [17] and processor arrays [16] have been proposed and implemented. These circuits, having a highly regular structure, should in principle benefit from the same defect tolerance schemes that have been employed in memory IC's. These schemes, however, have not been widely used for two main reasons. First, the cost-effectiveness of the simple spare rows/columns scheme is very low. More importantly, unlike memory IC's, where all defective cells can be identified by applying external test patterns, the identification of defective elements in logic IC's (even for those with regular structure) is more complex and usually requires the addition of some built-in testing aids. Thus, testability must also be a factor in choosing defect tolerant designs for logic IC's.

The situation becomes even more complex in random logic circuits like microprocessors. When designing such circuits, it is necessary to partition the design into separate components, preferably with each having a regular structure. Then, different redundancy schemes can be applied to the different components, including the possibility of no defect tolerance in components for which the cost of incorporating redundancy becomes prohibitive. This is exactly the strategy that has been adopted in the design of the Hyeti microprocessor.

The Hyeti microprocessor includes a control part and a data path. The control part is constructed of PLA's with built-in defect tolerance. The data path was designed following the bit-slice style which provides the needed regularity. Still, some units in the microprocessor have an irregular structure, and no defect tolerance has been incorporated into them.

The detailed design of the Hyeti microprocessor was presented in [13] and [14]. Its general architecture is briefly reviewed in Section II for completeness. Section III introduces the testing and reconfiguration strategies that are employed once a defect is detected. Sections IV, V, and

Manuscript received January 25, 1993. I. Koren was supported in part by NSF under contract MIP-9305912.

R. Leveugle and G. Saucier are with the Institut National Polytechnique de Grenoble/CSI, 38031 Grenoble, France.

Z. Koren is with the Department of Industrial Engineering and Operations Research, University of Massachusetts, Amherst, MA 01003 USA.

I. Koren is with the Department of Electrical and Computer Engineering, University of Massachusetts, Amherst, MA 01003 USA; e-mail: koren@euler.ecs.umass.edu.

N. Wehn is with the Institute of Microelectronic Systems, Darmstadt University of Technology, D64283 Darmstadt, F.R.G. and Siemens AG, Corporate Research and Development, D81739 Munich, F.R.G.

IEEE Log Number 9404359.

VI include the main contributions of this manuscript. Section IV presents a detailed analysis of the yield enhancement in the data path and the control part. In Sections V and VI we discuss the analysis results and draw some conclusions from the reported experiment.

II. General Architecture

The Hyeti microprocessor is a defect tolerant version of the HSURF microprocessor [10]. The HSURF microprocessor is targeted to highly dependable real time control systems and its characteristics have been determined in cooperation with potential customers. This has affected the general architecture and the chosen instruction set as outlined below. Also, to support high dependability, special emphasis has been given to on-line testing features.

The instruction set includes the following classes:

1) arithmetic instructions;
2) logic instructions with or without mask;
3) shift and rotate instructions;
4) comparison instructions with or without mask;
5) transfer instructions with or without mask;
6) instructions with implicit operand;
7) branch and jump instructions;
8) stack instructions;
9) interrupt instructions;
10) test instructions (for on-line test);

The masked instructions allow the processing of selected bits in a 16-bit word. The above instructions employ the following addressing modes: relative, inherent, immediate, direct, register indirect, and register indirect with post or pre-modification. For the register indirect addressing modes, the indirection register is automatically modified by adding a programmable offset. These modes were introduced to support matrix processing.

The Hyeti chip contains 50 000 transistors in a total area of 35 mm^2 and is implemented in the 1.2 μ double metal technology of SGS-Thomson Microelectronics. The general organization is depicted in Fig. 1 and the microphotograph is shown in Fig. 2. These two figures show the two major parts, namely, the control part, which consists mainly of PLA's, and the data processing part, which was designed in a bit-slice style as detailed below. Fig. 1 also shows the circuitry supporting the on-line test feature. The signature device allows the compaction of the information on the bus. It is a multiple input linear feedback shift register with a programmable characteristic polynomial. The dedicated on-line test instructions verify the instruction sequences based on the "adjusted signature" method [10].

A. Data Path

The data path of a microprocessor contains several functional units like registers, the arithmetic and logic unit (ALU), bus circuitry etc. Almost all the units in the data path have circuits that are replicated n times (where n is the number of bits in the data path) leading to the classical bit-slice organization. This regular organization can be exploited for yield enhancement by providing spare slices which can replace

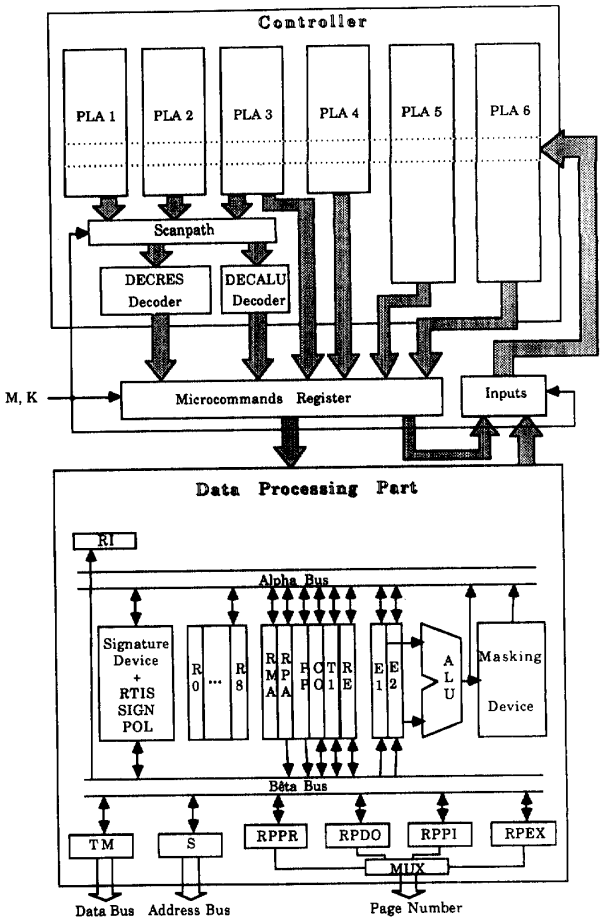


Fig. 1. The general organization of the Hyeti microprocessor. Reprinted with permission by Elsevier Science Publishers ©1990.

defective slices. One must be aware, though, that not all circuits in the data path consist of n completely identical subcircuits. An example for such an irregular circuit is the status register, where each bit is associated with a unique random logic. Such an irregular structure necessitates expensive solutions like duplication (or triplication). Therefore, the decision was not to provide redundancy within the status register, but to make this register a part of the nonreconfigurable portion of the data path.

After having decided on the bit-slice organization of the data path for yield enhancement, it was still necessary to decide on two other issues. One was the number of spare slices to be provided, and the other was the width of each slice, which can be set at one, two, or even four bits. Having a two bit wide spare slice might prove to be more cost-effective than two single bit spare slices in the case of clustered faults. Often two adjacent (single) bit slices will be faulty, and the switching circuitry for a single two-bit spare slice might be simpler and less area consuming than that for two single-bit spare slices.

The data path in Hyeti has been designed with 17 single bit slices, one of which serves as a spare slice. This is the most cost-effective design as shown in Section IV. The

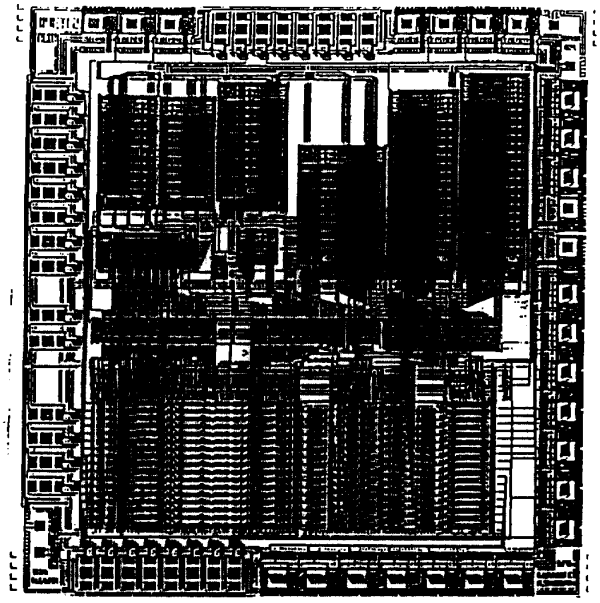


Fig. 2. The microphotograph of the Hyeti microprocessor. Reprinted with permission by Elsevier Science Publishers ©1990.

block diagram of the designed data path is depicted in Fig. 3, showing that each of the 16 slices consists of a right and left reconfigurable blocks and a nonreconfigurable block. The spare slice, numbered 16 in the figure, contains only the two right and left spare blocks. The switches shown in Fig. 3 and the method for bypassing defective slices are described in Section III.

B. Control Unit

Two approaches to the design of regular control units that allow the introduction of defect tolerance in a straightforward manner exist. The first is a hardwired control that is further partitioned into PLA's, where each PLA can be designed to support defect tolerance. The other is a microprogrammed unit where the control memory can have spare rows/columns in the same way as memory IC's. However, a microprogrammed unit requires additional circuitry like a sequencer, a microinstruction register, etc., which cannot employ the same simple strategy.

The decision between the two alternatives should depend, in general, on the desired performance of the microprocessor, its application and the need to allow future modifications and not on yield considerations only. The objective of the Hyeti project was defect tolerance and consequently, the decision was to design a hardwired control unit which can be implemented using PLA's only. Instead of implementing a single PLA which would be technically infeasible in addition to being very large and very sparse, eight PLA's, two of them decoders, were designed consuming an overall smaller silicon area. The decomposition into smaller PLA's is based on the method presented in [12] that minimizes the global size and the area needed for the interconnections.

The regular structure of a PLA allows a straightforward incorporation of redundancy for yield enhancement through

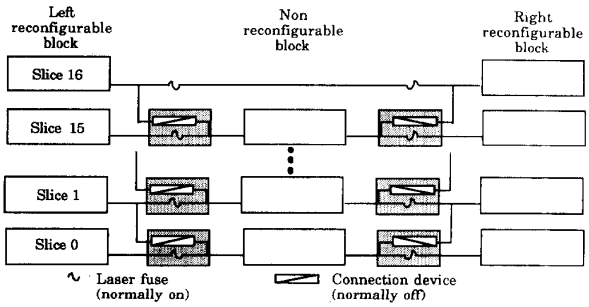


Fig. 3. The reconfigurable data path with a single redundant slice. Reprinted with permission by Elsevier Science Publishers ©1990.

the addition of spare product terms [9], [17], [18]. The number of spare product terms must be decided on separately for each PLA. Also, the design of the PLA must be modified to allow the identification of defective product terms. These two issues are discussed in the next section.

III. Testing and Reconfiguration Strategies

Once the general architecture of the two separate parts of the Hyeti was determined, strategies needed to be developed for the diagnosis of defective circuits and the corresponding reconfiguration for their replacement by spare circuits. Hardware support for these two, including built-in testing aids and reconfiguration switches, needed to be designed as well. Three types of switches have been considered in the preliminary phase of the Hyeti project, namely, laser fuses, antifuses and floating-gate FETs. The first two can be used when no reversibility is required while the third should be employed when several programming attempts may be necessary. However, only laser fuses were used in the end in an attempt to reduce the number of different types of switches, which in turn reduced the different reconfiguration steps and the corresponding number of machines in the production line. In what follows we outline the testing and reconfiguration strategies and the support circuitry for the data path and the control unit separately.

A. Data Path

Two issues require special attention when a bit-sliced design with spare slices is employed. One is the external connections of the 16 good slices; we must make sure that only the 16 fault-free slices will be connected. The second is the routing of signals between slices which must bypass a defective slice if one exists. As shown in Fig. 3, the data path contains a nonreconfigurable part and two reconfigurable blocks. Also, the data path includes two buses which are complemented precharged buses requiring both uncomplemented and complemented lines. Hence, four sets of switches are needed, consisting of laser fuses (normally on), some of which will eventually be cut, and connection devices (normally off). The connection devices can be implemented using antifuses. However, since it was decided not to use antifuses in the Hyeti project, connections using transfer gates were used instead.

Each slice has a special reconfiguration control signal, denoted by $Crec_i$, indicating whether the corresponding slice

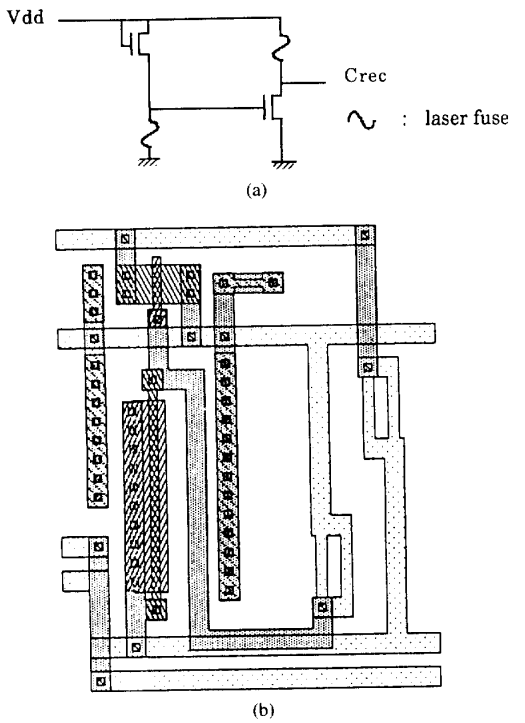


Fig. 4. (a) The circuit generating the signal $Crec$. (b) Its layout. Reprinted with permission by Elsevier Science Publishers ©1990.

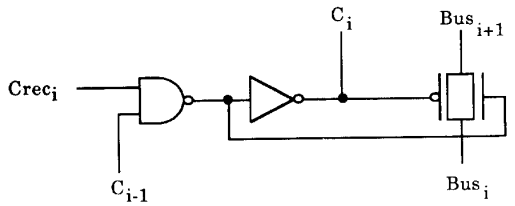


Fig. 5. The control circuit for the transfer gate. Reprinted with permission by Elsevier Science Publishers ©1990.

must be bypassed. The value of $Crec_i$ is determined by a laser fuse as shown in Fig. 4. On slice 0, the transfer gates are controlled directly by the $Crec_0$ signal while on the other slices the control gates are controlled by the following signal:

$$C_i = Crec_i \cdot C_{i-1}; \quad 1 \leq i \leq 15, \quad \text{with } C_0 = Crec_0$$

as depicted in Fig. 5. The $Crec_i$ signals also control the propagation of signals between slices bypassing any defective slice, as shown in Fig. 6.

B. Control Unit

The eight PLA's were initially synthesized using the ASYL system [15]. They were then modified to allow the identification of defective elements by adding inputs to the AND plane of the PLA. The additional inputs are employed only during the testing phase and enable the activation of individual product terms. This strategy is based on the PLA testing scheme presented in [1], [2], and [3]. A more detailed description of the design of a fully testable PLA appears in [17].

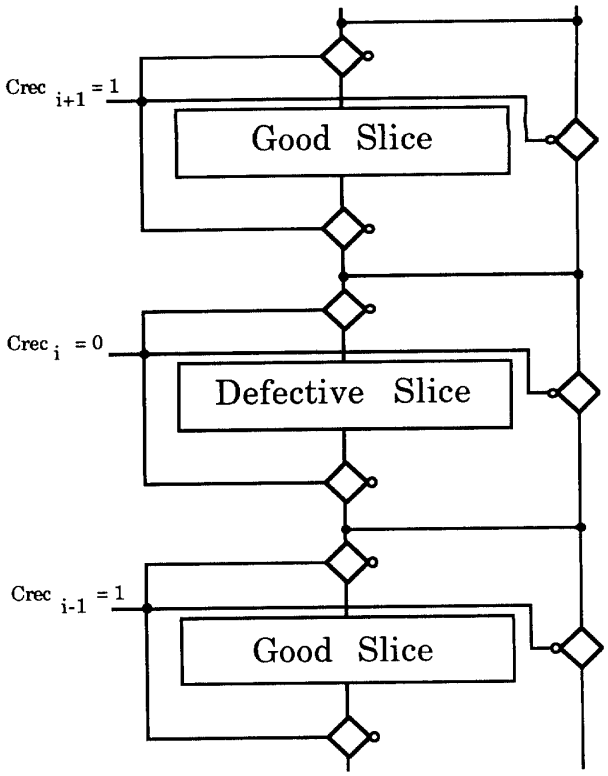


Fig. 6. Bypassing defective slices in the data path. Reprinted with permission by Elsevier Science Publishers ©1990.

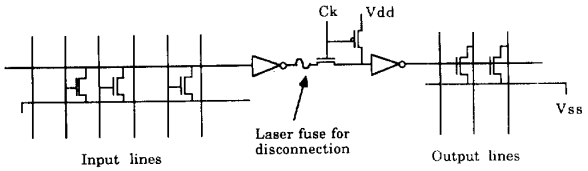


Fig. 7. An ordinary product term in a reconfigurable PLA. Reprinted with permission by Elsevier Science Publishers ©1990.

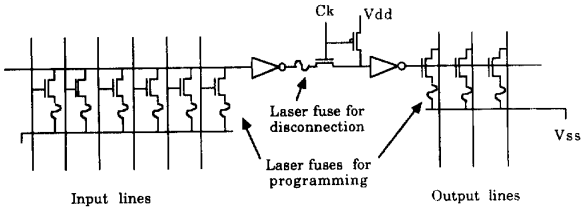


Fig. 8. A programmable product term in a reconfigurable PLA. Reprinted with permission by Elsevier Science Publishers ©1990.

A reconfiguration scheme that allows the replacement of defective product terms by spare product terms is required. This translates to the disconnection of a defective product term and programming a spare product term so that it realizes the given product term. Since reversibility of these two steps is not required, laser fuses were used in both cases. Fig. 7 depicts an ordinary product term including a laser fuse for disconnecting it once it has been identified as defective. Fig. 8 shows a spare product term that includes laser fuses for programming

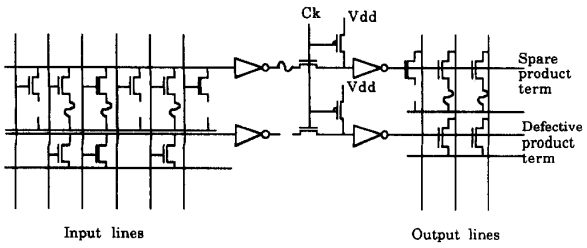


Fig. 9. Reconfiguration in a PLA. Reprinted with permission by Elsevier Science Publishers ©1990.

it to realize the required term and an additional fuse to disconnect it in case it is defective. Such a spare product term has no effect on the PLA before being programmed since it is connected to all true and complemented inputs. Fig. 9 illustrates the entire reconfiguration scheme including the disconnection of a defective product term and the programming of a spare product term.

The above strategy can handle product term failures but not (AND/OR) column failures. Redundant input or output lines were not added due to their large area overhead [18]. Instead, input and output lines were laid out on two layers that were connected at multiple points, protecting against open-circuit defects in the columns at the cost (due to increased capacitance) of a small reduction in speed.

C. The Testing Procedure

Three scanpath registers were included in the design, with two of them part of the interface between the controller and the data path and the third one inside the controller. The signals M and K shown in Fig. 1 control these scanpath registers. The scanpath registers together with the easily testable PLA's and a special design of the ALU as a C-testable iterative logic array result in a very efficient off-line testing. The testing procedure applied after manufacturing consists of the following steps:

1) parametric test (general validation of the chip);
2) test of the three scanpath registers;
3) test of each PLA in the control unit using the scanpath registers;
4) test of the data path using the scanpath registers;
5) test of the whole microprocessor in working mode.

If defects are detected in the reconfigurable parts in steps 3 or 4 of the above procedure, a reconfiguration phase is performed. Defective product terms in PLA's are replaced by spare ones by blowing the appropriate laser fuses. If a defect in slice i of the data path is detected, then this data path slice is bypassed by setting the Crec_i signal to 0 and blowing the laser fuses on the bus lines i, $i+1, \cdots, 16$ (see Figs. 3 and 6).

IV. Yield Analysis

The main purpose of incorporating redundancy into the design of a microprocessor is yield enhancement. We demonstrate in this section how the optimal amount of redundancy can be determined in order to maximize the yield. Clearly, yield enhancement via redundancy increases the chip area and reduces the number of chips that will fit into the given wafer area, so that marginal yield improvement becomes smaller as more redundant elements are added. The optimal amount of redundancy to be incorporated is, therefore, determined by maximizing the ratio between the yield and the area increase factor, called effective yield (or wafer-level yield [4]), rather than the yield itself. However, other constraints on the overall chip area may prevent the implementation of the optimal redundancy. In the `Hyeti` project the overall area devoted to built-in testing features and to redundancy for yield enhancement was restricted to no more than 25%.

The yield model we employ is the negative binomial model under the large area clustering assumption [6] with an average of λ faults per unit area and a clustering parameter α. We first introduce a few notations that are used in the yield analysis. We use the notion of *module* for a circuit that is replicated several times in the design and for which spares are provided. A module is a one or two bit slice in the case of the data path or a product term in the case of a PLA. Let N and R denote the number of standard modules and spare modules, respectively.

Let A_m and A_s denote the area of a standard module and a spare module, respectively. These two areas are not necessarily equal, e.g., the area of a spare product term in a PLA is larger than the area of a standard product term. Let A_{ck} denote the area of the support circuitry that has no redundancy and consequently, each fault occurring in it is a *chip-kill* fault [6]. Finally, let A_{sw} denote the area of the switching (reconfiguration) circuitry. As in the support circuitry, every fault in this area is fatal; but, unlike A_{ck}, the size of this area strongly depends on R. The total area of the microprocessor part under consideration is denoted by $A(N,R)$ and is given by

$$A(N,R) = N \cdot A_m + R \cdot A_s + A_{\mathrm{ck}} + A_{\mathrm{sw}}(R) \qquad (1)$$

When the circuit consists of only one type of module, the yield expression (i.e., the probability of at least N good modules out of $N+R$) for the negative binomial model is [6]

$$Y(N,R) = \sum_{j=0}^{R} (-1)^j \binom{N+R}{N+j} \binom{N+j-1}{j} \times \left(1 + \frac{\lambda[N \cdot A_m + j \cdot A_s + A_{\mathrm{ck}} + A_{\mathrm{sw}}(R)]}{\alpha}\right)^{-\alpha} \qquad (2)$$

and the effective yield to be maximized is

$$Y(N,R) \cdot A(N,0)/A(N,R) \qquad (3)$$

where $A(N,0)$ is the total area with no redundancy whatsoever.

Since the microprocessor under consideration consists of 8 different types of modules, (1)–(3) need to be extended to the multi-dimensional case. We demonstrate this extension for a circuit with K types of modules. The total area of the circuit is

$$A(N_1,R_1,\cdots,N_K,R_K) = \sum_{i=1}^{K} \left(N_i \cdot A_m^{(i)} + R_i \cdot A_s^{(i)} + A_{\mathrm{ck}}^{(i)} + A_{\mathrm{sw}}^{(i)}(R_i)\right). \qquad (4)$$

TABLE I
PARAMETERS OF THE SEVEN RECONFIGURABLE PLA'S

PLA name	PLA1	PLA2	PLA3	PLA4	PLA5	PLA6	DECALU
No. of inputs	15	15	12	26	26	26	4
No. of test inputs	3	2	3	3	3	3	1
No. of outputs	7	5	29	23	3	3	10
No. of product terms	96	94	101	108	166	178	13
A_m (mm^2)	0.0047	0.0044	0.0060	0.0072	0.0057	0.0057	0.0033
A_s (mm^2)	0.0304	0.0286	0.0390	0.0470	0.0370	0.0370	0.0212
A_{ck} (mm^2)	0.1558	0.1467	0.1999	0.2406	0.1896	0.1896	0.1089
$A_{sw}(module)$ (mm^2)	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002
$A_{sw}(spare)$ (mm^2)	0.0012	0.0013	0.0012	0.0012	0.0012	0.0012	0.0013

The yield of the circuit is

$$Y(N_1,R_1,\cdots,N_K,R_K) = \sum_{j_1=0}^{R_1}\cdots\sum_{j_K=0}^{R_K}(-1)^{\sum_{i=1}^{K}j_i} \cdot\prod_{i=1}^{K}\binom{N_i+R_i}{N_i+j_i}\binom{N_i+j_i-1}{j_i} \times\left(1+\frac{\lambda\sum_{i=1}^{K}\left[N_i\cdot A_m^{(i)}+j_i\cdot A_s^{(i)}+A_{\rm ck}^{(i)}+A_{\rm sw}^{(i)}(R_i)\right]}{\alpha}\right)^{-\alpha} \quad (5)$$

and the effective yield is

$$Y(N_1,R_1,\cdots,N_K,R_K)\cdot A(N_1,0,\cdots,N_K,0)/A(N_1,R_1,\cdots,N_K,R_K) \quad (6)$$

The various parameters in equations (4) and (5) assume different values for the data path and the control unit. For the data path, $A_m = A_s = 0.202356\,\mathrm{mm}^2$, $A_{\rm ck} = 2.410636\,\mathrm{mm}^2$ and $A_{\rm sw}(R) = 0.046277\cdot(16+R)\,\mathrm{mm}^2$, if both left and right reconfigurable blocks are disconnected at the same time. In the control unit, redundancy has been incorporated in only seven out of the eight PLA's. The 8th PLA is really a decoder and, as such, has no OR-plane. As a result, the method of adding spare product terms is not applicable. Any other redundancy scheme would result in a substantial area overhead and could not be justified. The parameters of the seven reconfigurable PLA's are summarized in Table I.

We have performed a detailed yield analysis of the microprocessor and some of the highlights of this analysis are presented next. We first searched for the redundancy in the data path and in the control unit that would maximize the effective yield. This search was performed for several values of λ and α and, more significantly, for different values of the area of circuitry added to the microprocessor. An important application of the Hyeti microprocessor is as a controller of an application-specific microprocessor-based integrated circuit. The additional circuitry to be controlled by the microprocessor may or may not have some built-in redundancy. If no redundancy is incorporated in the additional circuitry, then its area should be added to the chip-kill area, $A_{\rm ck}$. In either case, as the additional circuitry grows, the redundancy incorporated in Hyeti becomes more important and consequently, the optimal value of the redundancy goes up. However, the yield and the effective yield of the whole chip are expected to decrease with the total area. This is demonstrated in Fig. 10 which depicts the effective yield without redundancy in the microprocessor and with the optimal redundancy as a function of the added area (used for the extra circuitry controlled by the microprocessor) for two values of the defect density, namely, $\lambda = 0.02/\mathrm{mm}^2$ and $\lambda = 0.05/\mathrm{mm}^2$. The corresponding optimal redundancy is shown in Table II.

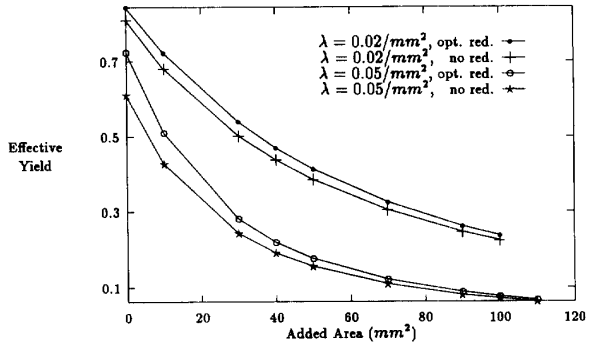


Fig. 10. The effective yield as a function of the added area, without redundancy and with optimal redundancy, for two values of λ ($\alpha = 2$).

In particular, it should be noted that the optimal redundancy for the data path is a single 1-bit slice. A 2-bit slice did not prove to provide a higher effective yield. We also tested the idea of controlling separately the switching of the right and left reconfigurable blocks in each slice (see Fig. 3). This separation could allow the data path to tolerate one fault in the right reconfigurable block of one slice and another fault in the left reconfigurable block of a different slice. However, our yield analysis showed that for the range of practical values of λ, α and the other parameters, the improvement in the yield due to such a separate control is much smaller than the area increase required for implementing it.

The increase in effective yield due to the added optimal redundancy shown in Table II, is summarized in Table III. This table presents more accurately the results depicted in Fig. 10. The effective yield goes up for an added area of 30 mm^2,

TABLE II
THE OPTIMAL REDUNDANCY FOR $\lambda = 0.02/\text{mm}^2$ AND $\lambda = 0.05/\text{mm}^2$

	PLA1	PLA2	PLA3	PLA4	PLA5	PLA6	DECALU	Data Path
Added Area	$\lambda = 0.02/mm^2$							
0 mm^2	1	1	1	1	1	1	0	0
10 mm^2	1	1	1	1	1	1	0	1
50 mm^2	1	1	1	1	1	1	0	1
70 mm^2	1	1	1	1	1	1	1	1
110 mm^2	1	1	1	1	1	1	1	1
Added Area	$\lambda = 0.05/mm^2$							
0 mm^2	1	1	1	1	1	1	0	1
10 mm^2	1	1	1	1	1	1	0	1
30 mm^2	1	1	1	1	1	1	1	1
110 mm^2	1	1	1	1	1	1	1	1

TABLE III
THE PERCENTAGE INCREASE IN EFFECTIVE YIELD DUE TO THE ADDED REDUNDANCY

Added area (mm^2)	0	10	30	40	50	70	90	100
% eff. yield increase $\lambda = 0.02/mm^2$	4.15	6.05	7.39	7.28	7.20	6.59	6.23	5.60
% eff. yield increase $\lambda = 0.05/mm^2$	21.53	19.60	16.05	14.84	13.46	11.10	10.00	9.00

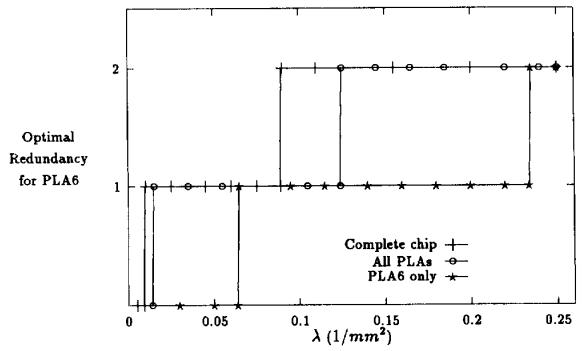


Fig. 11. The optimal redundancy for PLA6 as a function of λ for PLA6 only, as part of all PLA's and as part of the entire chip including the data path.

by roughly 7% for $\lambda = 0.02/\text{mm}^2$ and by roughly 16% for $\lambda = 0.05/\text{mm}^2$.

The search for the optimal redundancy in the Hyeti microprocessor (using (4)–(6)) is an eight-dimensional search and consequently, proved to be very time-consuming, especially when the negative binomial yield model is employed rather than the simple Poisson model. We decided therefore, to check the validity of the approach which is very often adopted, i.e., searching for the optimal redundancy of each subcircuit separately ((1)–(3)). While this approach greatly reduces the complexity of the search, it implicitly assumes statistical independence among the subcircuits, which is not a valid assumption when the negative binomial distribution is used [6]. Fig. 11 shows the optimal redundancy for PLA6 as a function of λ when taking into consideration the whole microprocessor, all the PLA's (excluding the data path), or PLA6 only.

This figure clearly demonstrates that searching for the optimal redundancy for each subcircuit separately, or even for a subset of subcircuits, leads to a suboptimal solution. However, as is evident from Fig. 11, the redundancy which maximizes the effective yield of a single PLA is smaller than that dictated for the same PLA when maximizing the effective yield of all PLA's, which, in turn, is smaller than the optimal redundancy for the same PLA when the effective yield of the whole chip is considered. Thus, the redundancies obtained from the separate analyses of the different subcircuits can serve as a lower bound and as a starting point for the multi-dimensional search. An upper bound for the optimal redundancy vector can be obtained as well, by using the Poisson distribution for which the calculations are very fast, but the yield tends to be more pessimistic. These two bounds can shorten the search time considerably.

V. DISCUSSION

The redundancy that has been added to the seven PLA's and the data path in the Hyeti microprocessor is [13]: (PLA1, PLA2, PLA3, PLA4, PLA5, PLA6, DECALU, DATA_PATH) = (2, 2, 2, 2, 4, 4, 1, 1).

This is higher than the calculated optimal redundancy as presented in the previous section even for high values of the added area and/or the average number of defects λ. A higher than optimal redundancy was implemented in some of the PLA's since the floorplan of the control unit (shown in Fig. 2) allows the addition of a few extra product terms to the PLA's with no area penalty. The extra product terms use part of the otherwise wasted area and can increase the yield of the chip.

Consequently, a more practical yield analysis should take into consideration the exact floorplan of the chip. We must distinguish between two types of reconfigurable units. The first type includes those units where any added redundancy would increase the total chip area, e.g., the data path in the Hyeti microprocessor. The second type includes units for which a certain amount of redundancy can be added without

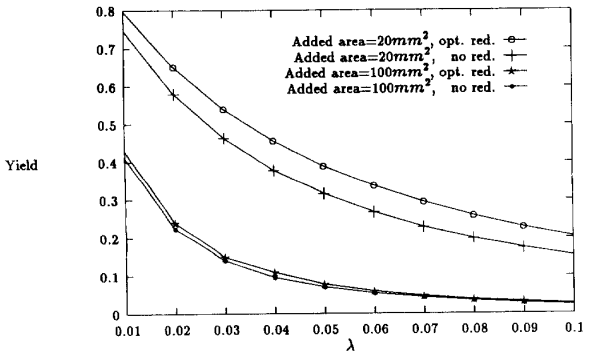


Fig. 12. The yield as a function of λ, without redundancy and with optimal redundancy, for two values of the added area.

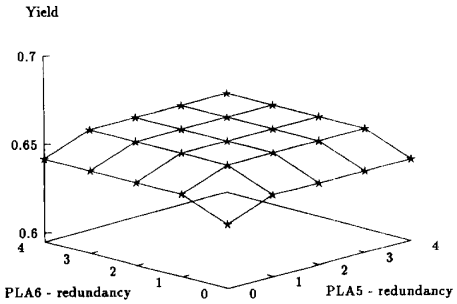


Fig. 13. The yield as a function of the number of redundant product terms in PLA5 and PLA6 ($\lambda = 0.02$ and $\alpha = 2$).

affecting the overall chip area. In the Hyeti microprocessor all seven PLA's (PLA1 through DECALU) in the control unit belong to the second type. The position of these PLA's in the floorplan of the Hyeti as depicted in Fig. 2 is the same as that shown in Fig. 1. This floorplan allows the addition of a limited number of redundant product terms to the seven PLA's without changing the overall chip area.

For the first type of unit, effective yield must be used while for the second type, the yield itself should be used as long as the added redundancy does not change the chip area. Once the additional redundancy increases the overall chip area, the effective yield measure must be used.

When considering effective yield, we obtained as optimal redundancy the addition of a single spare product term per PLA, for most of the applicable parameter space. To check whether any redundancy beyond this level will significantly increase the yield (when no change in the chip area is assumed), we calculated the yield, rather than the effective yield, for larger numbers of redundant product terms in the seven PLA's. The redundancy in the data path was kept at the level of a single bit slice, since any additional bit slice would increase the chip area. The results of the yield calculations are depicted in Fig. 12 where the yield is shown for a chip without any redundancy and with the optimal redundancy of (PLA1, PLA2, PLA3, PLA4, PLA5, PLA6, DECALU, DATA_PATH) = (1, 1, 1, 1, 1, 1, 1, 1).

The curve for any other organization with a higher level of redundancy (i.e., at least one PLA having more than a single redundant product term) was extremely close to the curve for the redundancy (1, 1, 1, 1, 1, 1, 1, 1). In other words, any further increase in the redundancy within the control unit resulted in only a marginal increase in the yield. This was found to be the case for any λ in the range of [0.01, 0.1]/mm^2 and for added area in the range of [20, 100] mm^2.

Fig. 13 illustrates the marginal improvement in yield when the number of redundant product terms in a PLA is higher than 1. The overall yield is shown as a function of the number of redundant product terms added to the two largest PLA's, PLA5 and PLA6, while the remaining PLA's and the data path have their optimal amount of redundancy, i.e., 1.

The conclusion that can be drawn is that even when extra redundancy does not result in an actual increase in the area of a chip, it does not necessarily imply that all the available area should be used up for spares. These spares still increase the chip-kill area, A_{ck}, since the switching area, $A_{sw}(R)$, will increase. The higher chip-kill area can offset the yield increase due to the added redundancy.

VI. Conclusion

We have reported in this paper on an experiment in designing a defect tolerant microprocessor and analyzed the improvement in yield due to added redundancy. This experiment demonstrates the feasibility of incorporating defect tolerance for yield enhancement in the design of a microprocessor, and proves that the use of defect tolerance is not limited to the highly regular memory integrated circuits but can be extended to some "random logic" designs as well.

The detailed yield analysis has demonstrated that it is suboptimal to find the best amount of redundancy for each unit within the chip by considering its yield separately, and a complete yield analysis of the whole chip must be performed. Another important conclusion is that the effective yield is not always the correct measure for deriving the optimal redundancy. An exact yield analysis must take into consideration the exact floorplan of the chip, and use a combination of the raw yield and the effective yield whenever some empty space in the chip layout can be used for adding redundant elements without increasing the chip area.

Finally, restricting the overall area overhead (used for testing, reconfiguration and redundant circuits) to 25% proved to have no practical impact on the final design since the optimal overhead turned out to be less than 25%. Moreover, when the effective yield is used as objective function, there is no need to restrict the overall overhead. This measure will determine the overhead so that the total number of good chips out of a wafer is maximized, and any artificial restriction will not be beneficial.

Acknowledgment

The Hyeti microprocessor has been developed under the ESPRIT 824 grant supervised by Dr. J. Trilhe from SGS-Thomson Microelectronics. The authors wish to thank M. Glesner, M. Soueidan and M. Karam for their important contributions.

REFERENCES

[1] E. H. L Aarts, F. P. M. Beenker, and M. M. Ligthart, "Design for testability of PLA's using statistical cooling," in *Proc. 23rd Design Automat. Conf.*, 1986.

[2] S. Bozorgui-Nesbat and E. J. McCluskey, "Lower overhead design for testability of PLA's," in *Proc. Int. Test Conf.*, 1984.

[3] S. Bozorgui-Nesbat and J. Khabaz, "Minimizing extra hardware for fully testable PLA design," *Proc. ICCAD*, 1985.

[4] I. Koren, Z. Koren, and D. K. Pradhan, "Designing interconnection buses in VLSI and WSI for maximum yield and minimum delay," *IEEE J. Solid-state Circ.*, vol. 23, pp. 859–866, June 1988.

[5] I. Koren and A. D. Singh, "Fault tolerance in VLSI circuits," *Computer*, (special issue on Fault-Tolerant Systems), vol. 23, pp. 73–83, July 1990.

[6] I. Koren and C. H. Stapper, "Yield models for defect tolerant VLSI circuits: A review," *Defect and Fault Tolerance in VLSI Systems*, I. Koren, Ed. New York: Plenum, vol. 1, 1989, pp. 1–21.

[7] Z. Koren and I. Koren, "A model for enhanced manufacturability of defect tolerant integrated circuits," in *Proc. 1991 IEEE Int. Workshop on Defect and Fault Tolerance in VLSI Syst.*, Nov. 1991, pp. 81–92.

[8] M. Kuboschek, H. J. Iden, U. Jagau, and J. Otte, "Implementation of a defect tolerant large area monolithic multiprocessor system," in *Int. Conf. Wafer Scale Integrat.*, 1992, pp. 28–34.

[9] S. Y. Kuo and W. Kent Fuchs, "Fault diagnosis and spare allocation for yield enhancement in large reconfigurable PLA's," *IEEE Trans. Comput.*, vol. 41, pp. 221–226, Feb. 1992.

[10] R. Leveugle, T. Michel, and G. Saucier, "Design of microprocessors with built-in on-line test," in *Proc. 20th Fault Tolerant Computing Symp.*, June 1990.

[11] R. Leveugle and G. Saucier, "Logical and topological optimization of multi-PLA large controllers," in *Proc. Int. Workshop on Logic Synthesis*, May 1989.

[12] R. Leveugle and M. Soueidan, "Design of an application specific microprocessor," in *Logic and Architecture Synthesis for Silicon Compilers*, G. Saucier and P. M. McLellan, Eds. New York: Elsevier Science Publishers, 1989, pp. 255–268.

[13] R. Leveugle, M. Soueidan, and N. Wehn, "Defect tolerance in a 16-bit microprocessor," *Defect and Fault Tolerance in VLSI Systems*, vol. 1, I. Koren, Ed. New York: Plenum, 1989, pp. 179–190.

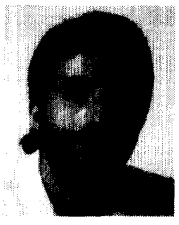
[14] ——, "A practical experiment in defect tolerance: The Hyeti 16-bit microprocessor," *Wafer Scale Integration III*, M. Sami and F. Distante Eds. New York: Elsevier Science Publishers, 1990, pp. 317–332.

[15] G. Saucier, M. Crastes dePaulet, and P. Sicard, "ASYL: A rule based system for controller synthesis," *IEEE Trans. Comput.-Aided Design*, vol. CAD-6, Apr. 1987.

[16] G. Saucier, J-L. Patry, A. Boubekeur, and E. Sanlaville, "Practical experiences in the design of a wafer scale 2-D array," *Defect and Fault Tolerance in VLSI Systems*, vol. 2, C. H. Stapper, V. K. Jain and G. Saucier (eds.), pp. 75–87, Plenum 1990.

[17] N. Wehn, M. Glesner, K. Caesar, P. Mann, and A. Roth, "A defect tolerant and fully testable PLA," in *Proc. 25th Design Automat. Conf.*, 1988, pp. 22–27.

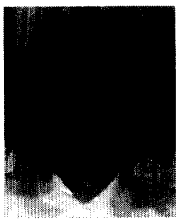
[18] C. L. Wey, "On yield considerations for the design of redundant programmable logic arrays," *IEEE Trans. Comput.-Aided Design*, vol. CAD-7, pp. 528–535, Apr. 1988.



Regis Leveugle (M'92) received the Ph.D. degree in microelectronics from the National Polytechnical Institute of Grenoble (INPG), France, in 1990.

He is currently an Assistant Professor at this institute. His main interests are computer architecture, VLSI design and test methods, fault-tolerant architectures and concurrent checking.

He is a member of AFCET.



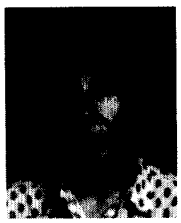
Zahava Koren received the B.A. and M.A. degrees in mathematics and statistics from the Hebrew University in Jerusalem in 1967 and 1969, respectively, and the D.Sc. degree in operations research from the Technion—Israel Institute of Technology in 1976.

She is currently with the Department of Industrial Engineering and Operations Research at the University of Massachusetts, Amherst. Previously she has held positions with 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 in Los Angeles. Her main interests are Stochastic Analysis of Computer Networks, Yield of Integrated Circuits and Reliability of Computer Systems.

Israel Koren (S'72–M'76–SM'87–F'91) received the B.Sc., M.Sc. and D.Sc. 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 Departments of Electrical Engineering and Computer Science at 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 Intel, Digital Equipment Corp., National Semiconductor, Tolerant Systems, and ELTA. His current research interests are Fault-Tolerant VLSI Architectures, Models for Yield and Performance, Floor-planning of VLSI chips and Computer Arithmetic.

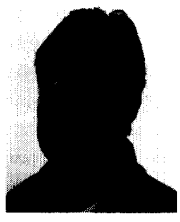
Dr. Koren has published extensively in the IEEE TRANSACTIONS. He was a Co-Guest Editor for IEEE TRANSACTIONS ON COMPUTERS, special issue on High Yield VLSI Systems, April 1989. Since January 1992, he serves on the Editorial Board of these Transactions. He also served as Program Committee member for numerous conferences. He has edited and co-authored the book, *Defect and Fault-Tolerance in VLSI Systems*, (Vol. 1, Plenum, 1989). He is the author of the textbook *Computer Arithmetic Algorithms*, (Prentice-Hall, 1993).



Gabriele Saucier (M'75–SM'77–F'91) received the Ph.D. degree from the Institut National Polytechnique de Grenoble, France, in 1970.

Presently, she is a Professor at the same institute and she heads a research laboratory whose main interests are synthesis and test of VLSI integrated systems on silicon, including fault and defect tolerant architectures and massively parallel architectures.

She is a member of the IEEE Computer Society.



Norbert Wehn was born in Neustadt/Aisch, Germany, in March 1959. He received the engineering degree and the Ph.D. degree in electrical engineering from the Darmstadt University of Technology, Germany, in 1984 and 1989, respectively.

Since 1984, he is working in the VLSI group at the Darmstadt University of Technology, Germany. From 1984 to 1989, he concentrated his activities on CAD for physical design problems and defect-tolerant circuits. In 1990, he moved his interests to architectural synthesis problems. In February 1991, he joined the Siemens synthesis group and is now sharing his job between university and industry.

Dr. Wehn has authored or coauthored more than 20 papers on international conferences and journals in the cited fields.