Modeling the Effect of Redundancy on Yield and Performance of VLSI Systems

ISRAEL KOREN, MEMBER, IEEE, AND DHIRAJ K. PRADHAN, SENIOR MEMBER, IEEE

Abstract—The incorporation of different forms of redundancy has been recently proposed for various VLSI and WSI designs. These include regular architectures, built by interconnecting a large number of a few types of system elements on a single chip or wafer. The motivation for introducing fault-tolerance (redundancy) into these architectures is two-fold: yield enhancement and performance (like computational availability) improvement.

Our objective in this paper is to develop analytical models that evaluate how yield enhancement and performance improvement may both be achieved by introducing redundancy into VLSI and WSI designs. Such models also allow us to evaluate the costeffectiveness of a given fault-tolerance strategy and calculate the amount of redundancy to be added.

Index Terms—Computational availability, fault tolerance, redundancy, reliability, VLSI designs, wafer-scale integration, yield.

I. INTRODUCTION

IMPORTANT innovations are likely to occur in two VLSI-based areas, namely, wafer-scale integrated architectures, and single VLSI chip/multielement architectures. The former has the potential for a major breakthrough with its ability to realize a complete multiprocessing system on a single wafer. This will eliminate the expensive steps required to dice the wafer into individual chips and bond their pads to external pins. In addition, internal connections between chips on the same wafer are more reliable and have a smaller propagation delay than external connections. The latter does make it possible to build a high-speed processor on a single chip, designed by interconnecting a large number of simple processing elements, memory modules and the like. These architectures already have captured the imagination of several computer manufacturers and researchers alike.

Much recent research has focused on these new architectural innovations, especially those created by interconnecting a large number of elements such as processors, memories, switches, communication links etc, all on a single chip or wafer. Concerns about fault tolerance in such VLSI-based systems stem from the two key factors of performance and yield enhancements.

IEEE Log Number 8611446.

Low yield (expected percentage of good chips out of a wafer) is a problem of increasing significance as circuit density grows. One solution suggests improvement of the manufacturing and testing processes, to minimize manufacturing faults. However, this approach is not only very costly, but also quite difficult (or even impossible) to implement, with the increasing number of components that can be placed on one chip. But incorporating redundancy for fault tolerance does provide a very practical solution to the low yield problem. This has been demonstrated in practice for high density memory chips (e.g., [1]) and should be extended to other types of VLSI circuits. In general, yield may be enhanced because the circuit can be accepted, in spite of some manufacturing defects, by means of restructuring, as opposed to having to discard the faulty chip.

Achieving reliable operation also becomes increasingly difficult with the growing number of interconnected elements and hence, the increased likelihood that faults can occur. Here too, redundant elements which are ready to replace faulty ones when the system is in operation, can increase the reliability and other performance measures like computational availability.

In summary, the justification for introducing fault tolerance (redundancy) into the architecture of VLSI-based systems is two-fold. One is to deal with manufacturing flaws and increase the yield. The other is to deal with operational faults and enhance the performance availability.

Our objective in this paper is to formulate analytical models that will enable us to analyze the effectiveness of a given faulttolerance technique in increasing yield and improving performance, or find the tradeoff between the two. These models will also allow us to compare various fault tolerance techniques, examine different system topologies and determine the optimal amount of redundancy to be added.

In the next section, the aspects that have to be considered when evaluating a fault tolerance strategy are detailed. In Section III, expressions for the actual and apparent yield of VLSI chip with added redundancy are derived. In Section IV we present models that allow us to compute various measures of combined performance and reliability. Then, an example of a VLSI-based system with redundancy is analyzed in Section V and final conclusions are presented in Section VI.

II. FAULT-TOLERANCE IN VLSI AND WSI

A variety of techniques for introducing fault tolerance into VLSI and WSI architectures with regular topologies have been recently proposed, [2], [3], [6], [7], [10], [15], [17], [18]. Because fault tolerance is an involved subject, completely

Manuscript received August 1, 1985; revised February 4, 1986 and June10, 1986. This work was supported in part by the United States Air Force Office of Scientific Research under Grant 84-0052 and by the National Science Foundation under Grant DCR-8509423.

I. Koren is with the Department of Electrical and Computer Engineering, University of Massachusetts, Amherst, MA 01003, on leave from the Technion—Israel Institute of Technology, Haifa 32000, Israel.

D. K. Pradhan is with the Department of Electrical and Computer Engineering, University of Massachusetts, Amherst, MA 01003.

different schemes might be cost-effective in different situations and for different objective functions.

Several aspects have to be considered when evaluating a fault-tolerance strategy for multielement systems. The first is the type of failures to be dealt with. There are two distinct types of failures with which fault-tolerance strategies can be designed to deal. These are production defects and operational faults. A relatively large number of defects is expected when manufacturing a silicon wafer in the current technology. Normally, all chips with production flaws are discarded leading to a low yield.

Operational faults have in comparison a considerably lower probability of occurrence, the difference of which may be in orders of magnitude. Improvements in solid-state technology and maturity of the fabrication processes have reduced the failure rate of a single component within a VLSI chip. However, the exponential increase in the component-count per VLSI chip has more than offset the increase in reliability of a single component. Thus, operational faults cannot be ignored although they have a substantially lower probability of occurrence compared to production defects. Consequently, a fault-tolerance strategy that enables the system to continue processing, even in the presence of operational faults, can be beneficial.

The two types of failures, manufacturing defects and operational faults, also differ in the costs associated with them. Defects are tested for before the IC's are assembled into a system and therefore, they contribute only to the production costs of the IC's. In contrast, faults occur after the system has been assembled and is already operational. Hence, their impact is on the system's operation and their damage might be substantial, especially in systems used for critical real-time applications. Clearly, a method which is cost-effective for handling defects is not necessarily cost-effective for handling operational faults, and vice versa.

For both types of failures in VLSI, a repair operation is impossible and the best one can do is to somehow avoid the use of the faulty part by restructuring the system links so as to isolate it. This restructuring capability can be either static or dynamic. When the manufacturing process is completed, the chips within the wafer are tested to determine the defective ones. A static scheme using for example fusible links or laserformed links, can be then employed to configure out the defective elements and interconnect most of the good ones to form an operational system. Dynamic restructuring is required during the normal system operation, when faulty parts have to be restructured out of the system without human intervention. Such a dynamic strategy might be appropriate to handle defects as well. Comparing the two alternatives, static schemes tend to use less hardware but consume operator time, while dynamic schemes which are controlled internally by the system usually require extra circuitry.

Another aspect that has to be considered when evaluating the effectiveness of a given fault-tolerance technique, is the required hardware investment. The hardware added can be in the form of redundant switching elements, [2], [18], [15] and/ or redundant processors, communication links, or other system elements [3], [10], [6]. When carrying out such an analysis we have to take into account the relative hardware complexity (silicon area) of all system elements, and their susceptibility to failures (manufacturing defects or operational faults).

Processing elements (PE's) are traditionally considered the most important system resource; hence, achieving 100 percent utilization of them is often attempted. For example, in [2], [15], and [18] switching elements are added between processors to assist in achieving this goal. In [3] and [10] connecting tracks are added on the wafer to be used in bypassing the defective PE's when connecting the fault-free ones. However, the silicon area that needs to be devoted to switching elements (e.g., switches capable of interconnecting 4 to 8 separate parallel busses [18]) or to additional communication links cannot be ignored. Consequently, such schemes might be beneficial only for PE's which are substantially larger than the switches and the additional links (e.g., [13]). Also, the addition of switching elements and especially the longer interconnection between active processors result in longer delays affecting the throughput of the system. To overcome this performance penalty, it has been suggested in [9] to add registers for bypassing faulty processors. The effect of this is to introduce extra stages in the pipeline, thus increasing the latency of the pipeline without reducing its throughput.

In the above mentioned schemes, one of the underlying assumptions is that the extra circuitry (e.g., switching elements, communication links or registers) are failure free and only processors can fail. However, larger silicon areas devoted to those elements increase their susceptibility to defects or faults; as a result, the above-mentioned assumption might not be valid any more.

In general, there are several alternative ways for introducing redundancy into the system. Redundancy can be introduced into the architecture at the basic element level and/or at the system level. In the case of system level redundancy, spare elements are added to the original design and they will be used to replace any faulty system element. In the case of element level redundancy, each element has some internal redundancy allowing it to remain operational even in the presence of certain internal faults (with possibly a lower computational capacity). Note that both element level and system level redundancies can be incorporated into the same system.

Several forms of redundancy can be used to handle manufacturing defects to increase the yield. The defective elements are configured out and the good ones are interconnected to form an operational system. Once this procedure is completed, the system goes into operation and it has to handle from this point on only operational faults. At this point the fault-tolerance capacity of the system is used to improve its performance availability. First, the remaining redundant elements (if any) can be used as spares and then, the system is gracefully degraded. We conclude therefore, that the same redundancy can be used for yield enhancement and for performance improvement as well.

III. THE YIELD OF FAULT-TOLERANT CHIPS

To evaluate the effectiveness of redundancy for yield enhancement we need an expression for the yield of a faulttolerant VLSI chip. Such expressions have been presented in [11] and [7]. A more general expression for the yield was proposed in [8]. In what follows we modify the latter to include some simplified yield models (as used in [3] and [10]) and to take into account the effect of incomplete testing on the yield.

The yield of any VLSI chip depends on the types of defects which may occur during the manufacturing process and their distribution. The majority of fabrication defects can be classified as random spot defects [20] caused by minute particles deposited on the wafer. The area of the system elements that we will be considering here and for which we will have spare ones (e.g., processors, memories, busses etc), is substantially larger than the expected area of a spot defect. Consequently, we assume in what follows that each spot defect affects only a single element.

For the statistics of the fabrication defects we can adopt one of the models suggested in the literature like Poisson, binomial, general negative binomial statistics and others. Under proper assumptions each one of these statistics can be used and the "correct" one is the one that fits the data best [20]. One model which has been shown to agree with experimental results, is the generalized negative binomial distribution [19]. Its attractiveness stems from the fact that it does not assume that all defects are evenly distributed throughout the wafer but rather allows defects to cluster. We adopt here this distribution although all our subsequent results remain valid if some other distribution is selected.

The negative binomial distribution has two parameters; d is the average defect density¹ and α is the defect clustering parameter. A low value of α can be used to model severe clustering of defects on a wafer, while for $\alpha \rightarrow \infty$ we obtain the Poisson distribution. These two parameters depend on the number and complexity of manufacturing steps performed. Consequently, different system elements might have different values for these two parameters. Let d_i and α_i be the two parameters for type *i* system elements, then the probability of having x_i defects in the system elements of type *i* on a chip is

$$\Pr \{X_i = x_i\} = \frac{\Gamma(x_i + \alpha_i)}{x_i! \Gamma(\alpha_i)} \cdot \frac{\left(\frac{A_i d_i}{\alpha_i}\right)^{x_i}}{\left(1 + \frac{A_i d_i}{\alpha_i}\right)^{\alpha_i + x_i}} \quad (3.1)$$

where A_i is the total silicon area devoted to type *i* elements within the chip.

The parameters d and α of the yield equation are estimated either by monitoring many wafers or from a few carefully placed test chips on each wafer. Another technique for estimating the parameters of the yield equation has been recently proposed by Seth and Agrawal [16] based on the commulative coverage of test patterns applied to the chips after manufacturing.

For nonredundant chips, the yield is the probability of

having zero defects in all its elements

$$Y = \prod_{i=1}^{n} \Pr \{X_i = 0\} = \prod_{i=1}^{n} \left(1 + \frac{A_i d_i}{\alpha_i}\right)^{-\alpha_i}$$
(3.2)

where n is the total number of different types of elements in the chip.

Note that in (3.2) we implicitly assume that all manufacturing defects result in logical faults which in turn cause erroneous behavior of the chip. Certain defects may however produce no faults at all. For example, a defect in the outer area of the chip which is usually occupied by bonding pads, may be harmless to the electrical performance of the circuit. To consider logical faults instead of fabrication defects, we will as a first approximation, multiply the defect density d by the probability that at least one fault is caused by a defect. If for example, we adopt the assumption made in [16] that the number of logical faults corresponding to a single defect follows a Poisson distribution with mean c, we should then multiply d by $(1 - e^{-c})$. For convenience, we will in what follows still refer to manufacturing defects (rather than logical faults) with average density d which equals the original defect density multiplied by the probability that a defect results in a logical fault.

The Yield of a Chip with Added Redundancy

Suppose now that redundancy is added to a chip at the system level so that s_i defective elements of type *i* can be tolerated, (i.e., substituted by good spares), and denote by N_i the total number of elements (including the spares) of type *i* in a chip. Then, the chip is acceptable with any number of manufacturing defects in type *i* elements as long as all of them are restricted to at most s_i elements. The yield, which is now the probability of a chip being acceptable, is given by

$$Y = \prod_{i=1}^{n} \Pr \{\text{There are defects in at most} \\ s_i \text{ elements of type } i\}$$
(3.3)

If we denote

 $a_i^{(i)} = \Pr$ {The defects in type *i* elements are all

distributed in exactly j out of the N_i elements}

then

$$Y = \prod_{i=1}^{n} \sum_{j=0}^{S_i} a_j^{(i)}.$$
 (3.4)

We may now obtain an explicit expression for the probability $a_i^{(l)}$ in two different ways. One is to follow [8] and define

 $Q_{x,i}^{(N)} = \Pr \{x \text{ defects are distributed into exactly} \}$

j out of N elements/There are x defects}

which equals

$$Q_{x,j}^{(N)} = \sum_{k=0}^{j} (-1)^k \binom{N}{k, j-k, N-j} \binom{j-k}{N}^x$$

for $x \ge j$ and $0 < j \le N$ (3.5)

¹ The average number of defects per chip $\bar{\lambda}$ (e.g., [8]) is d*A where A is the chip's area.

where $\binom{N}{k,j-k,N-j}$ is the multinomial coefficient. For x < j we have $Q_{x,j}^{(N)} = 0$ and for j = 0

$$Q_{x,0}^{(N)} = \begin{cases} 1 & \text{if } x = 0 \\ 0 & \text{otherwise.} \end{cases}$$

This expression was derived assuming that the defects are distinguishable, i.e., Boltzmann statistics are followed [20]. If we select Bose-Einstein statistics, the defects are indistinguishable and the resulting expression is

$$Q_{x,j}^{(N)} = \frac{\binom{N}{j}}{\binom{x+N-1}{x}} \sum_{k=0}^{j-1} (-1)^k \binom{j}{k} \binom{x+j-k-1}{x}$$
for $x \ge j$ and $0 < j \le N$. (3.6)

Using the probability $Q_{x,j}^{(N)}$ we obtain the following equation for $a_j^{(l)}$

$$a_{j}^{(i)} = \sum_{x=j}^{\infty} Q_{x,j}^{(N_{i})} * \Pr \{ \text{There are } x_{x} \text{ manufacturing} \\ \text{defects in type } i \text{ elements in the chip} \}.$$
(3.7)

The last term in (3.7) is Pr $\{X_i = x_x\}$ and we may substitute it by $(3.1)^{i}$ or a similar expression for any other defect distribution.

In this first approach to the derivation of $a_j^{(i)}$ we have considered the entire chip as the basic unit of silicon in which defects occur, and then we have distributed these defects uniformly among the individual elements. In the second approach to derive an expression for $a_j^{(i)}$, we consider the single element as the basic silicon unit, out of which larger area chips are constructed. Let Y_i denote the yield (probability of zero defects) of a *single* element of type *i*, then the appropriate expression for $a_j^{(i)}$ in this case is

$$a_{j}^{(i)} = \binom{N_{i}}{j} Y_{i}^{N_{i}-j} (1-Y_{i})^{j}.$$
 (3.8)

The assumption here is that each element of type *i* may be defective with an *independent* probability $(1 - Y_i)$. This approach has been adopted, for example, in [3] and [10].

When setting the parameters for the yield of a single element Y_i , we may require that the expected value and variance of the number of defects in the total chip area will be the same as in the first approach. The expected value and variance for the generalized negative binomial distribution are Ad and $Ad(1 + Ad/\alpha)$, respectively. Therefore, both A_id and α_i should be divided by N_i . Consequently

$$Y_i = \left(1 + \frac{A_i d}{\alpha_i}\right)^{-\alpha_i / N_i}.$$
 (3.9)

Before comparing these two alternatives for calculating the probability $a_j^{(i)}$ (given by (3.7) and (3.8)), we return to the general equation for the yield, i.e., (3.4). This equation can be multiplied by a "bypass coverage probability" [11], which is the conditional probability that an element can be bypassed

(isolated) given that it is faulty. By adding this probability one may consider less than perfect procedures for locating faulty elements and reconfiguring them out of the system.

To tolerate s_i defective elements of type *i*, at least s_i redundant ones are needed. However, the exact amount of required redundancy depends upon the specific static or dynamic reconfiguration scheme used. This in turn, determines the increase in chip area which must be taken into account when calculating the yield, since a larger number of defects is expected now.

Let γ_{si} denote the increase in the area A_i (due to the addition of redundancy), needed to tolerate these s_i faulty elements of type *i*. Let γ_s denote the increase in total chip area that is required to tolerate all $\bar{s} = (s_1, s_2, \dots, s_n)$ faulty elements. The factor γ_s is called the *redundancy factor* [7] and it depends on the system topology and the reconfiguration strategy. It assumes its lowest possible value when only s_i redundant elements are included in the total of N_i elements of type $i(i = 1, 2, \dots, n)$, hence

$$\gamma_{S_i} \ge \frac{N_i}{N_i - s_i} \text{ and } \gamma_s \ge \frac{\sum_{i=1}^n A_i}{\sum_{i=1}^n \frac{A_i}{N_i} (N_i - s_i)}.$$
 (3.10)

The larger chip area results in an increased expected number of defects. We should therefore, multiply the average number of defects $A_i d_i$ in (3.1) by γ_{si} . If we insist on having the same expected value and variance of the number of defects in the total silicon area (independently of how it is partitioned into chips or elements) then, as was shown above, the clustering parameter α_i should also be multiplied by γs_i .

In addition, the increase in chip area reduces the number of chips that will fit into the same wafer. Hence, instead of calculating the yield which is the probability that a single chip is acceptable, one has to calculate the expected number of acceptable chips out of a given wafer. This expression, which we call *wafer-equivalent yield*, is obtained from (3.4) after dividing it by γ_{s} .

By comparing the wafer-equivalent yield of the faulttolerant chip and the yield of the simplex one (with no faulttolerance features), we can determine whether it is beneficial when yield is considered, to have built-in fault tolerance and how many redundant elements should we add. This comparison can be done for various system topologies and different reconfiguration algorithms.

An analysis along these lines has been done in [11] and in [7]. In both it has been observed that the improvement in yield saturates above some amount of redundancy. This indicates that there is an optimal amount of redundancy that should be added.

Still, the exact value of this optimal amount of redundancy does depend upon the expression we adopt for the waferequivalent yield of a fault-tolerant chip. To illustrate this, we compare in the following example the optimal amounts of redundancy when using the above two alternative schemes for calculating $a_{i}^{(i)}$ (defined by (3.7) and (3.8), respectively).

Example: Consider a chip with only a single type of

element that can have defects. Assume further that at least 15 such elements are required and the redundancy factor is therefore, $\gamma_s = 1 + (s/15)$. The optimal percentage of redundancy (i.e., s/15) for the two schemes was computed for different values of the clustering parameter.

As was expected, for high values of α (i.e., no clustering), both schemes call for exactly the same amount of redundancy. Here, the generalized negative bionomial distribution approaches the Poisson distribution for which the two schemes are equivalent. This was observed for $\alpha \ge 10$ and Ad = 5defects per silicon unit area. For low values of $\alpha(\alpha \le 0.01)$, adding redundancy is not recommended by any of the two schemes; the defects are severly clustered resulting in a high yield of the simplex chip.

For values of α in (0.01, 10), the second scheme calls, in most cases, for a lower optimal amount of redundancy. Moreover, this scheme might be too optimistic since it achieves a higher wafer-equivalent yield (compared to that achieved by the first scheme) for a smaller amount of redundancy. For example, for $\alpha = 0.6$ the yield of the simplex chip is for both schemes 26.2 percent. The optimal amount of redundancy according to the first scheme is 33 percent achieving a maximum wafer-equivalent yield of 47.4 percent. The optimal amount of redundancy according to the second scheme is only 20 percent but the maximum waferequivalent yield is 78.2 percent. Still, only experimental data obtained by monitoring wafers can show which scheme is "correct."

The Effect of Imperfect Production Testing

For both schemes, a chip with redundancy is considered acceptable as long as the number of defective elements of one type does not exceed the number of spares for this type. This assumes that the applied production testing procedure is perfect and we accept only chips which satisfy the above requirements. However, testing is never perfect and consequently, chips having more than the allowed number of defective elements will be declared good resulting in a higher "apparent" yield [22] [16] defined by

$$Y_{\text{apparent}} = Y + Y_{bg}$$

where Y_{bg} is the yield of bad (defective) chips that are tested as good. This yield depends on the probability that the testing procedure when applied to an element fails, given that the element is defective. We denote this probability, which is usually called *fault coverage* probability, by *fc*. The expression for $Y_{bg}(fc)$ for the case of a single type of elements that have defects is

$$Y_{bg}(fc) = \sum_{j=s+1}^{N} a_j * \Pr \{ \text{at most } s \text{ out of } j \}$$

defective elements were detected}

$$=\sum_{j=s+1}^{N}a_{j}*\sum_{r=0}^{s}\binom{j}{r}(fc)^{r}(1-fc)^{j-r}.$$
 (3.11)

To generalize the above to the case of *n* different types of

elements, it is simpler to derive a general expression for the apparent yield. Assuming, for simplicity, that the fault coverage is the same for all types of elements, we obtain

$$Y_{\text{apparent}} = \prod_{i=1}^{n} \left[\sum_{j=0}^{S_i} a_j^{(i)} + \sum_{j=s_i+1}^{N_i} a_j^{(i)} \sum_{r=0}^{s_i} {j \choose r} (fc)^r (1-fc)^{j-r} \right]. \quad (3.12)$$

IV. RELIABILITY AND PERFORMANCE OF FAULT-TOLERANT CHIPS

In the previous section we have derived expressions for the actual and apparent yield of a fault-tolerant VLSI chip. In this section we develop models for operational fault-tolerant chips in order to derive expressions for their reliability and performance availability.

Once the manufacturing process has been completed, chips having s_i or less defects in type *i* elements $(i = 1, 2, \dots, n)$ are accepted and then reconfigured to avoid the use of the defective elements. If the number of defective elements of any type *i* is less than s_i , the chip has some "residual" redundancy which can then be used for performance enhancement, i.e., handle operational faults which occur during the life time of the system. Even chips in which no redundant elements are left when leaving the manufacturing site (i.e., there were originally $\sum_{i=1}^{n} s_i$ defects in the chip), can still benefit from the fault-tolerance capability, if graceful degradation is allowed.

To evaluate the effectiveness of the "residual" redundancy and the fault-tolerance capacity of the chip, we have to select some performance measures and we need a model that will allow us to calculate these measures. A natural choice for this purpose is a Markov model.

The proposed Markov model includes a single system failure state (F). At all other states of the model, the system is operational in the presence of some faulty elements. Consider now such a general state and let f_i and u_i denote the number of faulty elements and active (used) ones of type *i* at this state, respectively. Note that u_i is defined as the number of active elements and not just fault-free ones. We have therefore, the inequality

$$u_i \le N_i - f_i \tag{4.1}$$

where N_i is the original number of elements of this type. The exact value of u_i depends (among other factors) on the restructuring strategy employed. For static restructuring we may have $u_i = N_i - f_i$, while for dynamic schemes the inequality is more likely to hold.

The value of u_i may also depend on the exact locations of these faulty elements (and faulty elements of other types as well.) However, taking this into account will turn the model intractable. Hence, we define the states of the Markov model by the f_i 's and not the u_i 's. Thus, (f_1, f_2, \cdots) will denote a general state in our approximate mode. (This single state would have to be replaced by $\prod_{i=1}^{n} {N_i \choose f_i}$ separate states if the exact position of all faulty elements is taken into account.)

The f_i faulty elements may include several (up to s_i) elements that were found to be defective when the manufacturing was completed and other that became faulty only later on.

If the number f_i becomes too high, we may reach a point at which our system can not do useful computation any longer. Let m_i denote the maximum allowed number of elements that could become faulty if s_i ones were already defective at t = 0. Therefore,

$$f_i \le s_i + m_i < N_i. \tag{4.2}$$

This inequality means that if less than s_i elements were defective at t = 0, the system will endure the failure of more than m_i elements at t > 0.

An example of the suggested Markov model for a chip with two types of elements that can fail, is depicted in Fig. 1. In this model (f_1, f_2) is a state at which the system is operational in the presence of f_1 and f_2 faulty elements of type 1 and 2, respectively. A transition from state (f_1, f_2) to the system failure state (F) takes place when an additional element becomes faulty and the system fails to recover from its effect. The corresponding transition rate is denoted by $\alpha_{f_1f_2}^F$. Similary, $\alpha_{f_1,f_2}^{f_1+1,f_2}$ is the transition rate from state (f_1, f_2) to state $(f_1 + 1, f_2)$.

These transition rates depend on the failure rates of the system elements and on the probability of a successful recovery (by reconfiguration) of the system following an element failure. Let λ_i denote the failure rate of operational elements of type *i* (adopting the common assumption that the time to failure is exponentially distributed), and let p_i denote the probability of a successful recovery, i.e.,

 $p_i = \Pr$ {The system recovers (reconfigures) successfully/

A fault in type *i* element has occurred}

The expression for the transition rate from (f_1, f_2) to $(f_1 + 1, f_2)$ depends on the restructuring strategy employed. For example, if upon a failure all the remaining fault-free elements of both types are used (active) then

$$\alpha_{f_1,f_2}^{f_1+1,f_2} = (N_1 - f_1) * \lambda_1 * p_1$$

If however, the number of active elements of type 1 satisfies $u_1 < N_1 - f_1$, then the appropriate expression is

$$\alpha_{f_1,f_2}^{f_1+1,f_2} = u_1 * \lambda_1 * p_1 + (N_1 - f_1 - u_1) * \lambda_1.$$

This is based on the assumption that upon a failure of a nonactive element, the system will recover successfully with probability 1. The above expression is not always well-defined since in the general case (for complex system topologies and restructuring schemes), u_1 may be a function of f_1 and f_2 , and may depend on the exact positions of these faulty elements as well. Therefore, the value for u_1 to be used in the above expression for the transition rate, must be obtained according to some empirical rule. Several such rules can be envisioned, for example, the average over all possible positions of the faulty elements, or the worst case one, or the most probable one.

There are cases in which u_i depends only on the value of $f_i(i = 1, 2, \dots, n)$. In these cases, not only the above expression for the transition rate is accurate but the entire model can be



Fig. 1. A Markov model for a VLSI chip with defects and operational faults.

simplified by partitioning the Markov chain into n independent chains. Each will then be solved separately and the final results will be combined to obtain the required performance measures. This case is demonstrated in the next section.

State (0, 0) of the Markov model in Fig. 1 is the initial state of the system if no defects occurred while the chip was manufactured. If there were k_1 and k_2 defective elements of type 1 and 2, respectively, ($0 \le k_i \le s_i$, i = 1, 2) then (k_1, k_2) will be the initial state. The probability of this event is

$$a_{k_1}^{(1)} * a_{k_2}^{(2)} \quad (0 \le k_i \le s_i, \ i = 1, \ 2)$$
 (4.3)

since the probabilities of defects in different types of elements are independent [20]. We have, therefore, $(s_1 + 1) * (s_2 + 1)$ possible initial states if only chips having at most s_1 and s_2 defective elements of type 1 and 2, respectively, are accepted. If we sum the probabilities of all these initial states we obtain

$$\sum_{k_1=0}^{s_1} \sum_{k_2=0}^{s_2} a_{k_1}^{(1)} * a_{k_2}^{(2)} = \left[\sum_{k_1=0}^{s_1} a_{k_1}^{(1)} \right] * \left[\sum_{k_2=0}^{s_2} a_{k_2}^{(2)} \right] \quad (4.4)$$

which equal the yield Y as defined in (3.4).

However, since the chip screening procedure is imperfect, chips with larger numbers of defective elements may be accepted. Consequently, all the states in the Markov model are possible initial states. If for example, $k_1 > s_1$ and $k_2 \le s_2$, the probability of (k_1, k_2) being an initial state is then

$$\left[a_{k_1}^{(1)}\sum_{r=0}^{s_1}\binom{k_1}{r}(fc)^r(1-fc)^{k_1-r}\right]*a_{k_2}^{(2)}.$$
 (4.5)

The expressions for the probabilities of the other two kinds of initial states may be derived similarly. Summing the probability of each state being an initial state results in the apparent yield as given by (3.12).

A state like $(s_1 + m_1, f_2)$ in Fig. 1 is a terminal state with respect to failures in type 1 elements, i.e., a state from which, upon an additional failure of a type 1 element, the only transition possible is to the system failure state (F). Recall that m_1 is the largest number of faulty elements of type 1 that the system can tolerate if no redundant elements were left when the system went into operation. Similarly, all states $(f_1, s_2 +$ m_2) are terminal states with respect to failures in type 2 elements.

We define now the state probabilities of our model

$$P_{f_1,f_2}^{(k_1,k_2)}(t) = \Pr$$
 {The system is in state (f_1, f_2) at time $t/$
The system was initially in state (k_1, k_2) }

for $f_1 \ge k_1$ and $f_2 \ge k_2$) with $P_{k_1,k_2}^{(k_1,k_2)}(0) = 1$ and $P_{f_1,f_2}^{(k_1,k_2)}(0) = 0$ for $f_1 > k_1$ or $f_2 > k_2$.

The Markov model in Fig. 1 is described then by the following differential equations.

$$\frac{dP_{k_1,k_2}^{(k_1,k_2)}(t)}{dt} = -\alpha_{k_1,k_2}P_{k_1,k_2}^{(k_1,k_2)}(t)$$
(4.6)

$$\frac{dP_{f_1,f_2}^{(k_1,k_2)}(t)}{dt} = -\alpha_{f_1,f_2} P_{f_1,f_2}^{(k_1,k_2)}(t) + \alpha_{f_1-1,f_2}^{f_1,f_2} P_{f_1-1,f_2}^{(k_1,k_2)}(t) + \alpha_{f_1,f_2-1}^{f_1,f_2} P_{f_1,f_2-1}^{(k_1,k_2)}(t)$$
(4.7)

where

$$\alpha_{f_1, f_2} = \alpha_{f_1, f_2}^{f_1 + 1, f_2} + \alpha_{f_1, f_2}^{f_1, f_2 + 1} + \alpha_{f_1, f_2}^F.$$
(4.8)

We denote by v_b any vertex (state) at level b in Fig. 1, i.e., any state (i, j) satisfying i + j = b. Thus, the sequence (k_1, j) k_2 , $v_{k_1+k_2+1}$, \cdots , v_{i+j-1} , (i, j) corresponds to a path in Fig. 1 from the initial state (k_1, k_2) to the state (i, j).

Using this notation, the solution of (4.6) and (4.7) under the condition

$$\alpha_{i,i} \neq \alpha_{x,y}$$
 for all $(x, y) \neq (i, j)$

which is usually satisfied, is

$$P_{f_{1},f_{2}}^{(k_{1},k_{2})}(t) = \sum_{\substack{A \text{ path } (k_{1},k_{2}),\cdots,(f_{1},f_{2}) \\ b = k_{1}+k_{2}}} \alpha_{k_{1},k_{2}}^{\nu_{k_{1}+k_{2}+1}} \alpha_{\nu_{k_{1}+k_{2}+1}}^{\nu_{k_{1}+k_{2}+2}} \cdots \alpha_{\nu_{f_{1}+f_{2}-1}}^{f_{1},f_{2}} \\ \cdot \sum_{\substack{b=k_{1}+k_{2} \\ c = k_{1}+k_{2}}}^{f_{1}+f_{2}} \frac{e^{-\alpha_{v_{b}}t}}{\prod_{\substack{c=k_{1}+k_{2} \\ c \neq b}}}$$
(4.9)
nd

a

$$P_{k_1,k_2}^{(k_1,k_2)}(t) = e^{-\alpha_{k_1,k_2}t}$$
(4.10)

where $v_{k_1+k_2}$ and $v_{f_1+f_2}$ in (4.9) specify the states (k_1, k_2) and (f_1, f_2) , respectively.

The summation in (4.9) is over all paths (total of $\binom{f_1-k_1+f_2-k_2}{f_1-k_1}$ paths, each consisting of $(f_1+f_2) - (k_1+k_2)$ + 1 nodes) in the Markov model (Figure 1) from state (k_1, k_2) to (f_1, f_2) .

For such a Markov model we can calculate several performance measures like Reliability, Performability, Computational availability, and Area utilization [7]. Let $R_{k_1,k_2}(t)$ denote the reliability of a system (i.e., the probability that it operates correctly in the time interval [o, t], which had k_1 and k_2 defective elements of type 1 and 2, respectively, at t = 0. This reliability can be calculated from the above Markov model as follows

$$R_{k_1,k_2}(t) = \sum_{f_1=k_1}^{s_1+m_1} \sum_{f_2=k_2}^{s_2+m_2} P_{f_1,f_2}^{(k_1,k_2)}(t).$$
(4.11)

We may then define and compute

$$R(t) = \frac{\sum_{k_1=0}^{s_1+m_1} \sum_{k_2=0}^{s_2+m_2} \Pr\{(k_1, k_2) \text{ is the initial state}\} * R_{k_1, k_2}(t)}{\sum_{k_1=0}^{s_1+m_1} \sum_{k_2=0}^{s_2+m_2} \Pr\{(k_1, k_2) \text{ is the initial state}\}}$$
(4.12)

as the average reliability of a system having s_i (i = 1, 2) or less defects when manufactured.

The probability of (k_1, k_2) being an initial state is obtained either from (4.3) or from (4.5) or a similar appropriate equation. The denominator in (4.12) (which is a normalization factor so that the probabilities sum to 1) is the apparent yield (3.12). If the fault coverage fc is nearly 1, then equation (4.12) reduces to

$$R(t) = \frac{1}{Y} \sum_{k_1=0}^{s_1} \sum_{k_2=0}^{s_2} a_{k_1}^{(1)} * a_{k_2}^{(2)} * R_{k_1,k_2}(t)$$
(4.13)

The average reliability is in general a function of the amount of added redundancy. When searching for the optimal amount of redundancy to be incorporated into the chip, we should not ignore the cost associated with redundancy, i.e., a larger chip area resulting in a smaller number of chips fabricated out of a given size wafer. We may therefore, attempt to optimize the expected number of chips, in a given wafer, that are operating correctly in the time interval [0, t]. This equals the reliability of a single chip times the expected number of chips per wafer. The number of chips decreases as redundancy is introduced, by the factor of γ_{s} . Consequently, we may search for the values of s_1 and s_2 that optimize the following cost function, which we call wafer-level reliability

WR
$$(t) = \frac{R(t)}{\gamma_s}$$
. (4.14)

This will allow us to determine whether it is beneficial when reliability is considered, to introduce redundancy into the architecture of the system and how many redundant elements of each type we should include.

Example: The wafer-level reliability as a function of the



Fig. 2. The wafer-equivalent yield and wafer-level reliability of a fault-tolerant chip.

added redundancy has been calculated for the following example. A chip has one type of elements that can have defects or operational faults. Fifteen such elements are required but the chip can be used if at least 12 are operational (i.e., m =3). The model parameters chosen are $\alpha = 0.743$, Ad = 1.93[16] and p = 0.9. The wafer-equivalent yield of this chip is maximized when four redundant elements are added to the simplex chip. The improvement in wafer-level reliability also saturates above some amount of redundancy. The optimal amount of redundancy that maximizes the wafer-level reliability depends on the mission time of the system. Fig. 2 depicts the wafer-level reliability and wafer-equivalent yield as functions of the number of spare elements with the mission time as parameter. For a low value of t (time is measured in 1/ λ units), the optimal amount of redundancy is $s_{opt.} = 0$. For t = $0.25/\lambda$, $s_{opt.}$ = 3, and for $t = 0.35/\lambda$, $s_{opt.}$ = 6. The tradeoff between yield enhancement and reliability improvement depends therefore, on the mission time. Graphs like the one shown in Fig. 2, can be used to determine the final amount of redundancy to be added when both yield and reliability are considered.

Reliability is considered in many cases an insufficient measure for the performance of a system and hence, other measures have been proposed and used. Many such performance measures can be computed using the same Markov model. For example, the computational availability $A_c^{(k_1,k_2)}(t)$ which is the expected available computational capacity defined by

$$A_{c}^{(k_{1},k_{2})}(t) = \sum_{f_{1}=k_{1}}^{s_{1}+m_{1}} \sum_{f_{2}=k_{2}}^{s_{2}+m_{2}} c_{f_{1},f_{2}} P_{f_{1},f_{2}}^{(k_{1},k_{2})}(t)$$
(4.15)

where c_{f_1,f_2} is the computational capacity of the system in state (f_1, f_2) [7], expressed for example in instructions per time unit. The computational capacity depends on the number of

active elements of both types in state (f_1, f_2) . An expression for the computational capacity of an example system is developed in the next section.

Here too, when searching for the optimal amount of redundancy, we should employ a cost function which considers the additional silicon area needed when fault tolerance is introduced into the chip. Such a cost function, called area utilization in [7], is

$$U_{k_1,k_2}(t) = \frac{A_c^{(k_1,k_2)}(t)}{\gamma_s} .$$
 (4.16)

Other performance measures, like mean time to failure, can also be calculated. For example, let T_{k_1,k_2} denote the mean time to failure of a system which was initially in state (k_1, k_2) , then

$$T_{k_1,k_2} = \int_0^\infty R_{k_1,k_2}(t) \ dt. \tag{4.17}$$

The average mean time to failure can be defined similarly to (4.12).

A Model for a System with Element-Level Redundancy

The Markov model depicted in Fig. 1 describes a system which has redundancy only at the system level. Here, a single defect or fault in an element requires the switching out (isolation) of the failing element and the replacement of it with a spare one (if one exists). We may add redundancy at the element level which will allow us to use an element even if it has one defect or failure. We assume that the second failure is fatal meaning that an element having two failures will have to be switched out. An example might be a multiprocessor in which each node consists of a dual processor. If one of the two processors fails, we may still use this node (with possibly a lower computational capacity) as long as the second processor is operating correctly.

A Markov model suitable for such a system with a single type of elements (e.g., processors) that can fail is shown in Fig. 3. Here, (f, \hat{f}) is a state at which the system is operational in the presence of f faulty nodes and \hat{f} partially faulty nodes (i.e., nodes having a single faulty processor). As before, s + sm is the maximum number of defective and faulty nodes that the system can tolerate. Consequently, the state (s + m, N - m)s - m) is a terminal state. The initial state of the Markov model is determined by the number and distribution of the manufacturing defects. If s is the maximum number of defective nodes (partially or completely) that we are willing to tolerate and still accept the VLSI chip, then all the states (j, \hat{j}) ; $j = 0, 1, \dots, s; \hat{j} = 0, 1, \dots, N - j$ are possible initial states. Let $a_{j,\hat{j}}$ be the probability of (j, \hat{j}) being the initial state, then this is the probability of having 2j + j defects out of which exactly 2*j* appear in pairs, hence

$$a_{j,j} = \frac{\binom{N}{j}\binom{N-j}{\hat{j}}2^{j}}{\binom{2N}{2j+\hat{j}}}a_{2j+j} \qquad (4.18)$$

where a_{2i+j} is given by (3.7) with N being replaced by 2N.



Fig. 3. A Markov model for a VLSI chip with element-level and systemlevel redundancy.

The yield of the chip is

$$Y = \sum_{j=0}^{s} \sum_{j=0}^{N-j} a_{j,j}.$$
 (4.19)

An expression for the apparent yield, similar to (3.13), can also be derived.

If we insist on having at least N - s perfectly operating nodes, then s is the total number of completely or partially faulty nodes that we allow, and the yield is given in this case by

$$Y = \sum_{j=0}^{s} \sum_{j=0}^{s-j} a_{j,j}.$$

The state probabilities for the Markov model shown in Fig. 3 are defined as follows:

 $P_{f,\hat{f}}^{(k,\hat{k})}(t) = \Pr \{ \text{The system is in state } (f, \hat{f}) \text{ at time } t / The system was initially in state } (k, \hat{k}) \}$

 $k=0, 1, \cdots, s; \hat{k}=0, 1, \cdots, N-s;$

and $(f, \hat{f}) \ge (k, \hat{k})$ lexicographically.

with $P_{k,\hat{k}}^{(k,\hat{k})}(0) = 1$ and $P_{f,\hat{f}}^{(k,\hat{k})}(0) = 0$ for $(f, \hat{f}) > (k, \hat{k})$ lexicographically.

The above state probabilities satisfy the following differential equations:

$$\frac{dP_{k,\hat{k}}^{(k,\hat{k})}(t)}{dt} = -\alpha_{k,\hat{k}}P_{k,\hat{k}}^{(k,\hat{k})}(t)$$
(4.20)

$$\frac{dP_{f,f}^{(k,k)}(t)}{dt} = -\alpha_{f,f}P_{f,f}^{(k,k)}(t) + \alpha_{f-1,f+1}^{f,f}P_{f-1,f+1}^{(k,k)}(t) + \alpha_{f,f-1}^{f,f}P_{f,f-1}^{(k,k)}(t)$$
(4.21)

where

$$\alpha_{f,\hat{f}} = \alpha_{f,\hat{f}}^{f+1,\hat{f}-1} + \alpha_{f,\hat{f}}^{f,\hat{f}+1} + \alpha_{f,\hat{f}}^{F}.$$
(4.22)

We denote by v_b any state (f, \hat{f}) satisfying $2f + \hat{f} = b$. For example, v_5 is any one of the states (2, 1) and (1, 3) in Fig. 3. Thus, the sequence $(k, \hat{k}), v_{2k+\hat{k}+1}, v_{2k+\hat{k}2}, \cdots, v_{2f+\hat{f}-1}, (f, \hat{f})$ corresponds to a path in Fig. 3 from the initial state (k, \hat{k}) to the state (f, \hat{f}) .

Using this notation, the solution of (4.20) and (4.21) under the condition

$$\alpha_{f,\hat{f}} \neq \alpha_{x,\hat{x}}$$
 for all $(x, \hat{x}) \neq (f, \hat{f})$

which is usually satisfied, is

d O

$$P_{f,f}^{(\kappa,\kappa)}(t) = \sum_{A \text{ path } (k,\bar{k}),\cdots,(f,\bar{f})} \alpha_{k,\bar{k}}^{\nu_{2k}+\bar{k}+1} \alpha_{\nu_{2k}+\bar{k}+1}^{\nu_{2k}+\bar{k}+2} \cdots \alpha_{\nu_{2f+\bar{f}-1}}^{f,\bar{f}} \\ \cdot \sum_{b=2k+\bar{k}}^{2f+\bar{f}} \frac{e^{-\alpha_{v}b^{t}}}{\prod_{\substack{c=2k+\bar{k}\\c\neq b}}^{2f+\bar{f}}} (\alpha_{v_{c}}-\alpha_{v_{b}})$$
(4.23)

and

$$P_{k,k}^{(k,k)}(t) = e^{-\alpha_{k,k}t}$$
(4.24)

where $v_{2k+\hat{k}}$ and $v_{2f+\hat{f}}$ in (4.23) specify the states (k, \hat{k}) and (f, \hat{f}) , respectively.

The summation in (4.23) is over all paths (total of $\binom{2f-2k+\hat{f}-\hat{k}}{f-k}$ paths, each consisting of $(2f+\hat{f}) - (2k+\hat{k}) + 1$ nodes) in the Markov model (Fig. 3) from state (k, \hat{k}) to the state (f, \hat{f}) .

Using the state probabilities given by (4.23), one can derive expressions for various performance measures and determine the optimal amount of redundancy, as was done for the previous Markov model.

V. A MULTIPLE BUS MULTIPROCESSOR SYSTEM

As an example for a VLSI chip consisting of several types of system elements with some redundant ones of each type, consider a multiple bus multiprocessor system designed on a large area chip (or even a wafer). Let P, M, and B denote the number of homogeneous processors, memory modules and interconnecting busses, respectively, that are operational (fault-free). Such a system is depicted in Fig. 4 where each global bus can connect any processor to any memory module. Following the notation used in [12] and [21], we refer to this multiprocessor as a P * M * B system.

There are several ways to characterize the behavior of a P * M * B system. Our purpose in this section is only to illustrate



the application of the model presented in the previous section. We adopt therefore, a relatively simple characterization of the system which is based on two parameters. One is the time between two consecutive memory references and the second is the connection time between processor and memory in a single memory access. Both parameters are random variables and are assumed to be exponentially distributed with mean $1/\delta$ (processing time) and $1/\mu$ (connection time).

We wish to find an expression for the computational capacity of a P*M*B system denoted by $c_{P,M,B}$. An expression that will allow us to determine the optimal number of spare processors, spare memory modules and spare interconnection busses to be designed in the VLSI chip so that yield and/or performance are maximized.

We may define the computational capacity of the P * M * Bsystem as the expected number of active processors, i.e., processors which are executing their task and not idle while waiting to access a common memory module. This performance index is known as processing power. Other performance indexes like the average cycle time and the instruction execution rate can be simply derived from the processing power index [12].

To calculate the processing power index we may construct a queuing network model. The computational complexity of this model increases very rapidly with system size. Fortunately, as has been shown in [12] and [21], approximate models with reasonably small errors in the final results, can be employed. These are derived by lumping "equivalent" states of the model to obtain a Markov chain of substantially smaller size.

An example of such a model is shown in Fig. 5. In it, at state (P - i) there are P - i processors which are executing their tasks while the remaining *i* processors are idle being serviced or waiting to be serviced by a memory module.

At a rate of $\beta(i) * \mu$, one of the *i* idle processors will complete its service increasing the number of active ones to P - i + 1. $\beta(i)$ is the average number of processors, out of the *i* idle ones, that are serviced at a given time instant. Similarly, at a rate of $(P - i)\delta$, an active processor will generate a memory request and join the idle processors, reducing the number of active ones by 1.

To derive an expression for $\beta(i)$, we assume (as in [12] and [21]) that processors request service from the different memory modules with equal probabilities. Hence, the probability that all *i* requests of the idle processors will be directed



multiprocessor. to exactly j out of the M memory modules is $Q_{i,i}^{(M)}$ which is

to exactly j out of the M memory modules is $Q_{i,j}^{(m)}$ which is defined in (3.5). This probability has to be multiplied by min (j, B) since only B requests can be serviced if j is greater than the number of busses B. Finally we obtain

$$\beta(i) = \sum_{j=1}^{\min(i,M)} Q_{i,j}^{(M)} \min(j, B) \quad i = 1, 2, \dots, P. \quad (5,1)$$

The Markov chain in Fig. 5 is a birth and death one, whose solution is easily obtained. Let $\Phi(i)$ denote the steady-state probability of state *i*, then

$$\Phi(i) = \frac{\left(\frac{\mu}{\delta}\right)^{i} \frac{\prod_{k=0}^{i-1} \beta(P-k)}{i!}}{1 + \sum_{m=1}^{P} \left(\frac{\mu}{\delta}\right)^{m} \frac{\prod_{k=0}^{m-1} \beta(P-k)}{m!}} (i=1, 2, \cdots, P)$$
(5.2)

$$(0) = \frac{1}{1 + \sum_{m=1}^{P} \left(\frac{\mu}{\delta}\right)^m \frac{\prod_{k=0}^{m-1} \beta(P-k)}{m!}}$$

The processing power measure is given by

Φ

$$c_{P,M,B} = \sum_{i=1}^{P} i\Phi(i).$$
 (5.3)

Expression (5.3) can now by substituted into (4.15) to calculate the computational availability which is defined now as the expected available processing power. Equation (4.16) may then be used to compute the area utilization of the multiprocessor VLSI system.

To calculate the computational availability we also need the state probabilities of a Markov model similar to the one shown in Fig. 1, with three types of elements, namely processors, memory modules, and busses. Fortunately, in the multiple bus multiprocessor system, the number of active elements of any type depends only on the number of faulty elements of this type and is independent of faulty elements of the other two types. Consequently, the Markov model (like the one in Fig. 1) may be partitioned into three independent ones, each solved separately.



Fig. 6. The area utilization of the P * M * B multiprocessor as a function of the number of spare elements of different types.

The state probability which is needed for the calculation of the computational availability (and the system reliability as well), is equal to the product of the state probabilities obtained separately from the three simpler Markov models for the processors, memory modules, and busses.

This calculation was done for a multiple bus multiprocessor system with the following parameters:

1) Eight processors with $\alpha = 0.3$, Ad = 1.5, p = 0.9, and $\lambda = \lambda_0$ (time will be measured in $1/\lambda_0$ units). In addition, m = 2, and $\delta/\mu = 0.91$.

2) Eight memory modules with $\alpha = 0.2$, Ad = 1.2, p = 0.92, $\lambda = \lambda_0$, and m = 1.

3) Four buses with $\alpha = 0.12$, Ad = 0.9, p = 0.95, $\lambda = 0.75\lambda_0$, and m = 1.

For these system parameters three sets of values for the number of spare processors, spare memory modules and spare busses were obtained. For maximum wafer-equivalent yield 4, 3, and 2 spare processors, memories, and busses, respectively, are required. In this calculation we have assumed that fc = 1 and therefore, the apparent yield equals the actual one. For maximum wafer-level reliability (1, 3, 1) spares (processors, memories and busses, respectively) are required for a mission time of $t = 0.2 * 1/\lambda_0$. For the same mission time, the maximum area utilization (wafer-level processing power availability), is achieved for (2, 3, 1) spares (processors, memories, and busses, respectively).

A useful application for this model might be the analysis of the relative importance of spares for the three types of system elements. To perform this kind of analysis we can, for example, set the number of spare memory modules and spare busses at some fixed values (e.g., their optimal values for a desired mission time) and then, observe the dependency of the area utilization on the number of spare processors. Such an analysis has been done for the above system parameters, and the results are illustrated in Fig. 6.

In this figure, the area utilization is shown as a function of the number of spares (spare processors or spare memories or spare busses). The notation (-, 3, 1) means that the numbers

of spare memories and busses were fixed at 3 and 1, respectively, and the different values for the number of spare processors appear on the horizontal axis.

One conclusion that might be drawn from this figure is that the area utilization measure is more sensitive to the number of spare memory modules, than it is to the other two types of elements.

Another interesting phenomenon was observed while performing this analysis. The optimal number of spares of any type in Fig. 1, is independent of the numbers of spares of the other two types of elements. For example, the curves (2, -, 1) and (0, -, 0) have their maximum at exactly the same value of 3 spare memories. The same phenomenon was observed for a mission time of $t = 0.3 1/\lambda_0$. Here, the optimal values of spare numbers are: (4, 3, 2) for maximum wafer-equivalent yield (as before), (3, 5, 2) for maximum wafer-level reliability, and (4,5 2) for maximum area utilization. However, for a longer mission time (e.g., $t \ge 0.4 1/\lambda_0$) where the optimal values of the spare numbers are higher, the above independence is not preserved.

VI. CONCLUSIONS

VLSI and WSI architectures that use redundancy for yield and performance improvement have been considered. The available redundancy on the chip or wafer is primarily limited by the size of the chip or wafer; hence, it is imperative to find a method by which one can optimally share the available redundancy between yield enhancement and performance improvement.

We have developed in this paper analytical models for the evaluation of performance and yield improvement through redundancy. The models proposed can be used to study the effect of sharing element level and system level redundancy, between these two somewhat competing requirements.

ACKNOWLEDGMENT

The authors wish to thank D. Towsley and Z. Koren for helpful discussions and to thank Z. Koren for developing the program to calculate the optimal amount of redundancy for maximum yield and performance.

References

- R. P. Cenker, et al., "A fault-tolerant 64K dynamic random-access memory," *IEEE Trans. Electron. Dev.*, vol. ED-26, pp. 853-860, June 1979.
- [2] D. S. Fussel and P. J. Varman, "Fault-tolerant wafer-scale architectures for VLSI," in Proc. 9th Annu. Symp. Comput. Arch., May 1982.
- [3] J. W. Greene and A. El Gamal, "Configuration of VLSI arrays in the presence of defects," J. Ass. Comput. Mach., vol. 31, pp. 694–717, Oct. 1984.
- [4] D. Gordon, I. Koren, and G. M. Silberman, "Restructuring hexagonal arrays of processors in the presence of faults," to appear in J. VLSI Comput. Syst.
- [5] K. Hedlund and L. Snyder, "Wafer-scale integration of configurable, highly parallel (chip) processor," in *Proc. 1982 Int. Conf. Parallel Processing*, Aug. 1982, pp. 262–265.
- [6] I. Koren, "A reconfigurable and fault-tolerant VLSI multiprocessor array," in Proc. 8th Ann. Symp. Comput. Architect., May 1981, pp. 425-441.
- [7] I. Koren and M. A. Breuer, "On area and yield considerations for

fault- tolerant VLSI processor arrays," *IEEE Trans. Comput.*, vol. C-33, pp. 21-27, Jan. 1984.

- [8] I. Koren and D. K. Pradhan, "Introducing redundancy into VLSI designs for yield and performance enhancement," in *Proc. 15th Ann. Symp. Fault-Tolerant Comput.*, June 1985, pp. 330–335.
- [9] H. T. Kung and M. S. Lam, "Fault-tolerance and two-level pipelining in VLSI systolic arrays," in *Proc. of 1984 Conf. Advanced Res. VLSI*, M.I.T., Jan. 1984, pp. 74-83.
- [10] F. T. Leighton and C. E. Leiserson, "Wafer-scale integration of systolic arrays," *IEEE Trans. Comput.*, vol. C-34, pp. 448-461, May 1985.
- [11] T. E. Mangir and A. Avizienis, "Fault-tolerant design for VLSI: Effects of interconnect requirements on yield improvements of VLSI designs," *IEEE Trans. Comput.*, vol. C-31, pp. 609-615, July 1982.
- [12] M. Ajmone Marsan and M. Gerla, "Markov models for multiple bus multiprocessor systems," *IEEE Trans. Comput.*, vol. C-31, pp. 239– 248, Mar. 1982.
- [13] H. Mizrahi and I. Koren, "Evaluating the cost-effectiveness of switches in processor array architectures," in *Proc. 1985 Int. Conf. Parallel Processing*, Aug. 1985.
- [14] D. K. Pradhan, "Fault-tolerant architectures for multiprocessors and VLSI systems," in *Proc. 13th Symp. Fault-Tolerant Comput.*, June 83.
- [15] A. L. Rosenberg, "The diogenes approach to testable fault-tolerant arrays of processors," *IEEE Trans. Comput.*, vol. C-32, pp. 902– 910, Oct. 1983.
- [16] S. C. Seth and V. D. Agrawal, "Characterizing the LSI yield equation from wafer test data," *IEEE Trans. Comput.-Aided Des.*, vol. CAD-3, pp. 123–126, Apr. 1984.
- [17] C. H. Sequin and R. M. Fujimoto, "X-tree and Y-component," in VLSI Architectures, R. Randell and P. C. Treleaven, Eds., Eglewood Cliffs, NJ: Prentice-Hall, 1983, pp. 299-326.
- [18] L. Snyder, "Introduction to the configurable highly parallel computer," Computer, vol. 15, pp. 47-56, Jan. 1982.
- [19] C. H. Stapper, A. N. McLaren, and M. Dreckman, "Yield model for productivity optimization of VLSI memory chips with redundancy and partially good product," *IBM J. Res. Develop.*, vol. 24, no. 3, pp. 398-409, May 1980.
- [20] C. H. Stapper, F. M. Armstrong, and K. Saji, "Integrated circuit yield statistics," Proc. IEEE, vol. 71, pp. 453-470, Apr. 1983.
- [21] D. Towsley, "Approximate models of multiple bus multiprocessor systems," *IEEE Trans. Comput.*, vol. C-35, pp. 220–228, Mar. 1986.
- [22] R. L. Wadsack, "Fault coverage in digital integrated circuits," Bell Syst. Tech. J., vol. 57, pp. 1475–1488, May 1978.

reactions connection for a cover of a Boolean function (1, 1)is a cover for function F. A prime cover E for e is called equivalent of C if for any implecting F in C, there exists is pinterested of C if for any implecting F in C, there exists is pinterested if e is containing f. A minimum exclusion is a neared of minimum exclusion F are provided in the second size of a contain restricted if E accounting to the definition E. Extended provide the second of E are gravited as dependent of a context and restricted in the generate x information exclusion.

Let us introduce à fiew consensus operation. Let it and it he const water

A REAL AND A REAL AND AN A REAL AND A REAL A

BUT DOT A TO ASSASSADD OF BRANKE STA

Letting $I = \{i, j, k\}$ and I_j are disjoint in more than one variable, then, around $\{i_{1}, j_{2}\}$ and I_j and I_j modulater in extra by any variable i_{2} then account $i_{2} \in J = \{i_{1}, j_{2}, j_{2}, j_{3}\}$, where $i_{1} \in J_{2}$, $i_{2} \in J_{3}$, $i_{3} \in J_{3}$.



Israel Koren (S'72-M'76) received the B.Sc. (cumm laude), M.Sc., and D.Sc. degrees from the Technion—Israel Institute of Technology, Haifa, in 1967, 1970, and 1975, respectively, all in electrical engineering.

Since 1979 he has been with the Departments of Electrical Engineering and Computer Science at the Technion—Israel Institute of Technology, Haifa, Israel where he became the Head of the VLSI Systems Research Center in 1985. Previously he has held positions with the University of California,

Santa Barbara and the University of Southern California, Los Angeles. In 1982 he was on sabbatical leave with the University of California, Berkeley. He has been a consultant to National Semiconductor, Israel, in the areas of architecture of microprocessors and high-speed algorithms for arithmetic operations, in 1984–1986, to Tolerant Systems, San Jose, CA in architecture of fault-tolerant distributed computer systems in 1983, and to ELTA, Electronics Industries, Israel, in the area of architecture of parallel signal processors in 1981–1982. Currently he is a Visiting Professor at the University of Massachusetts, Amherst. His current research interests are fault-tolerant VLSI architectures and the reliability evaluation of computer systems.

Dhiraj K. Pradhan (S'70-M'72-SM'80) received the Ph.D. degree in 1972.

He is currently a Professor in the Department of Electrical and Computer Engineering, University of Massachusetts, Amherst. Previously, he has held positions with the University of Regina, Regina, Sask., Canada, Oakland University, Rochester, MI, Stanford University, Stanford, CA, and IBM Corporation. He has been actively involved with research in fault-tolerant computing and parallel processing since 1972. He has presented several

papers on fault-tolerant computing and parallel processing. He has also published extensively in journals such as IEEE TRANSACTIONS and *Networks*. His research interests include fault-tolerant computing, computer architecture, graph theory, and flow networks.

Dr. Pradhan has edited the Special Issues on Fault-Tolerant Computing, in IEEE COMPUTER (March 1980), and the April 1986 issue of this TRANSAC-TIONS and served as Session Chairman and Program Committee member for various conferences. He is an Editor for the *Journal of VLSI and Digital Systems*. He is also the Editor of a book entitled *Fault-Tolerant Computing: Theory and Techniques* (Prentice-Hall).

processing several and explanation implicants and prime implicants and thus can guareness a firmer cover for the Backenn flucture. The distoction of essential primes are prime cover. In this corresponter and produces has guaranted a prime cover. In this corresponcemers, however, we shall argue that it is advantageous to generate essential ordered before a prime cover is found. A method is presented when a can generate all explicit order is found. A method is presented cover. I computed with the tenermical biotechild for essential primes approach, the essential primes without generating a prime futures approach, the essential primes then Genavity approach addition computation for guaranteed generating infines but approach addition computation for guaranteed generating infines that its addition computation for guaranteed generating infines the instaally can apped up the tareaction guaranteed generating infines that its and the instaally can apped up the tareaction guaranteed generating in the start of the start apped on the tareaction guaranteed generating in the start of the start of

We care generate all especial primes by her generating a set of prime indicator called the private cosmical primes. Then we are not back to divert whather each particular cosmical prime are essent in innone. A necessary and addressed condition in detering associated in a set for a bootean function with binary impact has been been and some for a bootean function with binary impact has been gaves and bootean for detecting costant primes [6]. (Fee and bittles, we shall gove a measure set and sufficient conditions for buttles, we shall gove a measure set and sufficient condition for bittles, we shall gove a measure set of author of the set bittles. We shall gove a measure set of author of the set bittles are available on the manipus valued case. This semicing has been readed as appeared to be manipus valued case. This semicing

Museriet exercise biovendan 18, 1985, nevied blav 13, 1986 The surrow is with the instant of factorization Science, Academia Sidica associatives. Toberti

ALL REAL PRIMARY SALES