

INTRODUCING REDUNDANCY INTO VLSI DESIGNS FOR YIELD AND PERFORMANCE ENHANCEMENT⁺

Israel Koren

Dept. of Electrical Engineering
Technion - Israel Institute of Technology
Haifa 32000, Israel

Dhiraj K. Pradhan

Dept. of Electrical and Computer Engineering
University of Massachusetts
Amherst, MA 01003

ABSTRACT

New challenges have been brought to fault-tolerant computing research because of developments in IC technology. One emergent area in VLSI designs is development of architectures, built by interconnecting a large number of a few types of elements on a single chip or wafer. Two important topics, related to such VLSI designs, are the focus of this paper: they are yield enhancement and performance improvement.

In this paper we present analytical models that evaluate how yield enhancement and performance improvement may both be achieved by introducing redundancy into these VLSI designs.

Index Terms - Yield enhancement, performance improvement, reliability, computational availability, VLSI designs, fault-tolerance, redundancy.

1. INTRODUCTION

Two VLSI-based areas in which important innovations are likely to occur are in the wafer-scale integrated architectures, and in the single-chip/multi-processing element 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. In addition, internal connections between chips on the same wafer are more reliable and have a smaller propagation delay than external ones. The latter does make it possible to build a high-speed processor on a single chip, by interconnecting a large number of simple processing elements (PEs).

Much recent research has focused on these new architectural innovations, especially those created by interconnecting a large number of elements like, processors, 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.

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 to implement. 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.

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.

II. FAULT-TOLERANCE IN VLSI AND MSI

The issue of fault-tolerance in VLSI and MSI arrays has been the subject of recent studies, e.g., [3,4,7-10,13-15]. In these publications, various schemes have been proposed that introduce fault-tolerance into the architecture of regular arrays. Because fault-tolerance is an involved subject, completely different schemes might be cost effective in different situations and for different objective functions.

When evaluating a fault-tolerance strategy for multi-element systems we have to consider the following aspects:

- (a) Types of failures to be dealt with.
- (b) Costs associated with failure occurrences.
- (c) Applicable recovery methods.
- (d) Amount of additional hardware needed.
- (e) System objective function.

Fault-tolerance strategies can be designed to deal with two distinct types of failures, namely, production defects and operational faults. In the current technology, a relatively large number of defects is expected when manufacturing a silicon wafer, leading to a low yield.

Operational faults have in comparison a considerably lower probability of occurrence. Improvements in the 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 ICs. In contrast, faults occur after the system has been assembled and is already operational. Hence, their impact is on the system's operation 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

⁺ Supported in part by AFOSR contract B4-0052.

as to isolate it. Various link technologies are available now which allow such restructuring. Included among these are the laser-formed links, fusible links, and so on.

The restructuring capability can be either static or dynamic. Static schemes are suitable only to avoid the use of parts with production flaws. 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. Static schemes tend to use comparatively less hardware but consume operator time, while dynamic schemes are controlled internally by the system and 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 switching elements, (e.g., [3,13,15]) or redundancy in processors or communication links (e.g., [4,7,10]) When carrying out such an analysis we have to take into account the following two parameters:

- (1) The relative hardware complexity of processors, links, switches and other system elements.
- (2) The susceptibility to failures (defects or operational faults) of all types of elements.

Processing elements are traditionally considered the most important system resource; hence, achieving 100% utilization of them is many times attempted. For example, in [3,13,15] switching elements are added between PEs to assist in achieving this goal. In [3] and [10] connecting tracks are added on the wafer to be used in bypassing the defective PEs 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 [15]) or to additional links cannot be ignored. Consequently, such schemes might be beneficial only for PEs which are substantially larger than the switches and the links. Also, the addition of switches and especially the longer interconnections between active PEs 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 PEs. 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., switches, links or registers) is failure-free and only PEs 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 VLSI, the silicon area devoted to a system element might be more important than its hardware complexity. Consequently, 100% utilization of PEs is not necessarily the major objective, especially if this requires adding switches and/or links, which consume silicon real-estate. In the new technology, processors will be the expendable components, as gates were in SSI.

This may justify different fault-tolerance schemes which do not attempt to achieve 100% utilization of the fault-free PEs, when the array is restructured to avoid the use of faulty ones [7]. Such schemes, which give up the use of some fault-free PEs upon restructuring, can be attractive for operational faults (which are few in number). Here, the lack of additional hardware (switches or links) allows a larger number of PEs to fit into the same chip area, thereby offsetting the penalty of giving up the use of fault-free PEs when restructuring.

III. ANALYTICAL MODELS FOR YIELD AND PERFORMANCE

The motivation for incorporating redundancy into the architecture of VLSI-based systems is two-fold: yield enhancement and performance improvement. To achieve these two goals, redundancy has to be introduced either at the basic element level or/and at the system level. In the latter case, redundant elements can be added to the original design and they will be used to replace defective ones after the manufacturing process has been completed. Such a replacement is done by reconfiguring the system using either a static scheme or a dynamic one. Once this procedure is completed the system goes into operation and it has to handle from this point on only operational faults. This can be done using a dynamic scheme which might be different from the one used for defects. At this point the fault-tolerance capacity of the system is used to improve its performance. 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.

Our objective is to formulate an analytical model that will enable us to consider both defects and operational faults. Such a model will allow us to analyze the effectiveness of a given fault-tolerance technique in increasing yield and improving performance, or find the tradeoff between the two. It will also enable us to compare various fault-tolerance techniques, examine different system topologies and determine the optimal amount of redundancy to be added.

To formulate an appropriate model we need first an expression for the yield of a fault-tolerant VLSI chip. Such expressions have been presented in [8] and [11]. In what follows we propose a more general expression for 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 [17] caused by minute particles deposited on the wafer. Hence, each of them may affect only a single element (like a processor, bus, etc.) in a chip.

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 [17]. One model which has been shown to agree with experimental results, is the generalized negative binomial distribution [16]. 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. The probability of having x defects on a chip for this distribution is,

$$P\{X=x\} = \frac{[\Gamma(x+\alpha)]}{z! \Gamma(\alpha)} \cdot \frac{\left[\frac{\bar{\lambda}}{\alpha}\right]^x}{\left[1 + \frac{\bar{\lambda}}{\alpha}\right]^{\alpha+x}} \quad (1)$$

where $\bar{\lambda}$ is the average number of defects per chip 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.

For non-redundant chips the yield is the probability of having zero defects,

$$Y = P\{X=0\} = \left[1 + \left(\frac{\bar{\lambda}}{\alpha}\right)\right]^{-\alpha} \quad (2)$$

Suppose now that redundancy is added to a chip so that s defective elements can be tolerated, (i.e., replaced by good spares), and denote by N the total number of elements (e.g., PEs). Then, the chip is acceptable with any number of defects as long as all of them are restricted to at most s elements. The yield, which is now the probability of a chip being acceptable, is given by.

$$Y = \sum_{x=0}^s Pr\{x \text{ defects in at most } s \text{ elements}\} \quad (3)$$

If we denote,

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

$$\text{Then, } Y = \sum_{x=0}^s \sum_{i=0}^x Q_{x,i}^{(N)} \cdot Pr\{\text{There are } x \text{ defects}\} \quad (4)$$

The last term in the above equation is $Pr\{X=x\}$ and we may substitute it by (1) or a similar expression for any other defect distribution (e.g., Bose-Einstein statistics [11]).

The probability $Q_{x,i}^{(N)}$ is given by.

$$Q_{x,i}^{(N)} = \sum_{k=0}^i (-1)^k \binom{N}{i,k,N-i-k} \left[\frac{i-k}{N} \right]^x \quad (5)$$

where $\binom{N}{i,k,N-i-k}$ is the multinomial coefficient.

In the previous discussion we have assumed that only one type of elements can have defects. If two types of elements (e.g., PEs and busses) can have defects, then instead of (1) we have to use the following equation, which is the probability of having x_1 defects in type 1 elements and x_2 defects in type 2 elements [8].

$$Pr\{X_1=x_1, X_2=x_2\} = \frac{\Gamma(x_1+x_2+\alpha)}{x_1!x_2!\Gamma(\alpha)} \frac{\left(\frac{\lambda}{\alpha}\right)^{x_1+x_2} \delta^{x_2} (1-\delta)^{x_1}}{\left(1+\frac{\lambda}{\alpha}\right)^{\alpha+x_1+x_2}} \quad (6)$$

where δ is the percentage of defects in type 2 elements out of all chip defects. A first approximation for δ might be the percentage of silicon area devoted to type 2 elements. The generalization of (6) to three or more types of elements is straightforward.

Suppose now that s_1 defective elements of type 1 and s_2 defective elements of type 2, out of N_1 and N_2 elements, respectively, can be tolerated. Then, the yield is given by.

$$Y = \sum_{x_1=0}^{s_1} \sum_{x_2=0}^{s_2} \left[\sum_{i_1=0}^{x_1} Q_{x_1,i_1}^{(N_1)} \right] \left[\sum_{i_2=0}^{x_2} Q_{x_2,i_2}^{(N_2)} \right] \cdot Pr\{X_1=x_1, X_2=x_2\} \quad (7)$$

Equation (7) as well as (4) can be multiplied by a "bypass coverage probability" [11]. This 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.

In the following we will adopt the commonly used assumption that only one type of elements can fail (usually, the more complex one). The general case in which all system elements can have defects in them, can be analyzed based on expressions similar to (7).

To tolerate s defective elements, at least s 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_s denote the increase in chip area (due to the addition of redundancy) needed to tolerate these s faulty elements. The factor γ_s is called the redundancy factor [8] and it depends on the system topology and the reconfiguration strategy. It assumes its lowest possible value when only s redundant elements are included in the total of N elements, hence

$$\gamma_s \geq N / (N - s)$$

To take into account the increased number of expected defects due to the increase in area, we have to substitute λ by $\gamma_s \lambda$ in (1). This is based on the assumption that the average number of defects per chip increases linearly with the silicon area. A more complex relationship between λ and the increase in silicon area can be handled by the proposed model.

In addition, any increase in chip area will reduce 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, called equivalent yield in [8], is obtained from (4) after dividing it by γ_s . By comparing the equivalent yield of the fault-tolerant chip and the yield of one with no fault-tolerance 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 [8]. 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.

Chips having s or less defects will be accepted and then reconfigured to avoid the use of the defective elements. If the number of defects was less than s , the chip has some "residual" redundancy which can then be used for performance enhancement, i.e., handle operational faults. Even chips in which no redundant elements are left when leaving the manufacturing site (i.e., there were originally s defects in the chip), can still benefit from the fault-tolerance capability.

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 like the one employed by [8] and [2].

Suppose first that the same reconfiguration scheme is used to avoid defects and operational faults as well. This assumption implies that a dynamic scheme is employed since no static scheme can be used while the system is in operation. The suggested Markov model for this case is depicted in Figure 1, where (F) is the system failure state and (i) is a state at which the system is operational in the presence of i faulty elements. A transition from state (i) to state (F) takes place when an additional node becomes faulty and the system fails to recover from its effect. The corresponding transition rate is denoted by α_i^F . Similarly, α_{i+1}^i is the transition rate from state (i) to state (i+1). These transition rates depend upon the failure rates of the system's elements and the coverage probability [8].

State (o) in Figure 1 is the initial state of the system if no defects occurred while the chip has been manufactured. If there were i defective elements ($0 \leq i \leq s$) then (i) will be the initial state. Let α_i denote the probability of this event.

$$\alpha_i = \sum_{x=0}^i Q_{x,i}^{(N)} \cdot Pr\{X=x\} \quad (8)$$

Using α_i we can calculate the yield as

$$Y = \sum_{i=0}^N \alpha_i \quad (9)$$

State (s+m) in Figure 1 is a terminal state [8] (i.e., a state from which the only transition possible is to the system failure state), where m is the largest number of faulty elements that the system can tolerate if no redundant elements were left when the system went into operation.

Let $P_j^i(t) = P\{ \text{The system is in state } (j) \text{ at time } t /$

The system was initially in state (i) }

$$i = 0, 1, \dots, s; \quad j = i, i+1, \dots, s+m$$

with $P_i^i(0)=1$ and $P_j^i(0)=0$ for $j>i$.

The Markov model in Figure 1 is described then by the following differential equations:

$$\frac{dP_i^i(t)}{dt} = -\alpha_i P_i^i(t) \quad (10)$$

$$\frac{dP_j^i(t)}{dt} = -\alpha_j P_j^i(t) + \alpha_j^i P_{j-1}^i(t) \quad (11)$$

where $j=i+1, i+2, \dots, s+m$ and $\alpha_j = \alpha_j^f + \alpha_j^{i+1}$

The solution of (10) and (11) under the condition $\alpha_j \neq \alpha_k$ for all $(k) \neq (j)$ which is satisfied in most practical cases, is

$$P_j^i(t) = \alpha_i^{i+1} \alpha_i^{i+2} \dots \alpha_{j-1}^j \sum_{v=i}^j \prod_{\substack{u=i \\ v \neq u}}^j \frac{e^{-\alpha_u t}}{(\alpha_u - \alpha_u)} \quad (12)$$

$$P_i^i(t) = e^{-\alpha_i t} \quad (13)$$

For the Markov model shown in Figure 1 we can calculate several performance measures like Reliability, Performance, Computational availability and Area utilization [8]. Let $R_i(t)$ ($0 \leq t \leq s$) denote the reliability of a system which had i defects during manufacturing. This can be calculated from the above model as follows,

$$R_i(t) = \sum_{j=i}^{s+m} P_j^i(t) \quad (14)$$

We may then define and compute

$$R(t) = \sum_{i=0}^s \alpha_i R_i(t) \quad (15)$$

as the average reliability of a system having s or less defects when manufactured. This average reliability is a function of s (or in general, the amount of added redundancy). It can then be compared to the reliability of the system when $s=0$ to determine whether it is beneficial when reliability is considered, to introduce redundancy into the architecture of the system.

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 measures can be computed using the same model. For example, the computational availability $A_i^c(t)$ (the expected available computational capacity) and area utilization measure $U_i(t)$. The latter takes into account the additional area needed when fault-tolerance is introduced into the system, and is defined in the following way.

$$U_i(t) = \frac{\text{Computational Availability } A_i^c(t)}{\text{Total chip area } A}$$

The expression for the above introduced computational availability measure is,

$$A_i^c(t) = \sum_{j=i}^{s+m} c_j P_j^i(t) \quad (16)$$

where c_j is the computational capacity of the system in state (j) [8], expressed for example in instructions per time unit. The computational capacity depends mainly on the number of processors available for computation in state (j). This number is at most $N-j$ (where N is the number of PEs in the fault-free system), and is determined by the reconfiguration strategy. In addition, c_j depends on the current system structure and application since not all PEs are utilized in every possible structure or application.

Other performance measures, like mean time to failure, can also be calculated. For example, let T_i denote the mean time to failure of a system which was initially in state i , then

$$T_i = \int_0^\infty R_i(t) dt \quad (17)$$

The Markov model depicted in Figure 1 describes a system which has redundancy only at the system level. Here, a single defect or fault in an element requires the 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. The second failure will be assumed to be 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 is shown in Figure 2. Here, (i,j) is a state at which the system is operational in the presence of i faulty nodes and j partially faulty nodes (e.g., nodes having a single faulty processor). As before, $s+m$ is the maximum number of defective and faulty nodes that the system can tolerate. The initial state of the model is determined by the number and distribution of the 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 (i,j) : $i=0, 1, \dots, s$; $j=0, 1, \dots, N-1$ are possible initial states. Let $\alpha_{i,j}$ be the probability of (i,j) being the initial state, then this is the probability of having $2i+j$ defects out of which exactly $2i$ appear in pairs, hence,

$$\alpha_{i,j} = \frac{\binom{N}{i} \binom{N-j}{j} 2^j}{\binom{2N}{2i+j}} \alpha_{2i+j} \quad (18)$$

where α_{2i+j} is given by (8) with N replaced by $2N$.

The yield of the chip is,

$$Y = \sum_{i=0}^s \sum_{j=0}^{N-i} \alpha_{i,j} \quad (19)$$

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 therefore by,

$$Y = \sum_{i=0}^s \sum_{j=0}^{N-i} \alpha_{i,j}$$

The state probabilities for the Markov model shown in Figure 2 are defined as follows,

$P_{i,j}^k(t) = P\{ \text{The system is in state } (i,j) \text{ at time } t /$

The system was initially in state } $(k,l) \}$

$k=0, 1, \dots, s$; $l=0, 1, \dots, N-s$; and $(i,j) \geq (k,l)$ lexicographically, with $P_{i,j}^k(0) = 1$ and $P_{i,j}^k(0) = 0$ for $(i,j) > (k,l)$ lexicographically.

The above state probabilities satisfy the following differential equations:

$$\frac{dP_{k,j}^x(t)}{dt} = -\alpha_{k,i} P_{k,j}^x(t) \quad (20)$$

$$\frac{dP_{k,j}^x(t)}{dt} = -\alpha_{i,j} P_{k,j}^x(t) + \alpha_{i,j-1,j+1}^x P_{k,j-1}^x(t) + \alpha_{i,j-1}^x P_{k,j-1}^x(t) \quad (21)$$

where

$$\alpha_{i,j} = \alpha_{i,j+1}^{i+1} + \alpha_{i,j}^{i+1} + \alpha_{i,j}^F \quad (21)$$

We denote by v_m any vertex (state) (i,j) satisfying $2i+j=m$. For example, v_5 is any one of the states $(2,1)$ and $(1,3)$ in Figure 2. Thus, the sequence $(k,l), v_{2k+l+1}, v_{2k+l+2}, \dots, v_{2k+j-1}, (i,j)$ corresponds to a path in Figure 2 from the initial state (k,l) to the state (i,j) .

Using this notation, the solution of (20) under the condition $\alpha_{i,j} \neq \alpha_{k,i}$ for all $(x,y) \neq (i,j)$ which is usually satisfied, is

$$P_{k,j}^x(t) = \sum_{(k,l), v_{2k+l+1}, \dots, v_{2k+j-1}, (i,j)} \alpha_{k,l}^{v_{2k+l+1}} \alpha_{v_{2k+l+1}}^{v_{2k+l+2}} \dots \alpha_{v_{2k+j-1}}^{(i,j)} \cdot \sum_{m=2k+1}^{2i+j} \frac{e^{-\alpha_{v_m} t}}{\prod_{n \neq m} (\alpha_{v_n} - \alpha_{v_m})} \quad (22)$$

$$P_{k,j}^x(t) = e^{-\alpha_{k,i} t} \quad (23)$$

where v_{2k+l} and v_{2k+j} in (22) specify the states (k,l) and (i,j) , respectively. The summation in (22) is over all paths (each consisting of $(2i+j) - (2k+l) + 1$ nodes) in the model (Figure 2) from state (k,l) to the state (i,j) .

Using the state probabilities given by (22), one can derive closed-form expressions for various performance measures for different architectures and restructuring strategies.

The models presented can be extended in two directions in order to make them more general and more practical. One is to include two or more types of system elements that can fail like communication buses, switches etc. The second one is to allow the use of one reconfiguration scheme to handle defects and a different one to handle operational faults.

IV. CONCLUSIONS

Fault-tolerant architectures that use redundancy for yield and performance improvement have been considered. This paper is devoted to the development of an analytical model for the evaluation of performance and yield improvement through redundancy. 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. The models proposed here can be used to study the effect of sharing element level and system level redundancy, between these two somewhat competing requirements.

V. REFERENCES

- [1] R.P. Censer et al., "A Fault-Tolerant 64K Dynamic Random-Access Memory", *IEEE Trans. on Electron Devices*, Vol. ED-26, June 1979, pp. 853-860.
- [2] J.A. Fortes and C.S. Raghavendra, "Dynamically Reconfigurable Fault-Tolerant Array Processor," *Proc. of the 14th Annual Symposium on Fault-Tolerant Computing*, June 1984, pp. 386-392.
- [3] D. S. Fussel and P. J. Varman, "Fault-Tolerant Wafer-Scale Architectures for VLSI," *Proc. of the 8th Annual Symp. on Comp. Arch.*, May 1982.
- [4] J.W. Greene and A. El Garmal, "Area and Delay Penalties in Restructurable Wafer-Scale Arrays," *Proc. of the 3rd Caltech Conf. on VLSI*, March 1982, pp. 185-184.
- [5] D. Gordon, I. Koren and G.M. Silberman, "Fault-Tolerance in VLSI Hexagonal Arrays," Technical report, Dept. of Elec. Eng. Technion, Israel.
- [6] K. Hedlund and L. Snyder, "Wafer Scale Integration of Configurable, Highly Parallel (Chip) Processor," *Proc. of the 1982 Internl. Conf. on Parallel Processing*, August 1982, pp. 262-265.
- [7] I. Koren, "A Reconfigurable and Fault-tolerant VLSI Multiprocessor Array," *Proc. of the 8th Annual Symp. on Comp. Architecture*, May 1981, pp. 425-441.
- [8] I. Koren and M.A. Breuer, "On Area and Yield Considerations for Fault-Tolerant VLSI Processor Arrays," *IEEE Trans. on Comp.*, Vol. C-33, Jan. 1984, pp. 21-27.
- [9] H.T. Kung and M.S. Lam, "Fault-Tolerance and Two-Level Pipelining in VLSI Systolic Arrays," *Proc. Conf. on Advanced Res. in VLSI*, MIT, Jan. 1984, pp. 74-83.
- [10] F.T. Leighton and C.E. Leiserson, "Water-Scale Integration of Systolic Arrays," *Proc. of 27th Ann. Symp. on Foundations of Comp. Sci.*, Nov. 1982, pp. 297-311.
- [11] T.E. Mangir and A. Avizienis, "Fault-Tolerant Design for VLSI: Effects of Interconnect Requirements on Yield Improvements of VLSI Designs," *IEEE Trans. on Computers*, Vol. C-31, July 1982, pp. 609-615.
- [12] D.K. Pradhan, "Fault-tolerant Architectures for Multiprocessors and VLSI Systems", *Proc. of the 13th Symposium on Fault-Tolerant Computing*, June 83.
- [13] A. L. Rosenberg, "The Diogenes Approach to Testable Fault-Tolerant Arrays of Processors," *IEEE Trans. on Computers*, Vol. C-32, Oct. 1983, pp. 902-910.
- [14] C.H. Sequin and R.M. Fujimoto, "X-Tree and Y-Component," in *VLSI Architectures*, B. Randell and P.C. Treleaven, eds., Prentice-Hall, 1983, pp. 299-326.
- [15] L. Snyder, "Introduction to the Configurable Highly Parallel Computer," *Computer*, Vol. 15, January 1982, pp. 47-56.
- [16] C.H. Shapper, A.N. McLaren and M. Dreckman, "Yield Model for Productivity Optimization of VLSI Memory Chips with Redundancy and Partially Good Product," *IBM J. Res. Dev.*, Vol. 24, May 1980, pp. 398-409.
- [17] C.H. Shapper, F.M. Armstrong and K. Saji, "Integrated Circuit Yield Statistics," *Proc. of IEEE*, Vol. 71, April 1983, pp. 453-470.

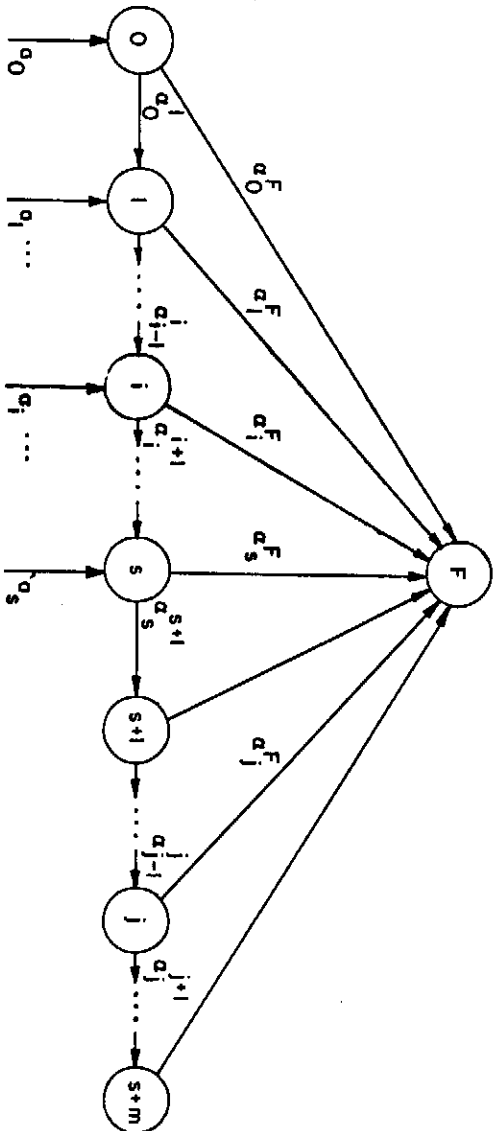


Figure 1: A Markov model for a system with defects and operational faults.

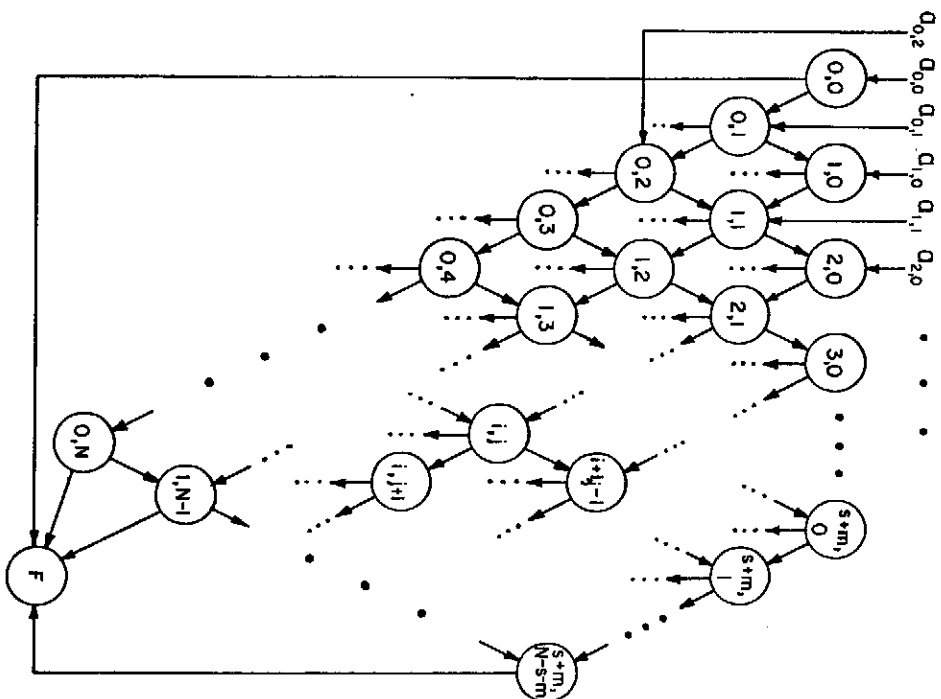


Figure 2: A Markov model for a system with element level and system level redundancy.