RELIABILITY MODELING OF NMR NETWORKS

J. A. Abraham
D. P. Siewiorek

Technical Report No. 56

DIGITAL SYSTEMS LABORATORY

Department of Electrical Engineering Department of Computer Science

Stanford University

Stanford, California

This work was supported by the National Science Foundation grant GJ-27527

ABSTRACT

A survey of the literature in the area of redundant system reliability modeling is presented with special emphasis on Triple Modular Redundancy (TMR). Areas where the classical method of TMR reliability prediction may prove inadequate are identified, like the interdependence of fault patterns at points of network fan-in and fan-out. This is especially true if the assumption of highly reliable subsystems, which is frequently made by the modeling techniques, is dropped, It is also not clear if the methods give an upper or a lower bound to the reliability. As a solution, a method of partitioning an arbitrary network into cells so that faults in a cell are independent of faults in other cells is proposed. An algorithm is then given to calculate a tight lower bound on the reliability of any such cell, by considering only the structure of the interconnections within the cells. The value of reliability found is exact if TMR is assumed to be a coherent system. An approximation to the algorithm is also described; this can be used to find a lower bound to the reliability without extensive calculation. Modifications to the algorithm to improve it and to take care of special cases are given. Finally, the algorithm is extended to N-Modular Redundant (NMR) networks.

1. INTRODUCTION

1.1 Introduction

The widely increasing use of computers in diverse areas has brought with it the need for very high reliability. Even if computers are constructed with components selected for very high reliability, these components will have a non-zero probability of failure. Thus highly reliable operation necessitates the use of some form of redundancy. Redundancy has been defined as the existence of more than one means of performing a function [1]. This could be brought about by providing extra time to perform the function, or by extra hardware within the computer, or by both.

Avizienis [2] has identified two forms of protective redundancy, massive (masking) redundancy, and selective redundancy. In massive redundancy, effects of faults are masked instantaneously by permanently connected and concurrently operating replicas of the faulty element. Selective redundancy requires detection, diagnosis, and corrective action to overcome the effects of faults. This latter approach generally assumes a hard core i.e., a set of logic circuits which must function continuously to insure the proper fault location and repair of the rest of the system. The system hard core is usually protected by some massive redundancy scheme.

Whereas accelerated life tests on many copies of a component may be feasible to experimentally determine the component. reliability as a function of time, computer systems are too complex and often too

expensive to subject to such tests. Thus to evaluate and compare various redundant system designs, a reliability modeling technique is required. With such a model it becomes possible to predict system behaviour and, in particular, determine whether the proposed system meets the design specifications.

Modeling requires a mathematical or physical representation which incorporates the salient parameters of the modeled system [3]. A model is an incomplete representation of the subject under study. To be of value, the modeling technique must be convenient to apply, and must successfully predict the behaviour of the subject under various parameter changes. If the reliability model is accurate, then insights can be gained as to how the system reliability changes as a function of the design parameters. This requires knowledge of the model's predictive properties under all possible system designs, i.e., is it an upper bound, a lower bound, or simply a "good guess"? The following discussion will illustrate some common network configurations where the reliability modeling techniques in the literature for massive redundancy are sometimes inadequate predictors of system reliability. Modifications which enable the classical reliability modeling techniques to handle the troublesome network configurations will be demonstrated. Finally, a new approach to reliability modeling will be presented which is much more accurate when compared with previous methods,

1.2 Background

The basic concept underlying massive redundancy reliability modeling schemes has been to enumerate or approximate the number of states for which a system still realizes its desired function. Each component in a system can have two states, failed or good, and the state of all the components represents the state of the system. Massive redundancy schemes are designed to tolerate component failures, thus the number of working states in a redundant system may be quite large, and the general approach taken is to partition the system into cells such that the system is working if all the cells are working, The cell reliabilities are then said to be statistically independent. Thus the system reliability is just the product of the cell reliabilities. Except in very specialized situations, the system cannot be partitioned like that, so that the cell reliabilities are not statistically independent. Then a small portion of the system is usually selected as a cell such that the statistical dependency between cell reliabilities is a second order effect, i.e., consists of higher powers of component unreliability than those considered in the cell [4,5].

To date most studies have pertained only to simple models of digital systems, namely the visualization of a computer as a cascade of single input, single output blocks [6,7] or as a tree network of double input single output blocks [8]. Even so the estimated relative magnitude of the second order effect in a very specialized network is 10% [4] while networks to be presented here show differences of 20% or more. To counteract this second order effect the customary assumption

is that the components are very reliable, say 0.99 or better. This may not be a bad assumption for current redundancy applications such as aerospace where system cost is not the primary design constraint, As redundancy techniques find more and more applications in the commercial sphere, the systems designer may trade the costly, highly reliable components in a nonredundant or low redundancy configuration for cheaper less reliable components in a highly redundant configuration to achieve the desired system reliability. In systems requiring maintenance-free operation over a long period of time, the designer may want to see the effects of component reliability degrading to below the high reliability values given above. In these cases, a reliability model which is accurate over all ranges of component reliability, not just high component reliability is needed.

The primary vehicle for this discussion will be Triple Modular Redundancy (TMR). TMR augmented by standby spares is a prime candidate for hard cores in self-repairing computers [9]. It has been used on the Saturn V launch vehicle computer [10]. TMR is easy for a designer to apply and has several good features [11]:

1) The scheme is equally effective for both wrong 0 and wrong 1 errors.
2) The correction mechanism (voters) may be realized in the same logic technology as the circuits being protected. No special elements are required.
3) The size of the module protected is unlimited, it may be a single gate or a whole computer.

4) No modifications to the modules is needed, either in network structure or in factors of usage such as fan-in or fan-out.

5) The scheme is directly extendable to higher orders of redundancy and may indeed employ different orders of redundancy within the same system without causing any special problems in design.

6) As mentioned earlier, TMR is very well suited for standby redundancy schemes, while other massive redundancy techniques like Quadded logic [12] do not lend themselves to this.

Finally, TMR cells are more readily defined and provide less intercell dependence than other existing massive redundancy techniques such as Quadded logic [12] and its descendants, radial logic [13] and dotted logic [14]. Hence the second order effects exhibited here will be even more dramatic in these other schemes of greater cell dependancy. The techniques to follow are translatable into handling these other redundancy schemes. The extension to NMR will be given later.

The discussion will consider the interconnection pattern of the logic modules in a TMR system without regard to the internal logic design of the modules. It will be assumed that the modules have a known reliability as a function of time. A module is assumed to be faulty if it produces a wrong output for some input combination, and we will assume that a wrong signal at the input of a module produces a wrong signal at the output of that module. These assumptions have to be made since we do not know the internal structure of the modules or voters and this gives a lower bound on the reliability. A different reliability model

which depends on the actual logic implementation of the modules and voters has been discussed by Siewiorek [15].

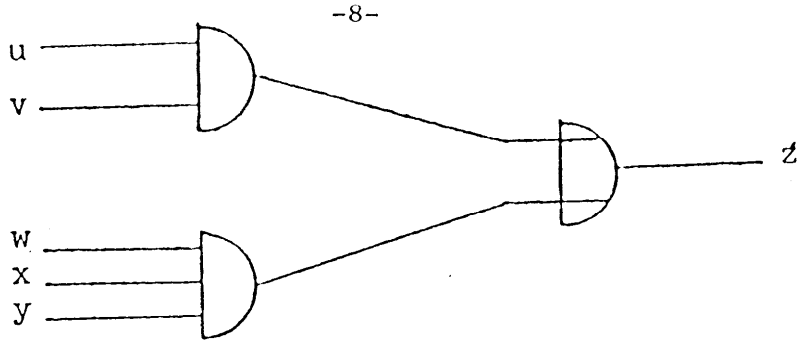
2. RELIABILITY MODELING OF TMR NETWORKS

2.1 Introduction

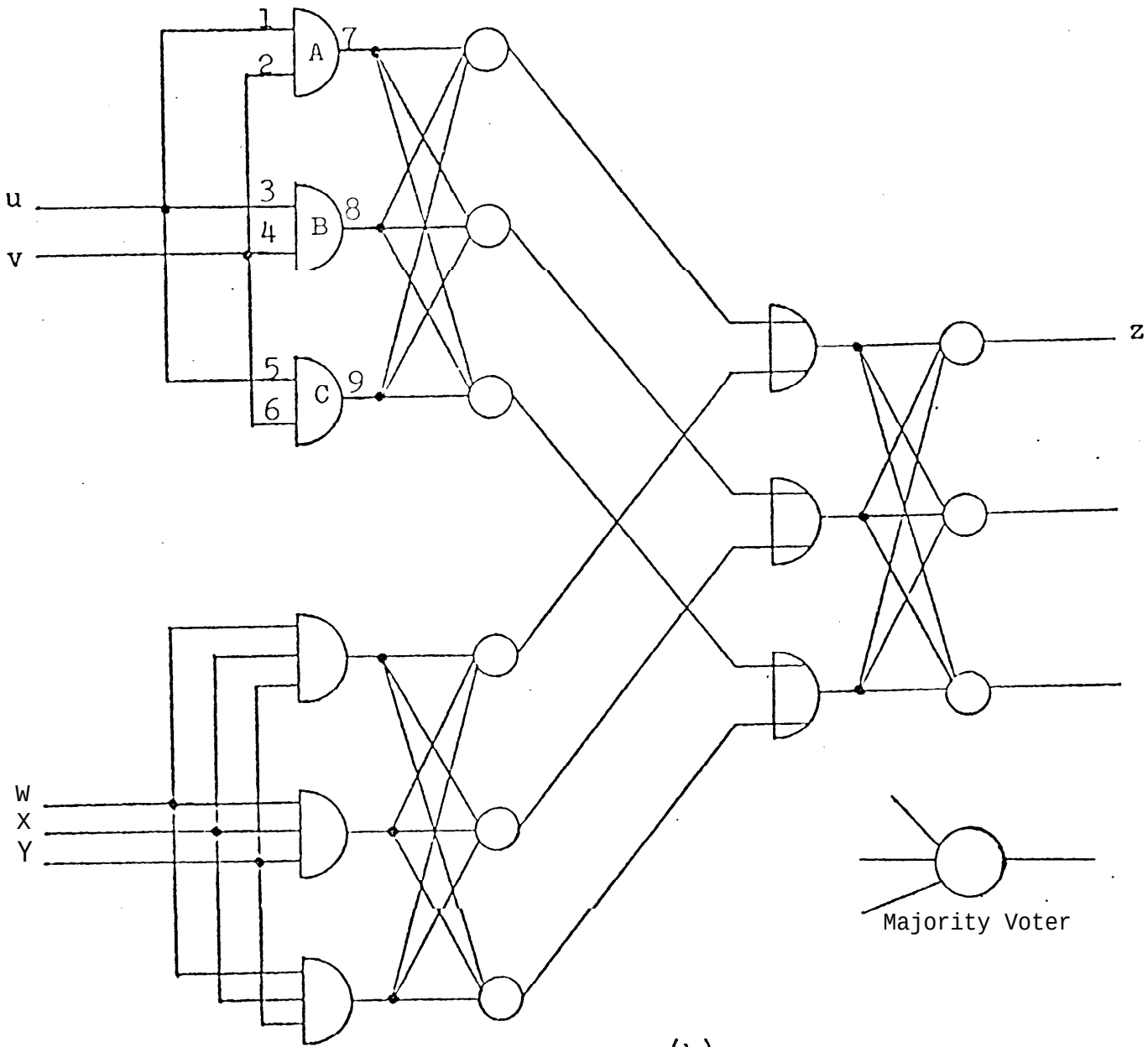
With the introduction of the restoring organ by von Neumann in 1956 [16] the groundwork was laid for the triple - modular redundancy (TMR) technique. Briefly, TMR consists of dividing a non-redundant circuit into several modules, triplicating the modules, and inserting a majority gate (sometimes referred to as a voter) between the triplicated modules.

Figure 1 depicts the application of TMR to a simple function. Here, the network is partitioned so that each logic gate represents a module. The gates are triplicated and trios of voters, where a circle represents a voter, are inserted between them. Each voter receives three inputs, one from each of the triplicated modules. Since the reliability model under consideration is independent of the internal module design, the modules of Fig. 1 could be represented by squares as in Fig. 2 (a)., As a notational convenience only one path from the system inputs to the system outputs will be shown as in Fig. 2 (b). This will uniquely define the redundant system. A path is defined as the components of a system which a logical input to the system could affect on its passage to a system output.

Figure 3 shows TMR in its simplest configuration -- triplicated modules followed by triplicated voters. Networks whose nonredundant form may be represented by a serial cascade of modules will be referred to as

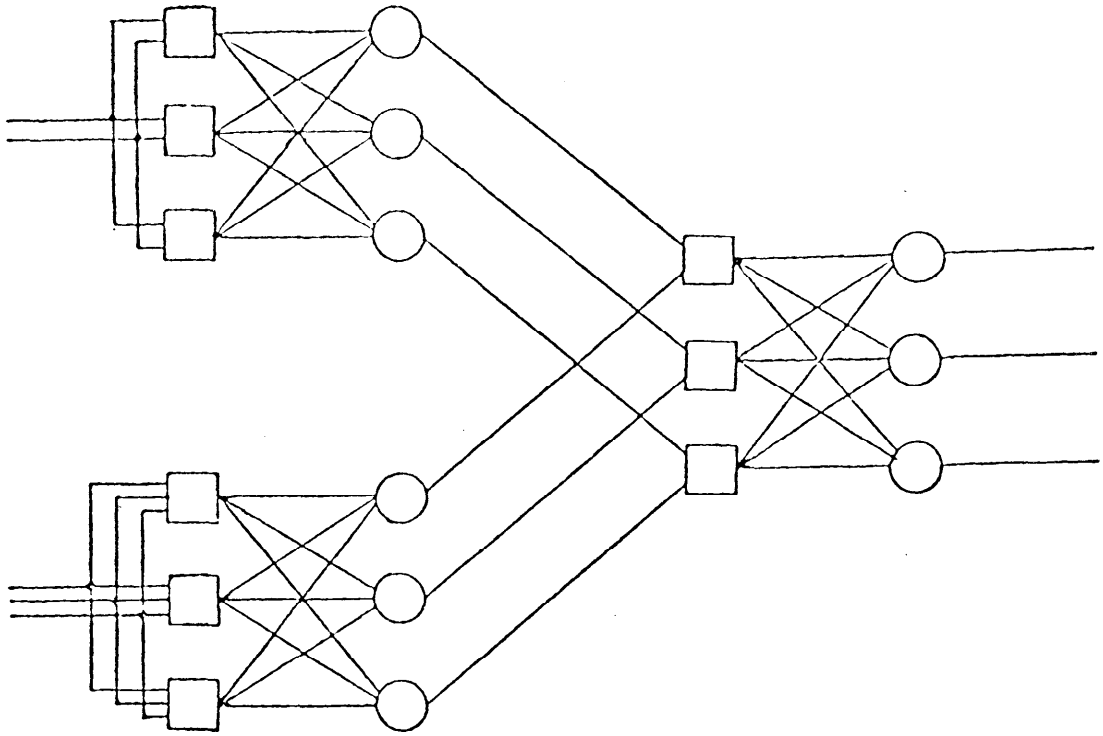


(a)

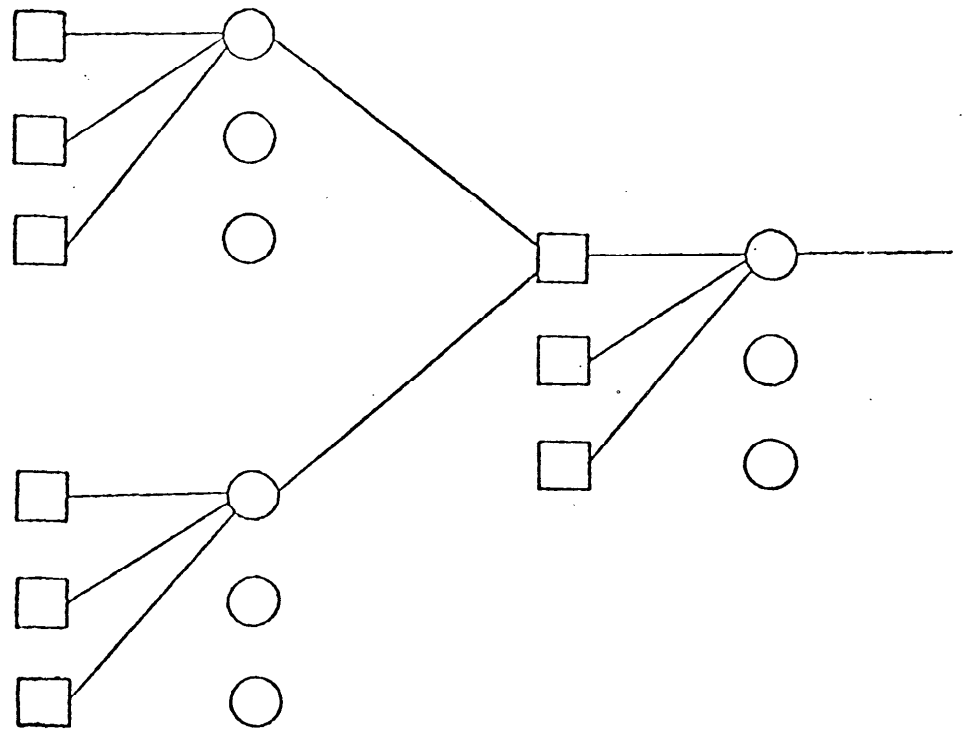


(b)

Fig. 1. The application of TMR to the function z = uv + wxy; the nonredundant version (a) and the redundant TMR version (b).



(a)



(b)

Fig. 2. A generalized form (a) for the modules of Fig. l(b) and (b) an abbreviated system representation,

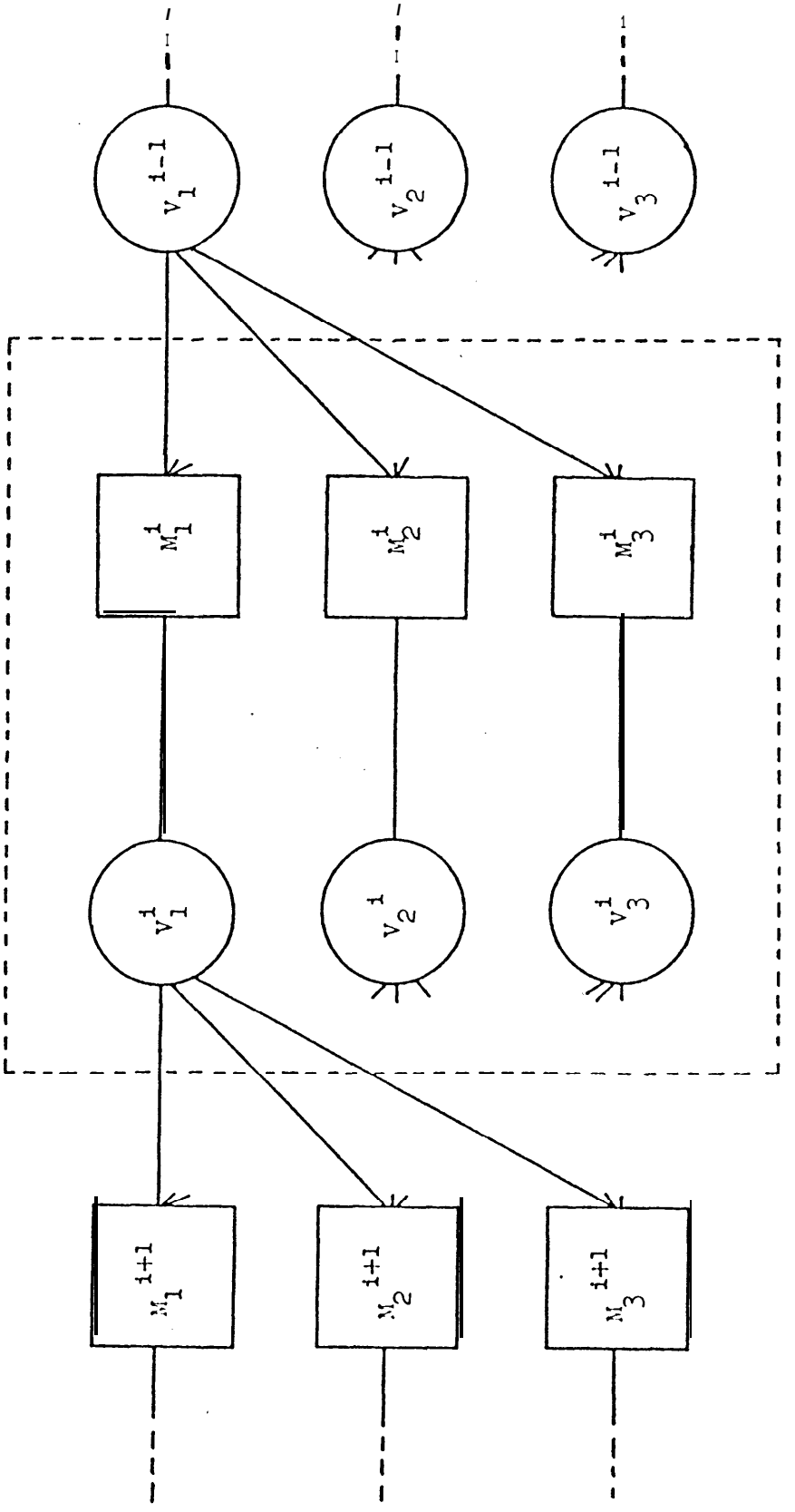


Fig. 3. A serial TMR cell: triplicated modules followed by triplicated voters.

serial TMR. Fig. 3 outlines a serial TMR cell. Normally the input and output lines for a module represent busses, Thus the voter symbol as well as the module symbol should be thought of as operations on vectors rather than on single bits of information.

A binary up-counter using J-K flip-flops would be an example of a serial cascade network if each flip-flop was taken as a module. In general, few networks can be characterized by a serial cascade of modules, However the reliability of serial TMR networks is easily calculated and hence the serial TMR cell has been used to predict the reliability of more complicated networks. But extreme care should be exercised in applying the results derived from consideration of this simple model. For example, a segment of an arithmetic unit – data bus system for a computer will be modeled in a subsequent section. The non-redundant system required 17 modules and exhibited a probability of failure of 0.0335 (for a particular failure probability of each module). With triple replication and the use of $3 \cdot 17 = 51$ voters the probability of failure was decreased to 0.0035. On the basis of visualizing the computer segment as a cascade of 17 serial cells the failure probability was predicted as only 0.0001.

For the remainder of this discussion the reliability of a system (module) will mean the conditional probability that the system (module) will be capable of performing its specified function at time t given that all system (module) components are functioning properly at time $t = 0$. For basic components, such as resistors, failures are

assumed to be an independent random variable of time and the number of expected failures is the same for any equal intervals of time. Then the reliability R (t) is given by the Poisson distribution for n = 0, where n is the number of expected failures in the time interval from 0 to t:

$$r(t) = e^{-\lambda t} \left| \frac{(\lambda t)^n}{n!} \right|_{n=0} = e^{-\lambda t}$$

The reliability of modules will usually be more complicated than the exponential given above. In a nonredundant module all the components must function for the module to function. The reliability will be the product of the exponential component reliabilities. On the other hand a redundant module will require only one of several subsets of its components to function. Thus its reliability will be a sum of products, where each product represents one of the subsets.

In the subsequent formulas time is an implicit variable. To calculate system reliability at time t the module reliability must be evaluated at time t. Wherever numbers appear for R, R_v, etc., a time t is implied.

The following notations will be adopted:

R: redundant system reliability

R_0: non-redundant system reliability

(non-redundant means that successful operation of all modules is a necessary condition for successful operation of the system,)

R_v: voter reliability

2.2 Background

Several investigators have addressed the problem of modeling the reliability of TMR or multiple-line networks [4,6,11,17,18,19,20,21, 22,23,24]. The first approach was to approximate the system by a serial TMR system, i.e., modeling the network as a cascade of single input, single output modules, adding extra voters if required, This was the essence of the procedures developed by Brown [17], Teoste [18] Rhodes [19], Longden [20], Lyons [24], and Gurzi [6]. A summary of their work follows.

Brown et.al. [17] considered the single and triplicated voter cases. When voters are triplicated they may be associated with the inputs to modules (in which case Brown added extra majority gates at network fan-out points to retain the serial voter-module arrangement depicted in Fig. 3) or with the outputs of modules. In the latter case Brown introduced the concept of symmetric chains (a specialized network situation where only voters fan-out to modules in the next layer of triplicated modules, the non-redundant system being essentially approximated by a cascade of modules). Brown then concluded that associating voters with module outputs would be more efficient than associating them with module inputs. But Brown penalized the voter-input pairing by introducing more voters into the network at fan-out points in order to help isolate the cells of the network. This was to facilitate reliability modeling and is not a restriction on the application of TMR. It should make no difference to the reliability prediction whether voters are associated with module inputs or outputs as long as the network is the same for both

approaches. The method essentially approximates network reliability by use of serial cells with no indication whether the prediction tends to be higher or lower than the actual reliability.

Teoste [18], Rhodes [19], Longden, et. al, [20], and Gurzi [6] also use serial voter-module cells to approximate network reliability. In the case of fan-out, Longden adds extra voters in a manner similar to Brown. Rhodes attempts to improve the calculation of cell reliability by including cases where multiple failures in a cell (such as one module output failing to a constant 1 and another to constant 0) could still be corrected by the next voter layer. Rhodes, however, incorrectly assumes that module failures to constant 1 and 0 are the only module failures possible. Rhodes also included some multiple failure situations which do not lead to correct network operation for all types of models, For example, Rhodes allows a voter to be stuck at logical one and a module which receives inputs from another voter to be stuck at zero. If the modules were inverters then there would always be two or more zeroes on the module outputs. Hence the network fails.

Rubin [21] drvides the network into augmented blocks in an attempt to find sections of the network in which failures impose no restriction on failure patterns in other sections, i.e., a failure in one cell cannot combine with a failure in another cell to cause system failure. He models networks as serial cells and inserts fictitious module trios where required to make all the cells serial cells. Then he alters the

standard serial voter-module reliability formula to take into account these added fictitious modules. He gives no algorithm in [21] to enumerate the augmented blocks (a non-trivial problem in large, complicated nets) and no indication as to whether this approach gives an optimistic or pessimistic prediction of the actual network reliability,

Two recent approaches that do not use the serial cell approach are by Klaschka [4] and Jensen [22]. Their procedures rely heavily on the work of Esary and Proschan [25] in regards to coherent systems, systems which having once failed cannot work properly again upon failure of more network components. Klaschka assigns each minimal cut of the network to a cell and then assumes the cell interdependence is a second order effect. A network cut is a set of components whose failure causes system failure. A minimal cut is a cut from which no members can be deleted without the set losing the property of being a network cut. The probability obtained by taking the product, over all minimal cuts, of the probability that the cut does not occur is a lower bound on coherent network reliability [25]. Jensen [5] demonstrates that a non-coherent network with the same minimal cuts as a coherent network is more reliable than the coherent system. TMR and Quadded logic form non-coherent networks as demonstrated in the next section. So Klaschka's, as well as Jensen's, approach utilizes an approximation to the lower bound on the reliability of a coherent system as an approximation to the reliability of a non-coherent network. Jensen uses matrix manipulation to establish the minimal cuts of a network. However, if there are n modules in the non-redundant network, Jensen's method in the

worst case requires on the order of n^{2n} operations and on the order of n^2 storage locations just to set up the matrices for determining the minimal cuts. A more typical case would still require about n^4 operations.

An adaptation of Jensen's technique is due to Goldberg et. al. [11]. The network graph of a redundant system can be considered as a collection of paths. When modules are connected without intervening voters, they occupy a single path. Voters receive inputs from each path (three in the case of TMR), intermix the signals, and issue a signal along another single path. Whereas Jensen considered any number of module failures along such a path, Goldberg assumes at most i failures, a value of two or three was suggested for i. Goldberg uses matrix techniques and requires on the order of n^{i+1} storage locations. This procedure also requires establishing the equivalent of Rubin's [21] cells and has all the problems thus entailed.

The cellular approach presented here is a rapid method to approximate very closely the reliability of an arbitrary TMR network. For now, a cell will be loosely defined as a segment of a network whose inputs all lead to voter trios and whose outputs issue from module trios. A more precise definition for a cell will be presented later but the intuitive notion of cells thus far established will be adequate until that time. Arbitrary cell types, in addition to the standard serial cell, are used to partition a network and approximate its reliability. The advantage over the methods of Klaschka and Jensen

resides in the fact that it is a specialized method for TMR and other multiple line redundancy schemes, and can take advantage of the known properties of the redundancy schemes. Klaschka and Jensen pay a penalty for utilizing a more general approach which is applicable to more than one class of redundancy schemes.

In [21] Rubin claims that the augmented block approach took significantly less time (1000-fold increase in speed for large nets) to calculate system reliability than the minimal cut, Monte Carlo, or actual reliability calculations for the same network . The cellular approach is on the order of complexity of calculation as the augmented block method once the cells and blocks have been determined. The augmented block approach, however, gives no clue as to whether in a particular situation it represents an upper or lower bound to actual system reliability. A brief discussion of when a redundant system is considered to have failed is presented before the actual problem of reliability modeling is undertaken.

2.3 Coherency

Esary and Proschan [25] define a structure function φ as a boolean function of $\underline{x} = (x_1, x_2, \ldots, x_n)$. Each component in the system is represented by an x_i where $x_i = 1$ if and only if the component functions properly and equals zero otherwise. A function φ is said to be monotone increasing if when $\underline{x} \leq \underline{y}$ (the comparison being made on a component by component basis), $\varphi(\underline{x}) \leq \varphi(\underline{y})$. A coherent system is then defined as one whose structure function is monotone increasing and such that $\varphi(\underline{1}) = 1$ and $\varphi(\underline{0}) = 0$.

If the system components are considered to be modules, the structure function for TMR networks is not well defined. Consider the TMR network of Fig. 1 where the modules are single gates. The AND gates lettered A, B, and C would be components of the structure function:

$$\varphi(x_A, x_B, x_C, x_D, \ldots, x_R).$$

Assume AND gate A and C failed such that their outputs became permanent logical zero. For u = v = 1, AND gate B would produce a logical one while gates A and C would be logical zero. The correct response of logical one would be outvoted. The network could thus produce an incorrect output for z and must be considered failed. For this situation:

$$\varphi(0, 1, 0, x_D, \ldots x_R) = 0.$$

For the purposes of the following discussion a failure pattern will be defined as an $\underline{x}$ where each x_i is evaluated. Now consider AND gate A failing so that its output takes on a permanent logical one value while AND gate C fails so that it realizes a permanent logical

zero output. Whatever value AND gate B assumes, the output of the voters will realize it since either gate A or gate C will agree with the properly functioning gate B and outvote and other faulty gate. The network will not fail. Such a situation where the majority of the modules in a cell can fail yet the network still functions properly will be referred to as compensating module failures and is explored in much greater detail in [15]. For this case:

$$\varphi\ (0,1,0,x_D, \ldots ,x_R) = 1.$$

The ambiguity arises from failed modules being able to take on both 0 and 1 values for the same vector $\underline{x}$ depending on the exact nature of the failures.. A module does not often fail such that its output is always in error as is often assumed in the literature [11,22].

Let us attempt to eliminate the difficulty above by letting x_i represent a lead rather than a module. φ is thus well defined, but TMR is not a coherent system as claimed by Jensen [22]. Again referring to Fig. 1, the leads in the triplicated version of the two input AND gate are numbered and represent system components:

$$\varphi\ (x_1,x_2,x_3,x_4,x_5,x_6,x_7,x_8,x_9,x_{10}, \ldots ,x_{33}).$$

Assume leads 1 and 9 are stuck to a logical zero. Both AND gate A and C would realize logical zero outputs and we have the first case that we described above. The network fails.

$$\varphi'(\underline{x}') = \varphi'(0,x_2,x_3,x_4,x_5,x_6,1,x_8,0,x_{10}, \ldots ,x_{33}) = 0$$

A further failure in lead 7 such that lead 7 were to take on a permanent logical one value causes AND gate A to realize a constant one while

AND gate C still realizes a constant zero, We have the second case above, the case of compensating module failures:

$$\varphi''(\underline{x}'') = \varphi''(0,x_2,x_3,x_4,x_5,x_6,0,x_8,0,x_{10},\ldots,x_{33}) = 1 .$$

But $x'' < x'$ while $\varphi'' > \varphi'$. The structure function is not coherent.

The reliability that will be calculated in later sections of this paper will be understood to be the reliability of the system with respect to sets of failed components, none of whose subsets could cause system failure. If subsequent component failures, such as in φ'' above, restore the network to a properly working state (i.e., the structure function is 1) the network will still be considered failed since it was capable of producing an erroneous output (under structure function φ' in our example) before the extra components failed. This forces the structure function to be monotone increasing and is the same as assuming that the first erroneous output signifies permanent system failure.

When reliability modeling is independent of module design, the x_i's of the structure function will represent modules or voters. When the internal design of the module is considered, as in [15], each x_i represents a lead.

With this concept of what constitutes a system failure, we will now consider methods for modeling the reliability of a TMR network.

2.4 Calculating Serial TMR Reliability

The serial cell reliability modeling technique will be demonstrated for some simple systems. The resultant reliability model will be compared to the reliability model to be presented in this paper. It will be shown that for these simple systems the predicted mission time can be increased by 50% just by using the more accurate model. An indication of why the serial cell technique is not always accurate will also be given.

First consider Fig-. 4 which graphically depicts the required definitions. A module trio is a group of three replicas of the non-redundant system module. All members of a module trio are identical.. A voter trio is a group of three voters whose inputs come from a module trio, In Fig, 4 modules (1, 2, 3), (4, 5, 6) and (10, 11, 12) form module trios. Module 1 is said to feed voter 7, 8 and 9 while module 10 is driven by module 4 and voter 7.

A cell is a portion of a network such that all the modules in the cell are fed by voters in the cell or by network inputs and all voters in the cell feed modules in the cell or network outputs. Fig. 4 shows three cells. Cell 1 is known as a module end cell and cell 3 as a voter end cell. Finally, a level is a vertical partitioning of a network which contains voters only, or modules only. It is somewhat analogous to the combinational logic concept of levels. Fig. 4 has four levels,

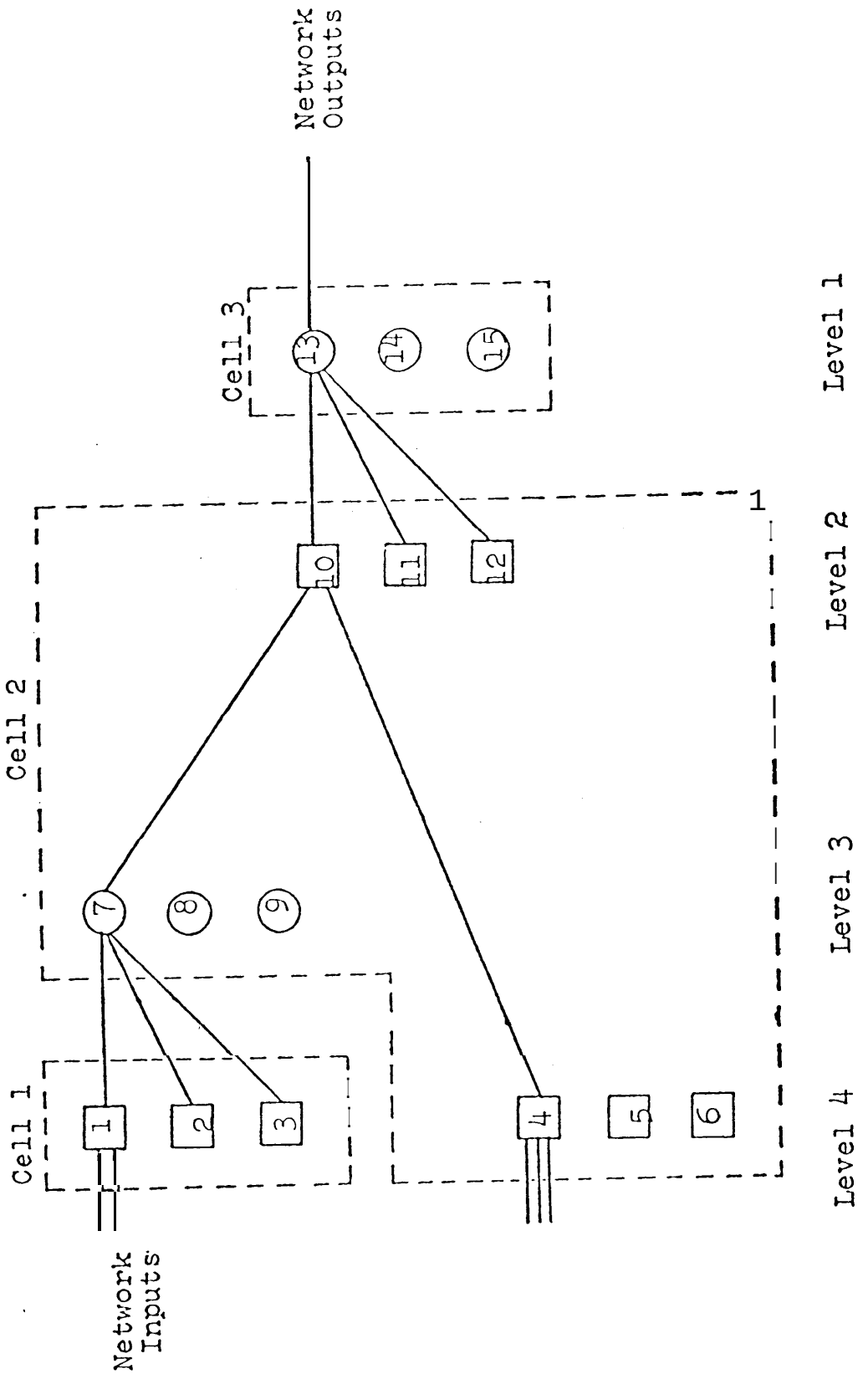


Fig. 4. A TMR network

The serial cell technique attempts to model all networks as a cascade of serial cells, Consider the serial cell of Fig. 3. Assume that the voters never fail, Then there exists four states or module failure patterns for which the system still realizes its design function. They are (1) no module failures, and (2) three states, each of which has a single module failure (the two remaining working modules will realize the design function and form a majority regardless of the behaviour of the failed module), Thus the cell reliability derived by summing over all the working states is given by:

$$R_{cell} = R_m^3 + 3R_m^2(1 - R_m)$$
$$= 3R_m^2 - 2R_m^3 \qquad (1)$$

A voter failure has the same effect as a module failure so replacing R_m by $R_m R_v$ in (1) yields:

$$R_{cell} = 3(R_m R_v)^2 - 2(R_m R_v)^3 \qquad (2)$$

To alleviate the ambiguity in the structure function when the system components are taken to be modules or voters, as illustrated in the section on coherency it will be assumed that all module failures in a trio are identical. Thus any two module failures in a trio would outvote the good module and cause system failure. This will lead to a worst case reliability model since all module failures need not be identical. In the calculations to follow we will ignore the casesof compensating module failures as described in [15] and consider only a worst case reliability function. We will also consider network

configurations more complex than serial cells, such as cells exhibiting fan-in and fan-out.

The use of serial cell reliability modeling for networks that exhibit fan-in and/or fan-out can lead to serious errors in estimates for overall system reliability. Yet such networks are fairly common candidates for the application of TMR.

Consider the 16 register multiplexed data bus system and ALU of Fig. 5 which might use TMR on a long space mission. The data register transfer block, block 1 in Fig. 4, exhibits fan-out. The contents of the data register can be supplied to any one of the 16 general purpose registers. Block 2, the ALU to multiplexer transfer, represents fan-in. The results of any one of 16 ALU operations is selected by the 16 multiplexers for transmission to the data registers.

Figure 6 shows a TMR configuration of Block 1, the data register to register transfer. One approach to handle fan-in/out in the serial cell reliability model is to assign the voters to the modules they drive [26] since a voter failure affects only the module it drives. Thus cell 2 in Fig. 6 shows one way to assign the voters to the driven modules. Now the serial cell reliability model will be developed.

The reliability of a module end cell such as cell 1 can be derived from (2) by letting $R_v = 1$. Similarily setting $R_m = 1$ in (2) yields the reliability of voter end cells such as cell 3; Next assume

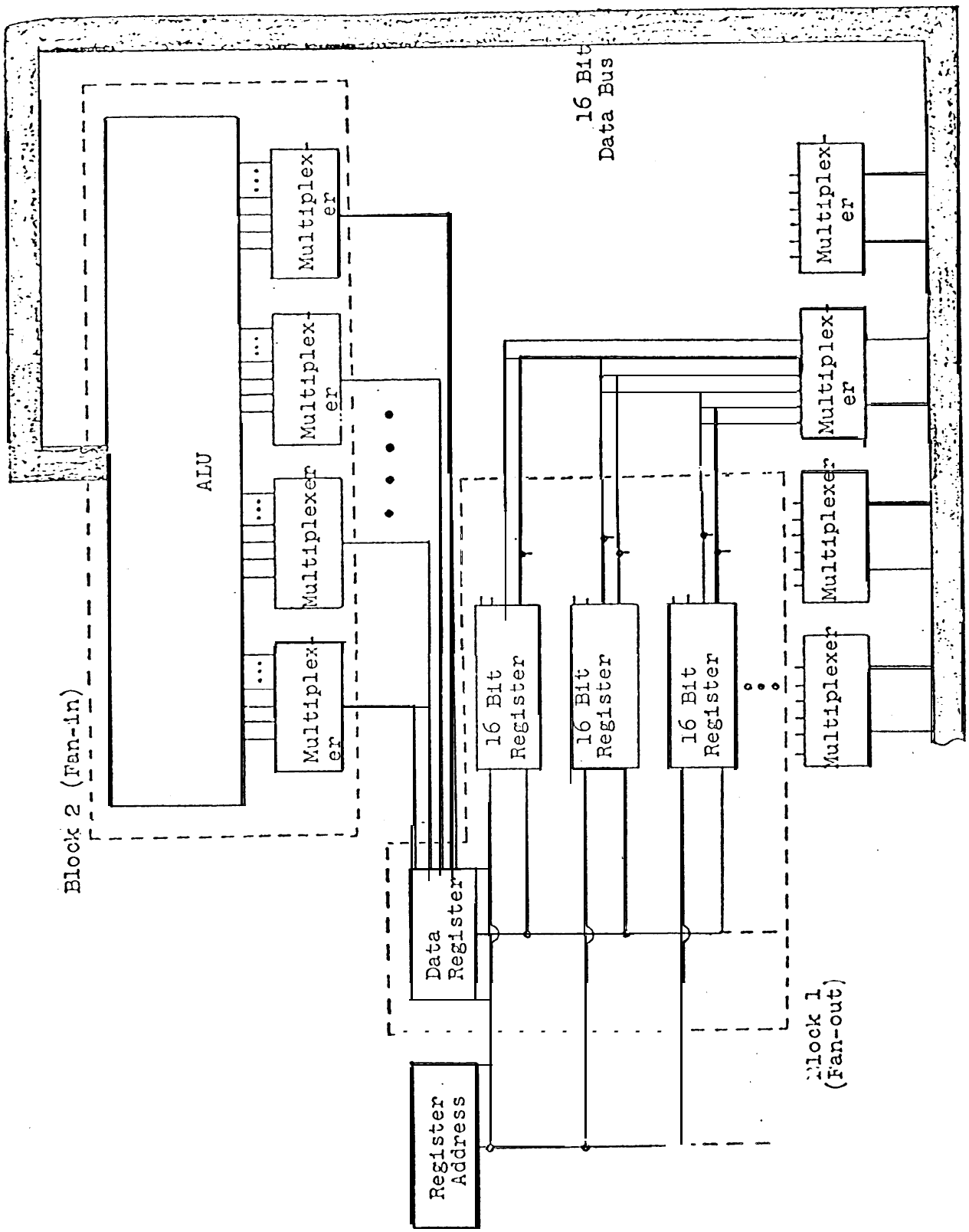


Fig. 5. Small computer Data Bus - ALU system.

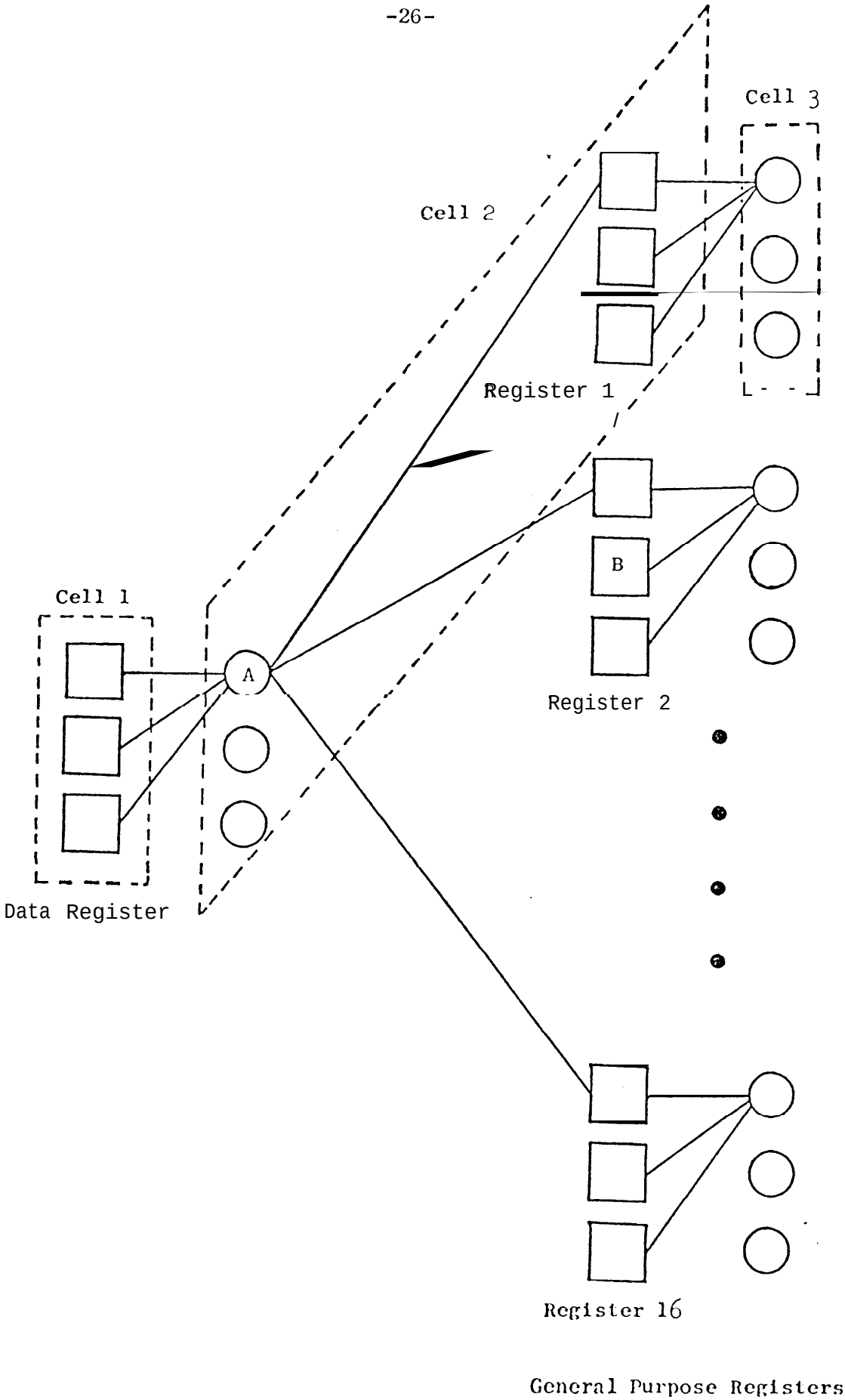


Fig. 6. The TMR configuration for one bit of the Data Register to Register fan-out block. Only one TMR path is shown.

$R_m = R_v = R$. Lyons [24] has shown that the maximum system reliability is obtained when modules and voters are on the same order of complexity, i.e., $R_m = R_v$. This will yield the network partitioning for the maximum obtainable system reliability, This simplification is not crucial and similar results are obtained when R_v and R_m retain their seperate identities, as will be demonstrated later. The end cell reliability is thus $3R^2-2R^3$. The serial cell reliability model for the system of Fig. 6 would consist of 17 end cells (16 voter and 1 module) and 16 serial cells, like cell 2, each of which share the one voter trio. The system reliability is thus modeled by,

$$R_{serial} = (3R^2-2R^3)^{17} \ (3R^4-2R^6)^{16} \ . \qquad (3)$$

Thus the model calculates the reliability of a corresponding system which replaces the fan-out voter trio by 16 voter trios.

The system reliability developed by techniques in this discussion, (which is known to be a lower bound), for the network of Fig. 6 is plotted with (3) as a function of module reliability in Fig. 7 and their difference is plotted in Fig. 8. In actual design situations a mission time, i.e., the desired operating life of the system, would be selected. This would a numeric value for module reliability which is then substituted for R_m in the equations developed by the modeling techniques. Note that here, we are only interested in the variation of the predicted system reliability for each modeling technique as a function of module reliability. Plots of the form of Fig. 7 and Fig. 8 were chosen to display this variation. We are comparing modeling

techniques and not redundant system desings. There are better methods for comparing redundant system designs presented in the literature, [3].

Various reliability models for the same redundant system may predict widely varying system capabilities. For example consider one interesting parameter for comparing redundant system designs, namely, mission time improvement, I [3]. Assume an exponential failure distribution, i.e., $R_m = e^{-\lambda_1 t_1}$ and $R'_m = e^{-\lambda_2 t_2}$. The reliability model for the two redundant systems is derived. A value for R_m is assumed and substituted in one equation. Then an R'_m is calculcated such that the two system reliabilities are identical. If we represent Rm by $R_m = R'^{I}_m$ then $\lambda_1 t_1 = \lambda_2 t_2 I$. Further if $\lambda_1 = \lambda_2$ then $t_1 = It_2$ and design one has the same system reliability at t_1 as design two does at time It_2. The mission time improvement is defined as I. This parameter can also be used to compare reliability models, Fig. 9 shows a plot of mission time improvement when system one is the serial cell model. It can be seen that a mission time improvement of 50% can be obtained by using a more accurate reliability model. Another way of looking at the parameter I is that if the serial cell model is used then the resultant system is overdesigned by 50% since it could meet its mission time specification with less reliable components. Alternatively it could use the same component reliability and contain 50% more components and still meet the specifications.

The source of the variation between the two techniques as displayed in Fig. 7, 8 and 9 lies in the serial cell approach assuming that the voter trio, which is the origin of the fan-out is replaced by sixteen

voter trios, one for each module trio the original voter trio feeds. Equation 3 counts failure patterns for which the network fails. As an example of the inclusion of a pattern for which the network fails, consider the failure of voter A in Fig. 6. If the failure were assigned to the serial cell marked 2 then module B could fail and the serial cell approach would predict correct system functioning. Yet the system has failed, since the second trio of fan-out modules containing module B could produce two incorrect signals, one due to module B failing and one due to the failure of voter A.

The serial cell approach is also pessimistic in the sense that it penalizes the network for components it does not have. In Fig. 6 the serial cell approach models the system by another system which has sixteen trios of voters instead of one fan-out trio. The actual system does not include this extra hardware and hence is penalized by the unreliability of the extra voters. The interaction of these two effects is very complex and it is very difficult to determine which one, if any, dominates in an arbitrary network. This is why we cannot say whether the serial cell approach is an upper or lower bound on actual network reliability.

In the case of fan-in, such as the ALU multiplexer block, (3) also applies if the serial cells are assigned as in Fig. 10 and it is assumed there are sixteen ALU functions to select from, The system reliability for this configuration, as derived using techniques to be developed in the next section, is also plotted in Fig, 7 and Fig. 8.

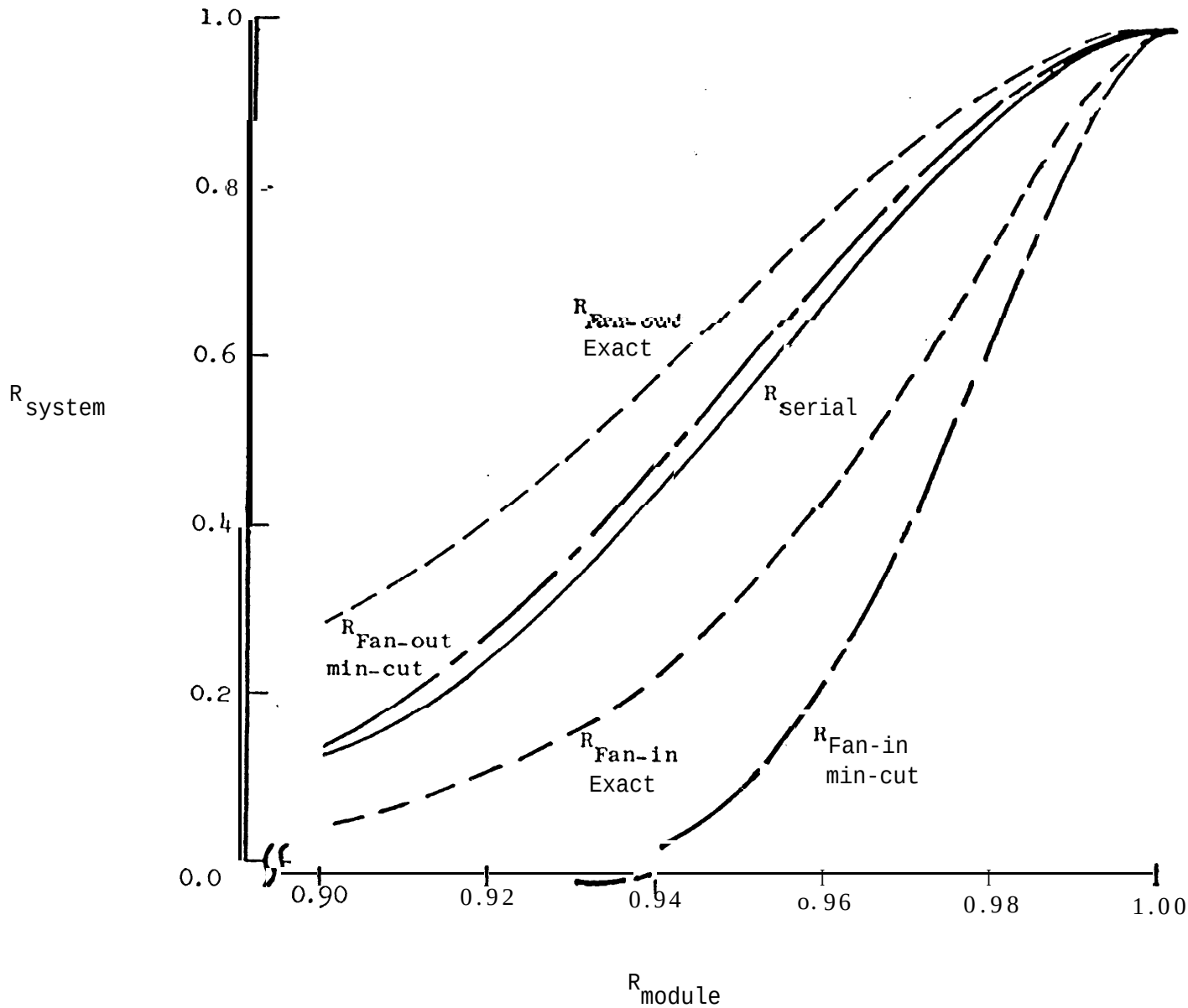


Fig. 7. System reliability as a function of module reliability for the fan-out network of Fig. 6 and the fan-in network of Fig. 10. The serial cell approximation to both networks is identical and plotted as the solid line.

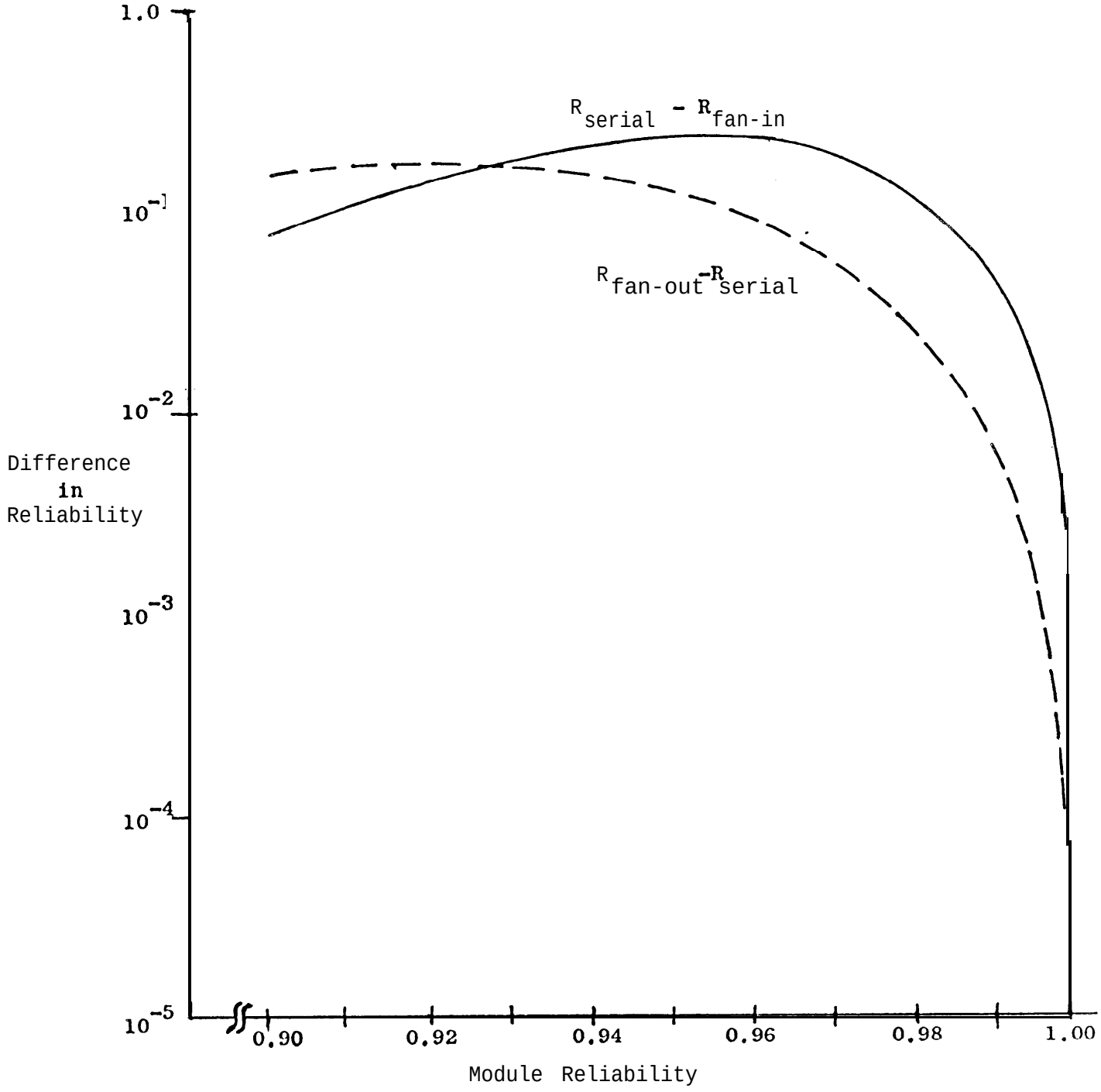


Fig. 8. The difference in system reliability as calculated by the techniques of this discussion and the serial cell approximation for the fan-out network of Fig. 6 and the fan-in network of Fig. 10.

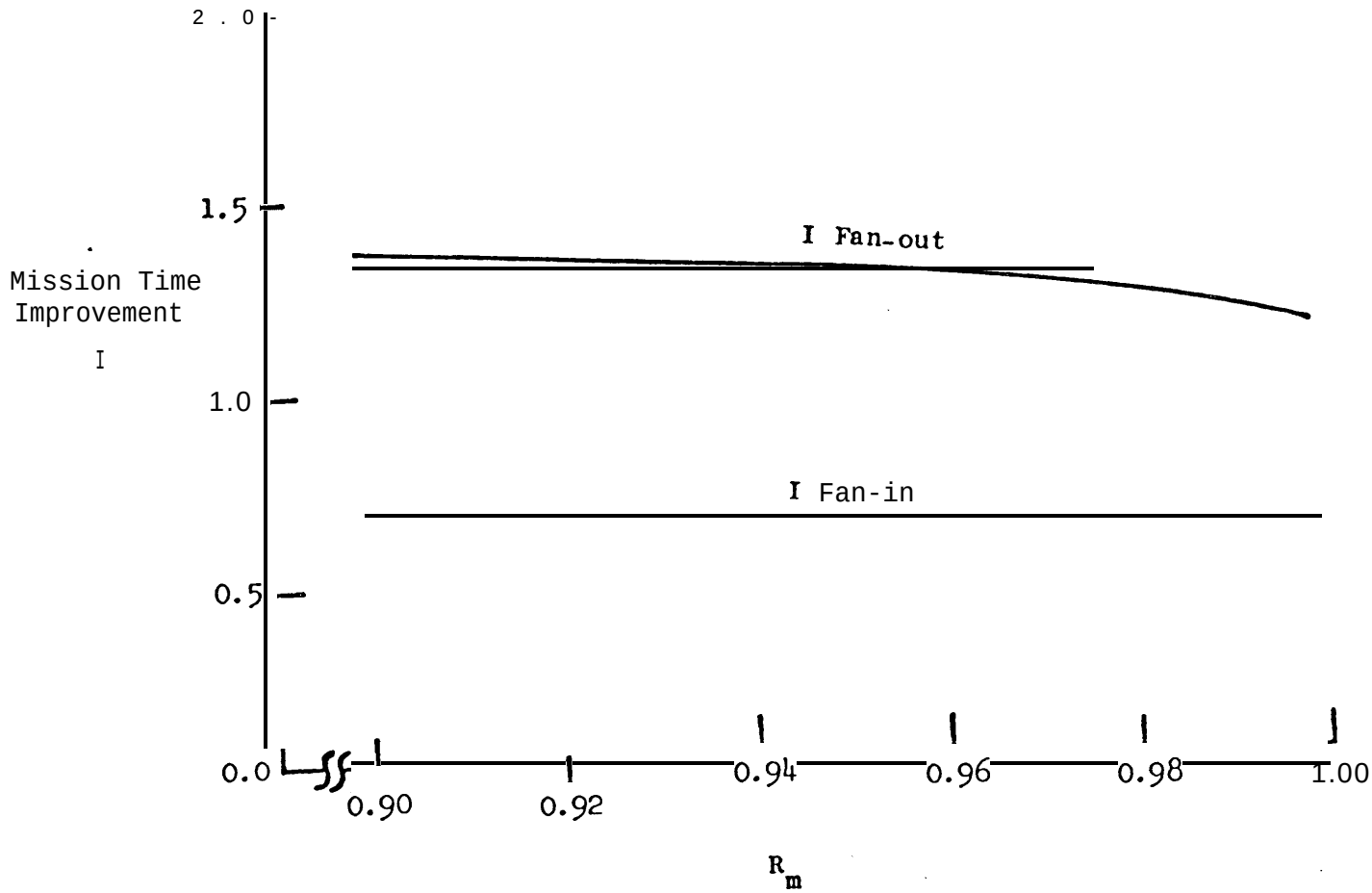


Fig. 9. Mission time improvement obtainable when utilizing the more accurate reliability model for the networks of Fig. 6 and Fig. 10.

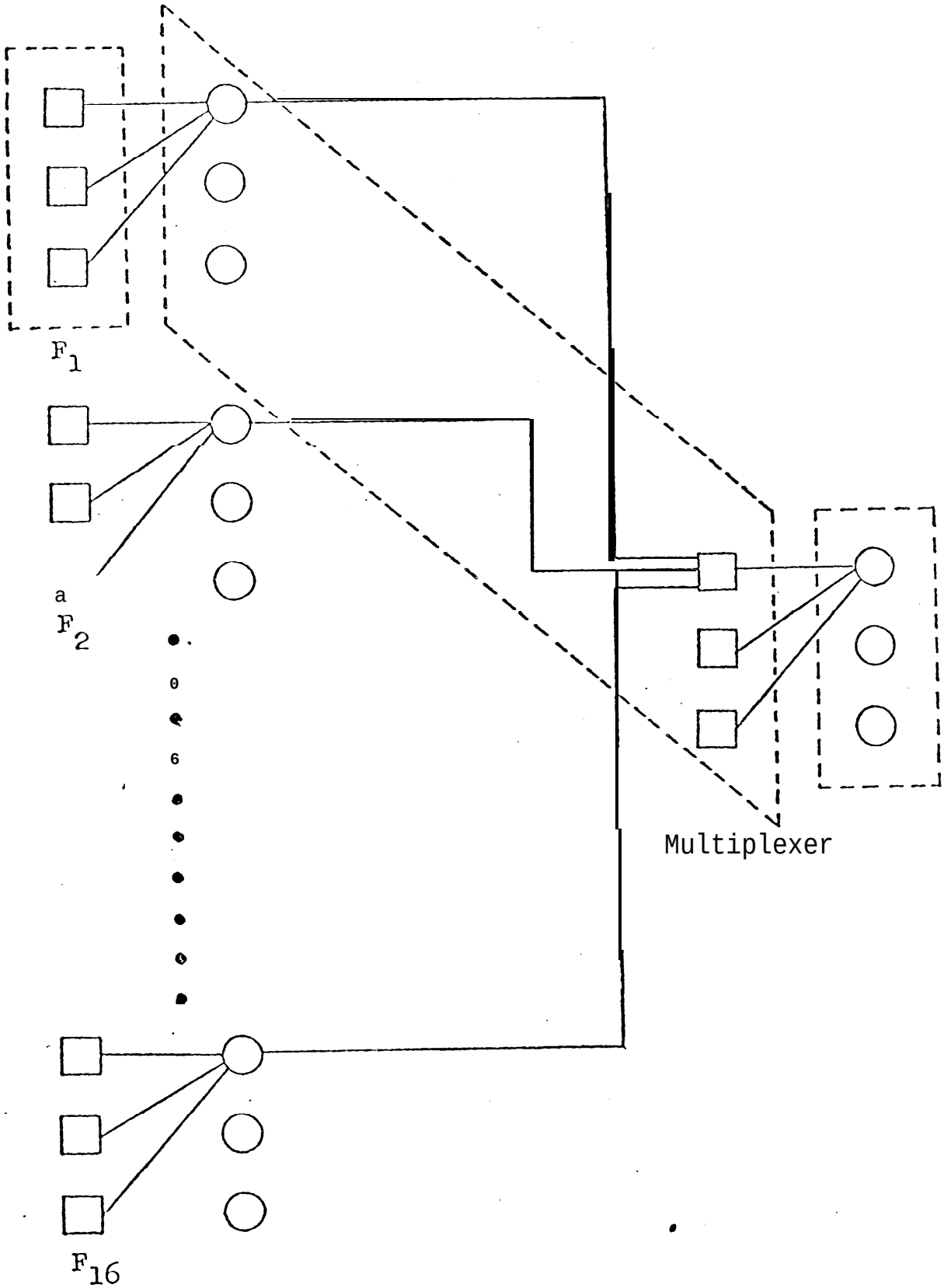


Fig. 10. The TMR configuration for one bit of the ALU to Multiplexer fan-in block. The ALU performs 16 functions. Only one TMR path is shown.

Now (3) is optimistic. Fig. 9 shows that the system has only 50% of its designed mission time. So the serial cell approach may not be an upper nor a lower bound to system reliability and for $R_m = 0.95$, Fig. 8 shows that it may not even be a "good guess",

In designing and comparing redundant systems, a good predictive technique for system reliability is needed, not one which is merely a guess and might be high or low depending on the network it modeled. If the exact reliability is too difficult to find then a lower bound is desirable. A technique which gives the exact reliability, and if desired, a tight lower bound in return for a saving in time is presented in the next section.

3. CALCULATING THE EXACT COHERENT RELIABILITY OF A TMR NETWORK

3.1 Introduction

The algorithm we present will calculate the exact reliability assuming TMR is a coherent system. The basic assumptions when treating TMR as a coherent system are [5].

1. Once a module or voter has failed it will always give an incorrect output.
2. Once a module has a failed input its output is also failed.

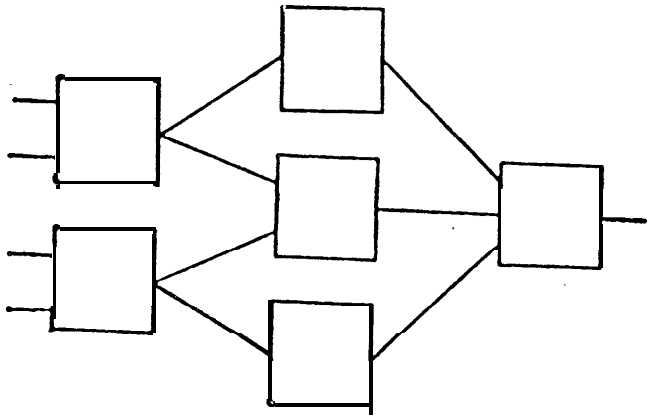
It should be noted that TMR is not a coherent system when considering failure modes other than complete failure. For example, one input to a voter could be stuck-at-l and another stuck-at-O. Since two voter inputs have failed the system has failed by the coherent system assumption. In actuality the system functions correctly. These compensating failures can be incorporated into the reliability model at the expense of more computation time. The coherent system reliability calculated is thus a lower bound on actual system reliability.

For the remainder of this discussion the reliability of a system (module) will mean the conditional probability that the system (module) will be capable of performing its specified function at time t given that all system (module) components are functioning properly at time t = 0,

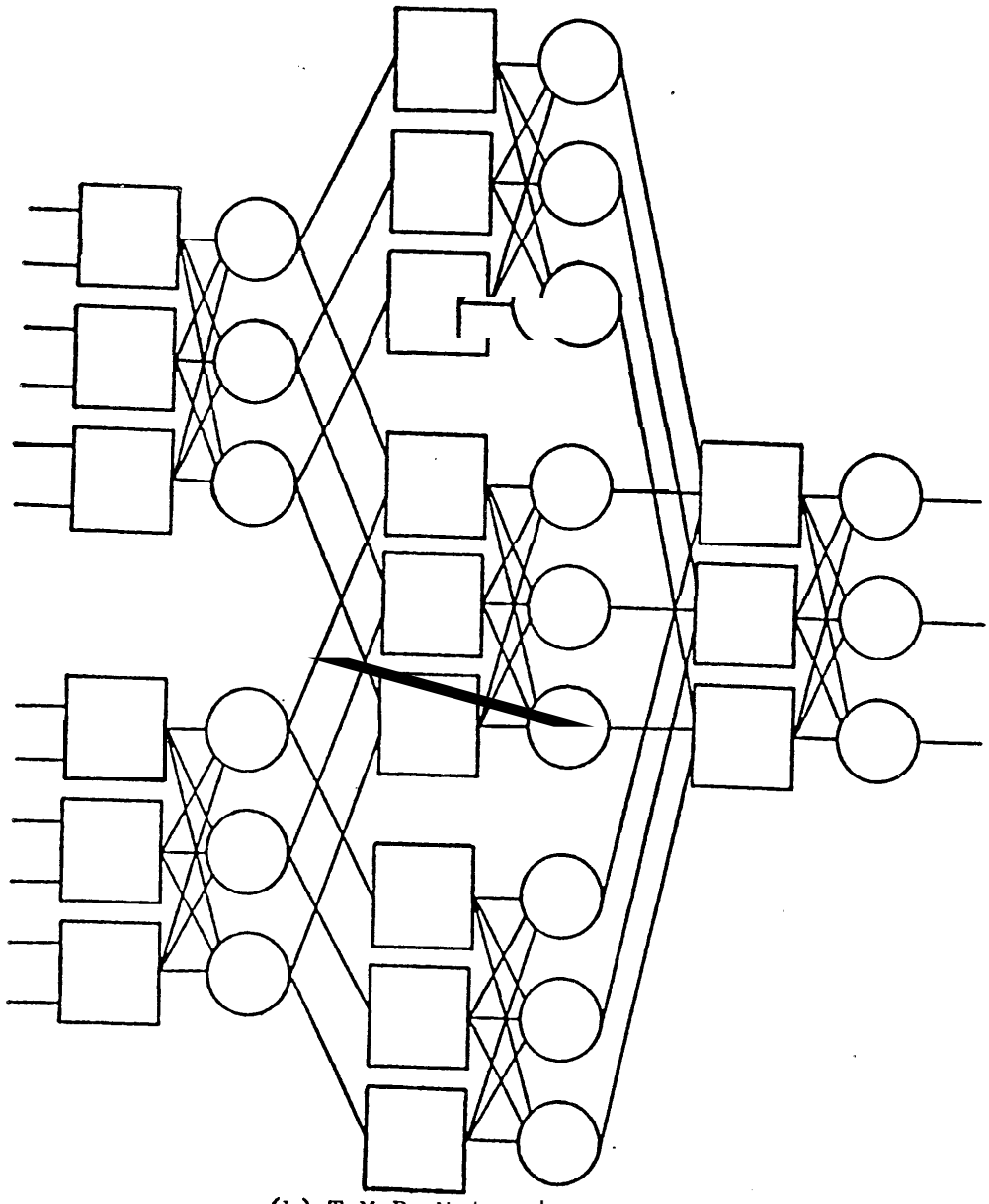
In the subsequent formulas time is an implicit variable. To calculate system reliability at time t the module reliability must be evaluated at time t.

Our approach is to partition an arbitrary TMR network into independent "cells" so that a failure in one cell cannot combine with a failure in another cell to cause system failure. The reliability of each cell is found and the reliability of the whole network is found from the cell reliabilities. This is much simpler than finding the reliability of the whole network at one time. If there are N modules in a network which can be partitioned into n independent cells of m modules each. where $N = m \cdot n$, and if the complexity of the reliability evaluation algorithm is a function ψ of the number of modules, it is easily seen that $n \cdot \psi(m) >> \psi(m \cdot n)$, especially when ψ is exponential, as is usually the case. Also, this method is a specialized one for TMR, and takes advantage of the known properties of TMR.

Consider Fig. 11 where the non-redundant network (a) and its TMR counterpart (b) are depicted. Each of the triplicated modules or voters will be referred to as a module or voter trio, and each module or voter in a trio is said.to occupy a particular position in the trio. It is to be noted that the modules need not be a single output module, and that there need not be voters after every module trio. System failure in a TMR system occurs when there are two or more errors in any of the (triplicated) output lines. Under assumption (1), system failure will occur if any of the module or voter trios have more than one failed module or voter. Assumption (2) implies that system failure can also occur if more than one module or voter in a trio has a wrong input or if one module in a trio is failed and another has a failed input. The reliability of a network is then the probability that system does not have one of these failure modes.



(a) Unredundant Network



(b) T.M.R. Network

Fig. 11. Network showing T.M.R. structure and division into cells.

3.2 Partitioning a Network into Cells

We will now discuss the partitioning of an arbitrary network which simplifies the task of reliability evaluation.

In a TMR network a voter or module trio p is defined to be directly connected to a voter or module trio q if a single fault in a particular position of p allows only the single fault in the corresponding position of q without causing system failure. In Fig. 12 for example, if a sirrgle fault occurs in voter trio p - say the voter marked x has failed - then only one of the three modules in module trio r (the one marked x) can fail without causing system failure. Therefore, p is directly connected to r. Similarly, q is directly connected to r, and p is directly connected to q. On the other hand, neither p, q, or r is directly connected to s. We denote the relation "is directly connected to" by D. Clearly, D is a symmetric relation. Further, we define that every trio is directly connected to itself, i.e., D is reflexive.

For any set of trios in a TMR network, two trios p and r are defined to be connected if there exists a sequence of trios in the set (possibly a null sequence) $q_1, q_2, \ldots, q_n$ such that

$$p \; D \; q_1, q_1 \; D \; q_2, \ldots . q_n \; D \; r.$$

Let C be the relation "is connected to". It is obvious that C is an equivalence relation.

Therefore, an arbitrary TMR network can be partitioned into equivalence classes using the relation C. We call these equivalence classes cells. As an example, Fig. 13 shows a TMR network with the cells enclosed in dotted lines. The trios within a cell which feed trios in other cells or network outputs are known as cell output trios.

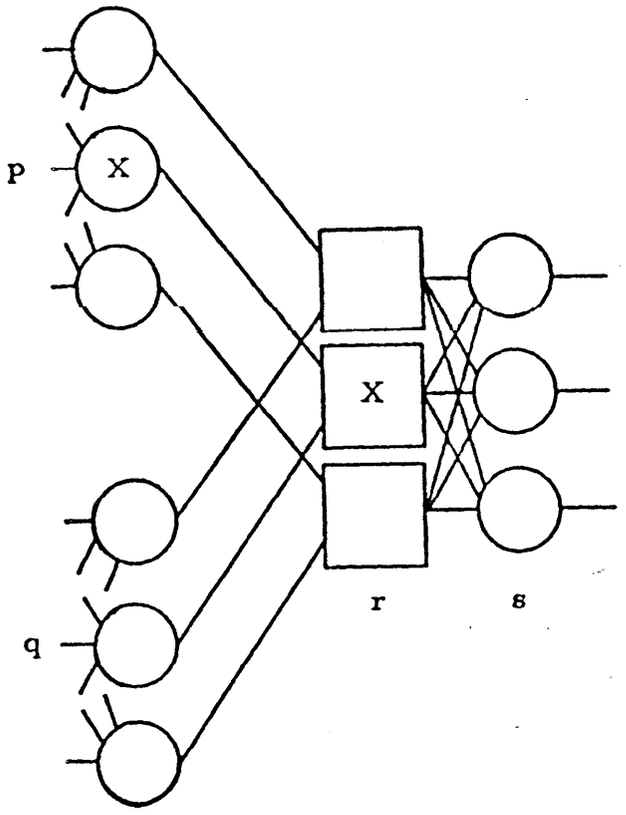


Fig. 12. A portion of a redundant network.

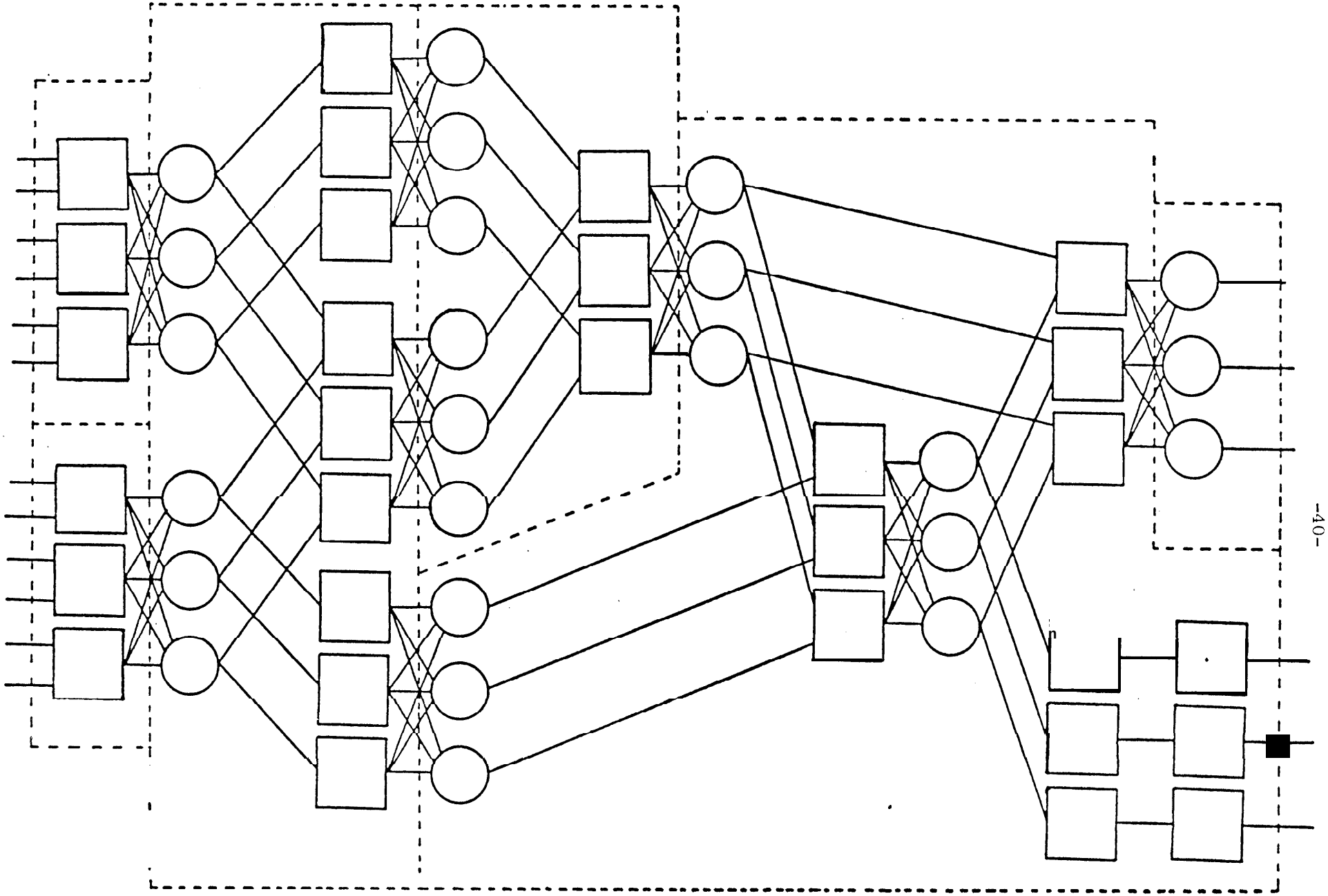


Fig. 13. Partitioning a redundant network into cells.

Any single error at an output trio of a cell will be corrected by the input (voter) trio of the next cell, while two or more errors will result in system failure. Therefore the cell reliability is defined as the probability of at most one error at each output trio of the cell. The network reliability is then the product of all the cell reliabilities.

3.3 Assumptions and Definitions used in the Algorithm

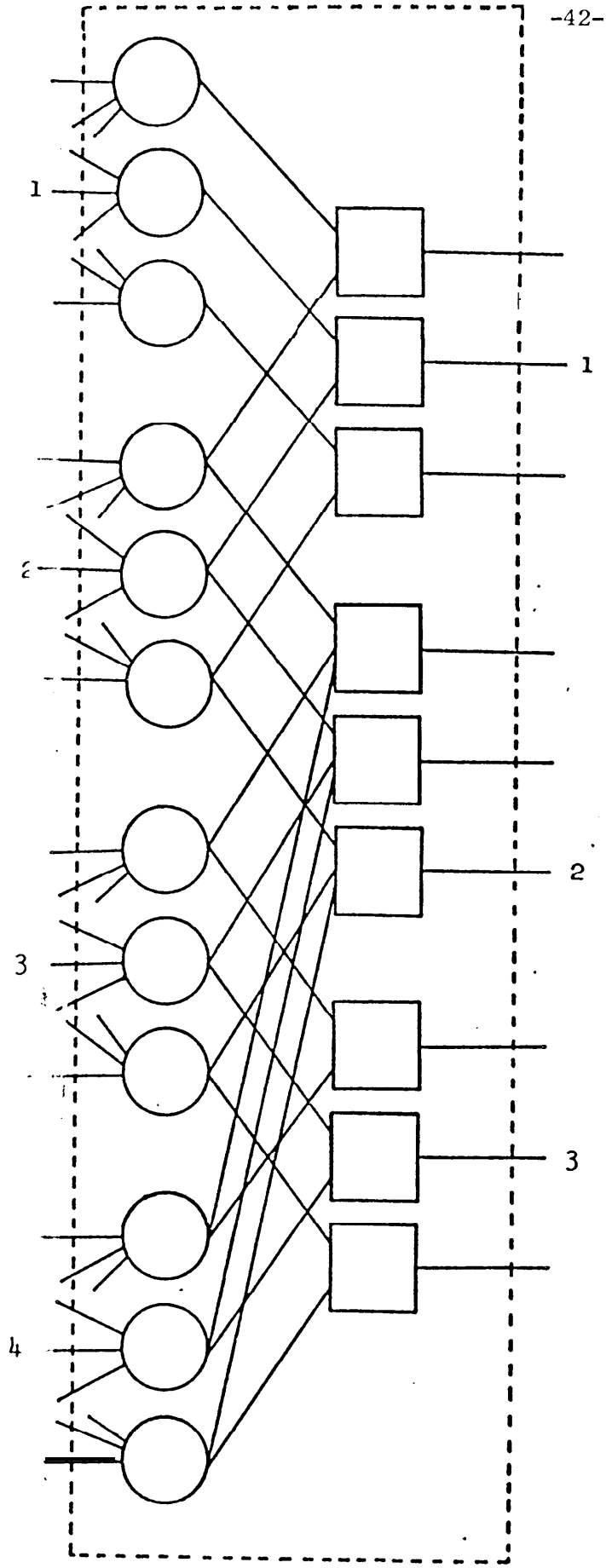
To simplify the explanation of the algorithm, only networks with single output modules and voters following all the modules will be used. For the present, we will also assume that all the modules in a cell have the same reliability. The algorithm can be readily extended to include more complex cases, as will be discussed briefly later.

The cell shown in Fig, 14 (a) will be used as the example to illustrate the algorithm. Let N_v and N_m be the number of voter and module trios in a cell respectively. In the example, $N_v = 4$, and $N_m = 3$.

The Structure Matrix, $\underline{S}$, of a cell is defined as follows. This matrix can be written down from inspection of the cell, and indicates which voter trios of the cell have paths to which module trios, Each of the voter and module trios is numbered arbitrarily, the voter trios from 1 to N_v, and the module trios from 1 to N_m. In Fig. 14 (a), the voter trios are numbered from 1 to 4, and the module trios from 1 to 3. The Structure Matrix S is then defined to be an $N_v \times N_m$ matrix such that

$S(i,j) = 1$, if there is a path from voter trio i to module trio j

$= 0$, otherwise.



(a). Cell of redundant network

	1	2	3
1	1	0	0
2	1	1	0
3	0	1	1
4	0	1	1

(b). Structure Matrix of the cell in (a)

Fig. 14. Cell used in the example.

The Structure Matrix of the cell of the example is thus obtained in Fig. 14 (b). For example, there is a path from voter trio 1 to module trio 1, but no path from voter trio 1 to module trios 2 and 3. Therefore, $S(1,1) = 1$ but $S(1,2) = S(1,3) = 0$. The other rows are obtained in a similar manner.

The Fault Matrix, $\underline{F}$, of a cell is defined as an $(N_v + 1) \times (N_m - 1)$ matrix, where $F(i,j)$ is the number of exactly i voter faults and j module faults that the cell can have and yet remain reliable, i.e., produce at most one error at each output trio. If F can be obtained, then calculating the reliability of a cell is a simple matter, since F enumerates all possible fault patterns that the cell can tolerate.

Given a set with N elements, a combination of i elements is defined as one of the $\binom{N}{i}$ subsets of N with i elements. A combination of trios can be further partitioned into equivalence classes generated by C, and these are called groups.

For a combination of i voter trios in a cell, G_v is defined as the number of ways in which i voter failures (one from each trio) can occur without causing system failure. Suppose these i voter trios can be partitioned into n groups. Each voter in a group is connected to the other voters in the group, and so the voters in a group can fail in only three ways. Then, for this particular combination, $G_v = 3^n$, since the groups are elements of a partition. From the cell of Fig. 5(a), consider the combination of three voter trios, (1,3,4). There are two groups, ((1), (3,4)), and $G_v = 3^2 = 9$.

For a combination of voter trios, L is defined as an N_m length binary vector such that,

L(j) = 0, if and only if there is no path from any voter trios in the given combination to module trio j

= 1, otherwise.

For the combination of voter trios (1,2) for example, L = 110, since there is no path from voter trios 1 and 2 to module trio 3.

For a combination of i voter and j module trios, G_m is defined as the number of ways in which j module failures (one from each module trio) can occur, given that i voter failures have occurred, without causing system failure. All the modules to which the i voter trios have paths can fail in only one way, while each of the module trios in the set of j module trios which are not connected to the i voter trios can fail in three ways. If the number of such module trios in the second set is m, then, $G_m = 3^m$. From the definition of L it can be seen that if we take the L corresponding to the combination of i voter trios, m is the number of zeros in the positions of L corresponding to the j module trios. For the voter trio combination (1,2) which has an L = 110 and a module trio combination (2,3) for example, the number of zeros in positions 2 and 3 of L is 1, thus m = 1.

3.4 Algorithm to Calculate the Reliability of a Cell

The algorithm to be described generates the Fault Matrix directly from the Structure Matrix of the cell. Table 1 gives the development of the algorithm for the cell of Fig. 14(a) and Table 2 is the Fault Matrix of the cell.

If no voters fail, the modules can fail independently, one module from each trio. The number of ways in which j modules can fail is then given by the number of ways of choosing j out of N_m trios, multiplied by the number of ways j modules can be chosen from the j trios, so that,

$$F(0,j) = \binom{N_m}{j} 3^j, \qquad j \geq 0 .$$

This gives the first row of the Fault Matrix in Table 2.

Consider $F(i,0)$, $i > 0$, which is the total number of ways in which i voters and 0 modules can fail, If we take any combination of i voter trios, the number of ways in which i voter failures can occur is given by G_v. Therefore the total number of ways in which i voter failures and 0 module failures can occur is the sum of G_v over all the possible $\binom{N_v}{i}_0$ combinations,

For each combination, the partition into groups can be made in many ways, but one way quite attractive for programming on a digital computer is the following. If two voter trios i_1, i_2 are directly connected, then the rows of the Structure matrix corresponding to them (rows i_1, i_2) will both have a 1 in the same position, and the AND of the two rows will not be the 0 vector. (A logical binary operation on two vectors is carried out by performing the binary operation on corresponding bits of the two vectors). They then belong to the same

Voter combinations		G_v	L	Module combinations - G_m								
				1 module			$\Sigma_m G_m$	2 modules			$\Sigma_2 G_m$	3 modules
				(1)	(2)	(3)		(1,2)	(1,3)	(2,3)		(1,2,3)
1 voter	(1)	3	100	1	3	3	7	3	3	9	15	9
	(2)	3	110	1	1	3	5	1	3	3	7	3
	(3)	3	011	3	1	1	5	3	3	1	7	3
	(4)	3	011	3	1	1	5	3	3	1	7	3
2 voters	(1,2)	3	110	1	1	3	5	1	3	3	7	3
	(1,3)	9	111	1	1	1	3	1	1	1	3	1
	(1,4)	9	111	1	1	1	3	1	1	1	3	1
	(2,3)	3	111	1	1	1	3	1	1	1	3	1
	(2,4)	3	111	1	1	1	3	1	1	1	3	1
	(3,4)	3	011	3	1	1	5	3	3	1	7	3
3 voters	(1,2,3)	3	111	1	1	1	3	1	1	1	3	1
	(1,2,4)	3	111	1	1	1	3	1	1	1	3	1
	(1,3,4)	9	111	1	1	1	3	1	1	1	3	1
	(2,3,4)	3	111	1	1	1	3	1	1	1	3	1
4 voters	(1,2,3,4)	3	111	1	1	1	3	1	1	1	3	1

Table 1.

Development of the Algorithm.

modules

voters	0	1	2	3
0	1	9	27	27
1	12	66	108	54
2	30	102	114	42
3	18	54	54	18
4	3	9	9	3

Table 2.

Fault Matrix of the Cell in Fig. 5.

group. The OR of the two vectors is found and this is compared to the rest of the rows; all rows not giving a zero vector when the AND operation is performed correspond to voter trios belonging to the same group, and they are all OR-ed together. This process is continued until the combination has been partitioned.

Table 1 shows the result for the cell of Fig. 14(a). Every combination of $i = 1,2,\ldots,N_v$ rows of the Structure Matrix S is taken and the value of G_v found for each, The vector L corresponding to a combination is the OR of all the rows of S corresponding to the combination. The 0^{th} column of F is then obtained from,

$$F(i,0) = \sum_{\substack{\text{all combinations} \\ \text{of i rows of S}}} G_v , \qquad i > 0$$

To find F(2,0) in the example we have to take the six possible combinations and sum the value of G_v for these, which gives 30. For the combination (3,4), the vector L is the OR of rows 3 and 4 of S, and is equal to 011.

Now consider $F(i,j)$, $i,j > 0$, which is the total number of ways in which i voters and j modules can fail without causing system failure. Given a combination of i voter trios, G_m is the number of ways in which j module failures can occur in a combination of j module trios. The number of ways in which i voters and j modules can fail for any given combination of i voter and j module trios is then $G_v \cdot G_m$, and the total number of ways in which i voters and j modules can fail is the sum of $G_v \cdot G_m$ over all such combinations of i voter and j module trios. Thus for every combination of voter trios, we take every possible combination of $j = 1,2,\ldots,N_m$ bits of L, and for each of these,

$G_m = 3^m$ where m is the number of zeros in that combination of bits of L. This is shown in Table 1.

Taking the example again, consider the vector L of voter combination (1,2), which is 110. For module combination (1,3) the number of zeros in the positions 1 and 3 of L is 1, and G_m for this combination is $3^1 = 3$, but for module combination (1,2) there are no zeros in those positions, and G_m for that combination is $3^0 = 1$. F(1,3) is given by 3'9 + 3·3 + 3·3 + 3.3 = 54.

Thus the rest of the entries of the Fault Matrix are,

$$F(i,j) = \sum_{\substack{\text{all combinations of i rows of S; all combinations of j digits of L corresponding to the i combinations}}} G_m \cdot G_v, \quad \begin{matrix} i > 0. \\ j > 0. \end{matrix}$$

If all the modules of a cell have the same reliability, we do not need the separate entries of G_m but only the sum. In that case, if L has m zeros in it, G_m for j module combinations is

$$\sum_{\substack{j \text{ module} \\ \text{trios}}} G_m = \sum_{k=0}^{j} \binom{m}{j-k} \cdot \binom{N_m - m}{k} \cdot 3^{j-k}$$

It is assumed here that when k is negative or greater than n, the binomial coefficient $\binom{n}{k}$ is zero. The above expression is obtained by considering a particular combination of j digits of L. If it has k l's and j-k O's in it, these can be arranged in $\binom{m}{j-k} \cdot \binom{N_m - m}{k}$ ways, and for each of the arrangements, the value of G_m is 3^{j-k}. We then sum over all possible values of k.

The reliability of the cell is then given by,

$$R_{cell} = \sum_{i=0}^{N_v} \sum_{j=0}^{N_m} F(i,j) \cdot R_v^{3N_v - i} \cdot (1-R_v)^i \cdot R_m^{3N_m - j} \cdot (1-R_m)^j$$

where R_v and R_m are the reliabilities of a single voter and a single module respectively.

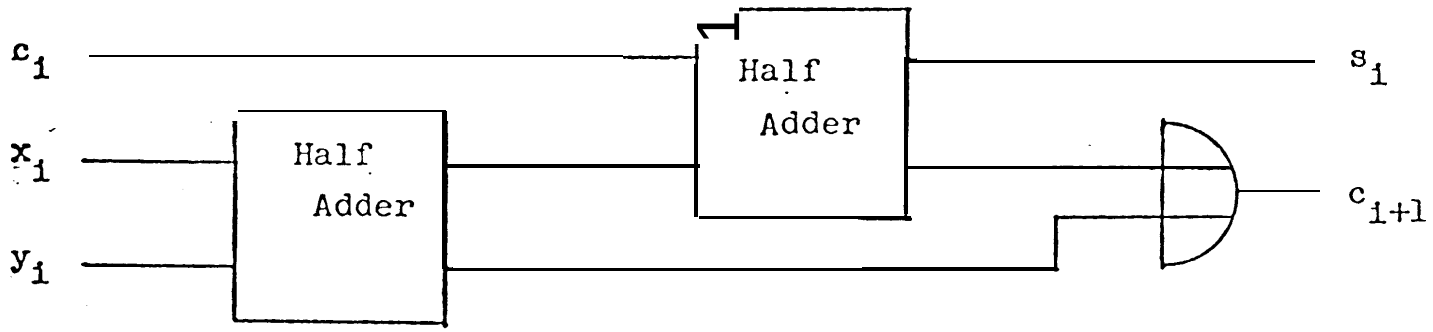
3.5 An Example

Fig. 15(a) shows a block diagram of a full adder, while (b) is one possible NAND implementation of it. Fig. 16 is a TMR version of the NAND gate realization with one data path sketched in. The reliability of the TMR network was calculated both by the algorithm given in the previous section, and by the serial cell approximation, for comparison. In order to get a better idea of the difference between the two methods, Klaschka's "reliability improvement index" [4] was used as the basis of comparison. This is a ratio of the logarithms of the nonredundant and redundant reliabilities, and is given by,

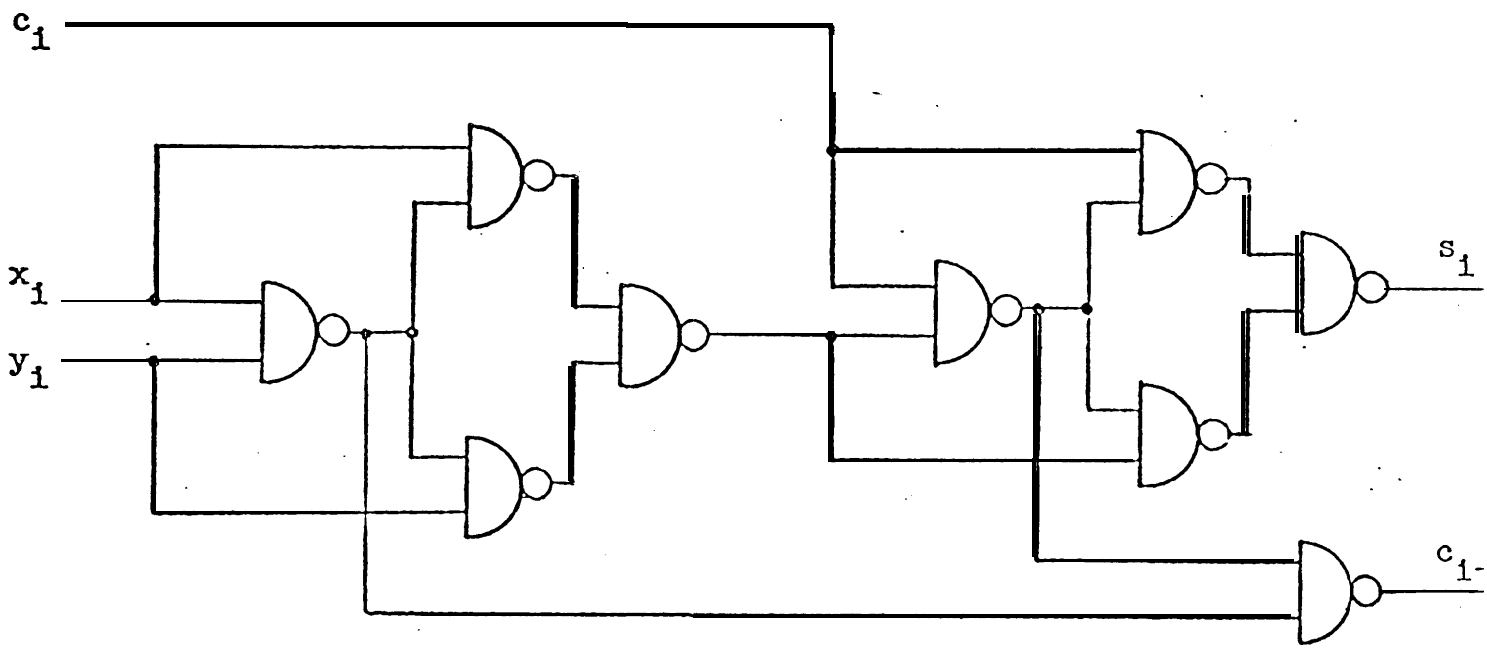
$$\text{Reliability improvement index} = \frac{\log\left(R_{\text{non-redundant}}\right)}{\log\left(R_{\text{redundant}}\right)} .$$

This index gives a better idea of the improvement in reliability obtained by using the redundancy scheme [4].

Fig. 17 shows the comparison for the example. Here, the reliability improvement index is plotted against module failure probability, for a fixed voter failure probability. As can be seen from the graph, the algorithm described gives a much better lower bound to the reliability. The improvement increases as the modules



a)



(b)

Fig. 15. A full adder made up of (a) half adders and (b) NAND gates.

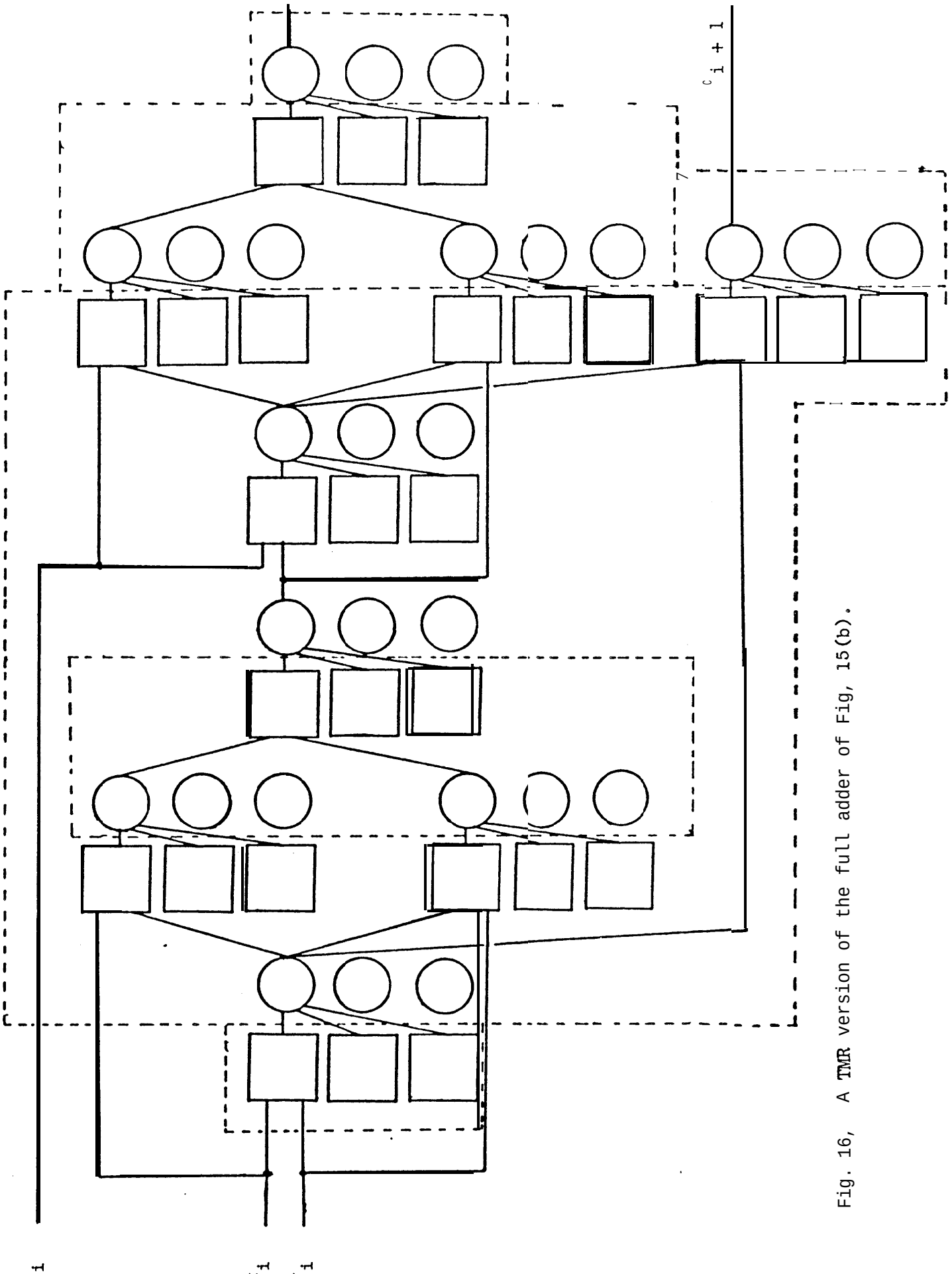


Fig. 16, A TMR version of the full adder of Fig, 15(b).

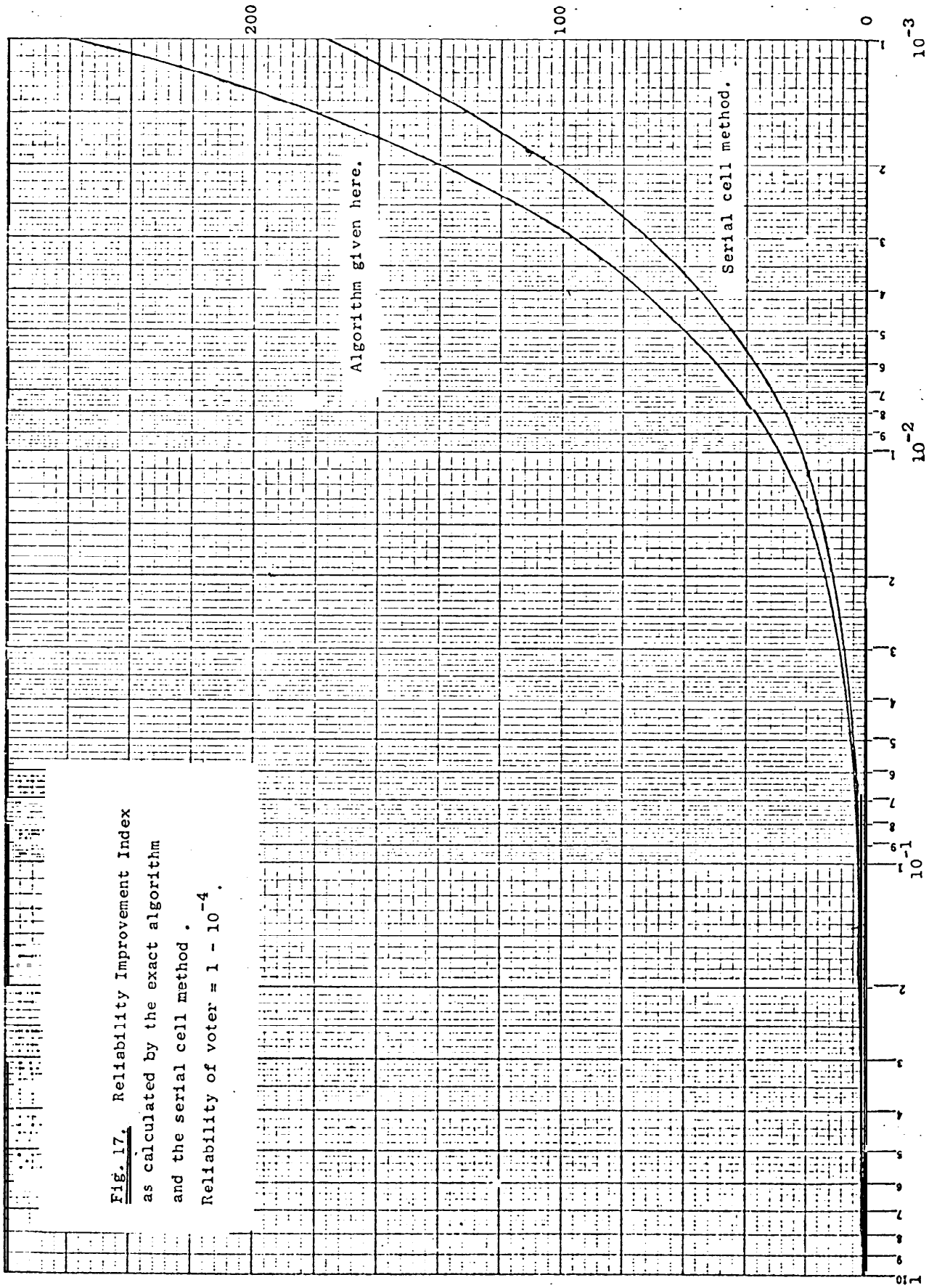


Fig. 17. Reliability Improvement Index as calculated by the exact algorithm and the serial cell method.
Reliability of voter = $1 - 10^{-4}$.

become more reliable.

3.6 Approximation to the Algorithm to Reduce the Computational Complexity

The algorithm described in the previous pages provides a means of finding the exact reliability of a coherent TMR network. The algorithm does not take much storage space, since each combination of rows of S is generated one at a time, and the G_m and G_v values found for that combination. There is no need to remember the combination from one row of the table to the next. What is sacrificed is execution time, since for n voters and m modules in a cell, on the order of 2^{n+m} operations is required, because we have to take all possible combinations of voters and modules. We are in effect trading time for accuracy. The entries in the Fault Matrix are the possible fault patterns for voters and modules which do not cause system failure, A method will now be described to obtain approximate values for some of the entries so that the total execution time is reduced; the reliability estimate is, nevertheless, very close to that which would have been obtained by using the exact method.

For an arbitrary cell, if we assume that every voter trio feeds every module trio, i.e., the S matrix consists of all l's, we get a lower bound on the entries of the F matrix. This is because the assumption restricts the number of failure patterns. The number of ways in which voters and modules can fail increases if some voters or modules can fail independently of others. In the given case, (S matrix consisting of all l's), no voter or module can fail independently of another. Then G_v for every combination of voter trios is 3, and G_m for every

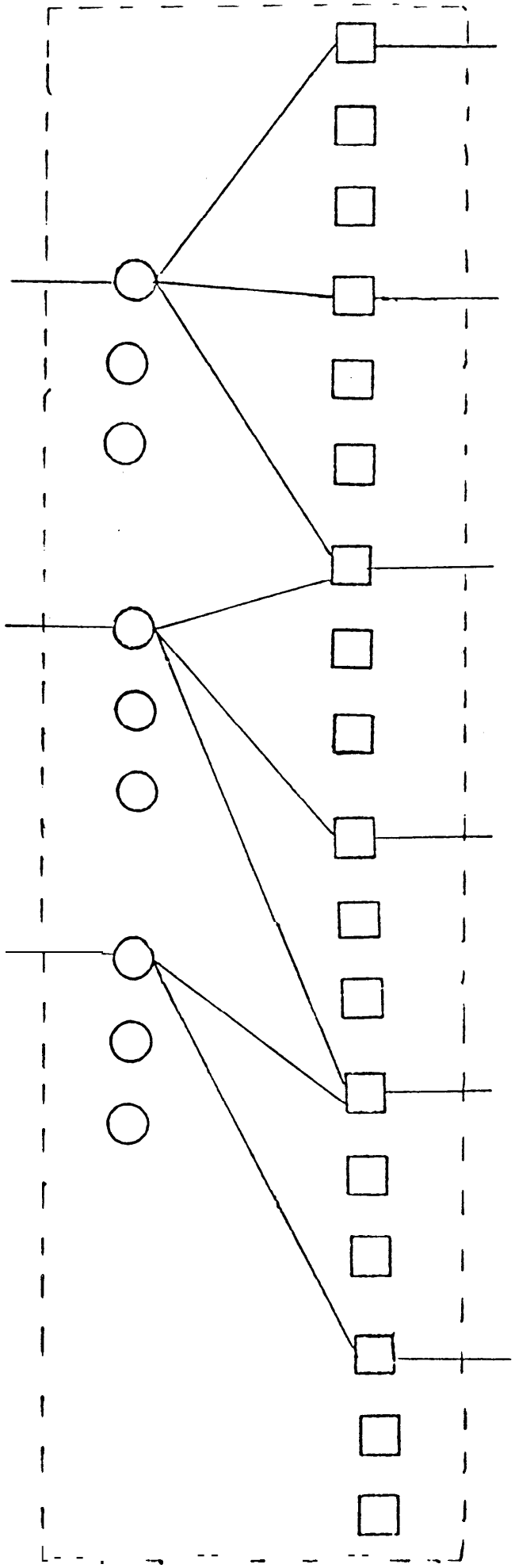
combination of module trios is 1, and the entries of the Fault Matrix are given by,

$$F(i,j) = 3^{\binom{N_m}{j}}, \quad i > 0$$

When we take combinations of i voter trios, they represent cases where i voter failures occur. If the voters are made of single gates (as in threshold voters), or are single integrated circuit chips, they will usually have a very high reliability. Therefore, for i voter failures, the term $(1 - R_v)^1$ term in the reliability equation becomes larger. Hence for large i, we are justified in using the lower bound given above.

One way to use the approximate method to save time without sacrificing too much accuracy is to use the exact method for $i = 0,1,\ldots,i'$, and then use the approximate algorithm for $i = i' + 1,\ldots,N_v$. The choice of i' is dictated by the time available, and the accuracy required; the accuracy depends on the voter and module reliabilities. If an accuracy and a time limit are specified, the reliability can be calculated as described above, and then, i' can be increased by 1, and the reliability again calculated. If the difference in the two reliabilities is less than the accuracy required, we can stop. If not, and there is more time available, the iteration can be continued. If we run out of time, the accuracy to be expected can be returned by the program.

To illustrate this method, the reliability of one of the cells in the Full adder (used in the previous example) is found by the exact method and then approximated. Fig. 18 gives the cell and the Fault Matrices, one using the exact method for all the rows, and the other



	0	1	2	3	4	5	6
0	1	18	135	540	1215	1458	729
1	9	114	579	1500	2079	1458	405
2	15	126	417	708	657	318	63
3	3	18	45	60	45	18	3

(a) Fault Matrix for the cell on the left, using the exact method for all the rows.

	0	1	2	3	4	5	6
0	1	18	135	540	1215	1458	729
1	9	114	579	1500	2079	1458	405
2	9	54	135	180	135	54	9
3	3	18	45	60	45	18	3

(b) Fault Matrix for the cell on the left, using the exact method for rows 0, 1 and the approximation for rows 2, 3.

Fig.18 . Cell used to compare the exact and approximate methods

using the exact method for i = 0,1 and the approximation for i = 2,3. Fig. 19 plots the difference in the reliabilities as calculated by the two methods. The approximate method always gives a lower value of reliability, but, as can be seen from the graph, it is not much lower, only of the order of 10^{-6} to 10^{-5}.

The approximate method for n voters and m modules in a cell requires on the order of n.m operations. Therefore, this method used in conjunction with the exact method can significantly reduce the time required for the reliability calculation.

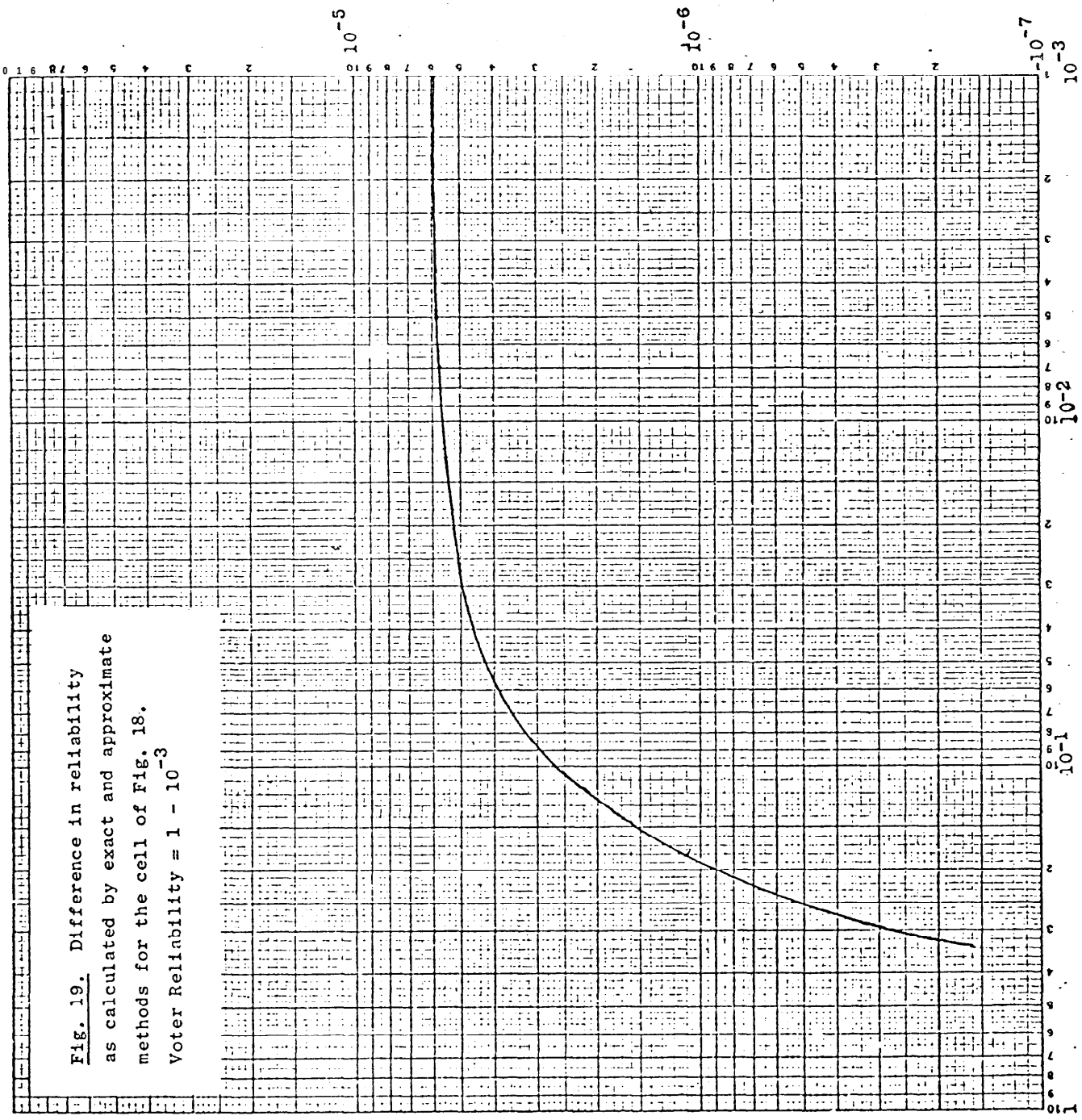


Fig. 19. Difference in reliability as calculated by exact and approximate methods for the cell of Fig. 18. Voter Reliability = $1 - 10^{-3}$

3.7 Modifications to the Algorithm

3.7.1 Different Module Reliabilities

If the reliabilities of the different modules in a cell are different, the entries of the Fault hlatrix must be split up in order to reflect the different failure modes of the different module trios, This information is readily available when the algorithm is developed. An example will show the procedure necessary. Suppose the three module trios in the cell of Fig. 5(a) have modules with reliabilities R_{ml}, R_{m2}, and R_{m3} instead of the same A_m. Consider the combination of voter trios (1,2) for which $G_v = 3$ (from Table 1) and the combination of module trios corresponding to these, (2,3) for which $G_m = 3$. Then the term in the reliability of the cell corresponding to these failures is

$$3 \cdot 3 \cdot R_v^{10}(1-R_v)^2 \; R_{ml}^3 \; R_{m2}^2 \; (1-R_{m2}) \; R_{m3}^2 \; (1-R_{m3})$$

Thus we do not find $\sum G_v \cdot G_m$ but consider each $G_v \cdot G_m$ product as above. Therefore, with only a slight modification to the algorithm, the fact that different modules have different reliabilities can be taken into account,

3.7.2 Compensating Failures

In the previous discussion of the algorithm, only one failure per module trio was assumed. But if one module in a trio is stuck-at-0 at the output, another stuck-at-l at the output, while the third functions correctly,the next level of voters will vote on the correct output. Clearly, neglecting such "compensating failures" gives a lower bound on the reliability.

Such failures may be taken into account by adding a term to the final reliability, as found in the previous section. Suppose there are N_m module trios in a cell, with the reliability of a module being R_m. Let the probability of a module being stuck-at-zero and stuck-at-1 be P_0 and P_1 respectively, where $1-(P_0 + P_1) = R_m$. Considering compensating failures in j module trios, the number of ways in which this can happen is

$$\binom{N_m}{j} 6^j$$

since a compensating failure can occur in 6 different ways in a module trio; the corresponding term in the reliability is

$$\binom{N_m}{j} 6^j \; P_o^{\,j} \; P_i^{\,j} \; R_m^{(3N_m-2j)} \; R_v^{3N_v}$$

for a cell with N_v voter trios. So the term to be added to take care of compensating failures is

$$\sum_{j=0}^{N_m} \binom{N_m}{j} 6^j \; P_0^{\,j} \; P_1^{\,j} \; R_m^{(3N_m-2j)} \; R_v^{3N_v}$$

3.7.3 Module Trios Not Followed by Voter Trios

If every module trio is not followed by a voter trio, the modules cannot fail independently, since they are directly connected to each other. If they are assumed to be independent, we get an upper bound, and if they are all assumed to be connected to each other we get a lower bound to the reliability.

The exact count of failure (or non-failure) patterns in this case is non-trivial. More information must be maintained in the

structure matrix, for example, having rows corresponding to modules too. More work is being done in this area.

4. RELIABILITY MODELING OF NMR NETWORKS

4.1 Introduction

The Triple Modular Redundancy (TMR) concepts described earlier can be generalized [9] to an N-tuply Modular Redundancy (NMR) system having N = 2t + 1 modules and voters, each voter being a t + 1 out of N voter. Such a system can be used where high reliability is required. NMR can also be used as the hard core in standby systems. It might seem that in such systems a TMR core is best since it provides maximum utilization of the modules (when a system has failed, there is only one good module left), but it has been shown [27] that certain switch designs make an NMR core more practical. Also, a TMR core allows only one failure in the core, which may not be ideal.

The algorithm given earlier for the reliability calculation of TMR cannot be carried over per se for the reliability modeling of NMR. The problem arises because in TMR only one failure per trio is allowed, and this fact was implicitly used in the algorithm. In NMR, however, up to t failures per module or voter N-tuple (where N = 2t + 1) are allowed, and this complicates matters. The following section gives a modification of the algorithm for a general NMR.

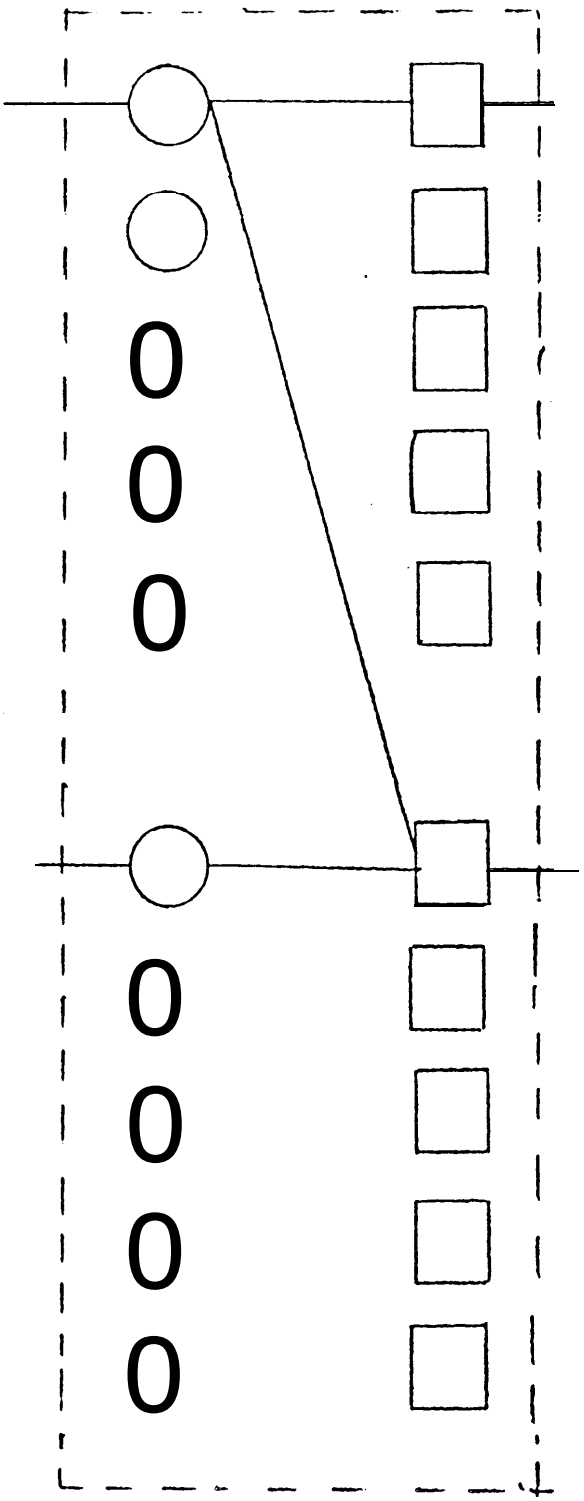
4.2 Extension of the Algorithm to NMR

We will use the cell of Fig. 20 as an example. The development of the algorithm is given in Table 3. The structure matrix S is the same as before, but the Fault Matrix F is $N_v \cdot t + 1 \times N_m \cdot t + 1$ Instead of taking all the possible combinations of the rows of S as previously, we take all possible combinations of rows of S with repetition, with the number of repetitions being restricted to t. This is to include the possibility of multiple failures in a group. The number of repetitions of a row will be kept track of by a superscript. For example, in Table 3, the possible combinations of two rows of S with repetition, the repetitions being restricted to two are,

$$(1^2),\ (1,2),\ (2^2).$$

Now, L is an integer vector of length N_m, and is found by taking the arithmetic sum of the corresponding rows of S, with repetition if necessary. For example, the L for combination $(1^{2'},2)$ in Table 3 is found by summing the vectors (11), (11), and (01), which gives L = (23).

To find G_v for a voter combination, we have to seperate the combination into independent groups as before, While in the previous (TMR) case each group could fail only in three ways, this is not true any more. Suppose one group consists of the rows of S, p^a, q^b, r^c, ..., a lower bound for the partial G_v for that group is found as follows. We take each of the superscripts a,b,c, . . . in turn and use the current superscript and the sum of the previous superscripts. The sum of the previous superscripts, say k, gives the total number of failures in the voters up till this, and the current



$$S = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$$

Fig. 20. Cell used to illustrate the general algorithm.

Voter combinations	G_v	L	Module combinations, G_m (1)	(2)	$\sum 1$	(1^2)	(1,2)	(2^2)	$\sum 2$	$(1^2,2)$	$(1,2^2)$	$\sum 3$	$(1^2,2^2)$
(1)	5	11	5	5	10	4	25	4	33	20	20	40	16
(2)	5	01	5	5	10	10	25	4	39	50	20	70	40
(1^2)	10	22	2	2	4	.1	4	1	6	2	2	4	1
(1,2)	25	12	5	2	7	4	10	1	15	8	5	13	4
(2^2)	10	02	5	2	7	10	10	1	21	20	5	25	10
$(1^2,2)$	20	23	2	2	4	1	4	1	6	2	2	4	1
$(1,2^2)$	20	13	5	2	7	4	10	1	15	8	5	13	4
$(1^2,2^2)$	10	24	2	2	4	1	2	1	6	2	2	4	1
				I I									

Table 3.

Development of the general algorithm applied to the cell of Fig. 20.

	0	1	2	3	4
0	1	10	45	100	100
1	10	100	360	550	280
2	45	285	645	615	210
3	40	220	420	340	100
4	10	40	60	40	10

Table 4.

Fault Matrix of the cell in Fig. 20.

superscript, say ℓ, gives the additional failures in the voters. Since only up to t failures are allowed per group, the current ℓ failures can happen, given k failures already, (if k>t, k is taken as t) in,

$$\sum_{j=k+\ell-\min(k,t)}^{\ell} \binom{\min(t,k)}{j} \cdot \binom{N-k}{\ell-j} \qquad \text{ways.}$$

This follows because the j failures can be chosen from the positions of the k voters already failed or from other positions,, and j varies from $k + \ell - t$ to ℓ since only t failures are allowed per group. Here, as before, $\binom{N}{k} = 0$ if $k < 0$ or $k > N$. This is a lower bound since all voters need not be connected to all voters, and failure patterns may exist which are not counted above. However, this is a tight lower bound, and finding the exact value is very complex, if not impossible. Then the lower bound for the partial G_v for that group is the product of the values for each of the superscripts,

$$\prod_{\substack{\ell = a,b,c, \ldots \\ k = a,\ a+b,\ a+b+c,\ \ldots}} \sum_{j = k+\ell-\min(k,t)}^{\ell} \binom{\min(t,k)}{j} \cdot \binom{N-k}{\ell-j}$$

The G_v for the combination is then the product of the partial G_v's of the independent groups. For example, G_v for combination $(1,2^2)$ in Table 3 is,

$$\sum_{j=-1}^{1} \binom{0}{j} \cdot \binom{5}{1-j} \times \sum_{j=1}^{2} \binom{1}{j} \cdot \binom{4}{2-j}$$

$$= 5 \cdot 4 = 20$$

Gm values are found by taking the combinations of modules with up to t repetitions. Suppose an L is $(a_1, a_2, a_3, \ldots)$ and a combination of modules is $(w^{b_1}, x^{b_2}, y^{b_3}, \ldots)$. Then as for G_v, each of the superscripts $b_1, b_2, b_3, \ldots$ is compared with the entry of L corresponding to w, x. y, . . . , If the entry of L is a, (this means that a voters feeding this module have failed, and for a lower bound we assume that all the a are different), and the superscript is b, (i.e., b modules have failed), the partial G_m is,

$$\sum_{j\,=\,a+b-t}^{b} \binom{a}{j} \cdot \binom{N-a}{b-j}$$

since j module failures can be assigned to positions corresponding to a voter failures or the remaining positions, but only up to t - a of these. This is also obviously a lower bound. The G_m for the combination is then the product of the partial G_m's. For example, for voter combination (2^2), L = (02). Consider module combination $(1^2, 2)$.

$$G_m = \sum_{j=2-2}^{2} \binom{0}{j} \cdot \binom{5}{2-j} \quad x \quad \sum_{j=2+1-2}^{2} \binom{2}{j} \cdot \binom{3}{1-j}$$

$$= \binom{0}{0} \cdot \binom{5}{2} \quad x \quad \binom{2}{1} \cdot \binom{3}{0}$$

$$= 20$$

The entries in the Fault Matrix are found by summing $G_m . G_v$ as for the TMR case. The reliability is then given by,

$$R_{NMR} = \sum_{i=0}^{N_v . t} \sum_{j=0}^{N_m . t} F(i,j) \; . R_v^{N . N_v - i} . (1-R_v)^i . R_m^{N . N_m - j} \cdot (1-R_m)^j$$

5. CONCLUSIONS

Various approaches given in the literature for calculating the reliability of TMR netwroks have been described, and their limitations pointed out. An algorithm has been given to find a very tight lower bound on the reliability of an arbitrary TMR network, and this algorithm has been shown to give a much better lower bound than previous methods. If a network is divided into cells, the cell reliability may be a poor predictor of system performance. For example, if a network has 10 cells, and if the reliability of each cell is found by a method which gives a value about 5% lower, then the reliability of the whole network will be found to be calculated to be about 40% lower than the actual reliability.

For n modules in a cell, the algorithm takes on the order of 2^{2n} operations in the worst case. If there are m cells in a network, for a total of $m \cdot n$ modules, the total operations required are on the order of $m \cdot 2^{2n}$, in the worst case. A general algorithm using cut sets like Jensen's [22] will take on the order of $m \cdot n \cdot 2^{2 \cdot m \cdot n}$ operations in the worst case. This illustrates the advantage of the cellular method, which has to work on only one cell at a time, as compared to other methods which consider the network as a whole. Other methods which use the cellular approach like Klaschka's [4] require that we take all combinations of 1 2, ... n things which is on the order of 2^n operations. Thus, in comparison, the method given is not too time consuming.

An approximation has been given to the algorithm which takes

much less time, and gives a lower bound on the reliability. The algorithm has also been extended to general NMR networks,

REFERENCES

[1] Naresky, J. J., "Reliability Definitions," IEEE Transactions on Reliability, vol. R-19, No. 4, Nov. 1970, pp. 198-200.

[2] Avizienis, A., 'Design of fault-tolerant computers," FJCC, vol. 31, 1967, pp. 733-743.

[3] Bouricius, W. G., W. C. Carter and P. R. Schneider, "Reliability modeling techniques for self-repairing computer systems " Proc. 24 Natl. Conference ACM, Publication P-69, 1969, pp. 295-309.

[4] Klaschka, T. K., 'Reliability improvement by redundancy in electronic systems. I.A method for the analysis and assessment of redundancy schemes," Royal Aircraft Establishment Technical Report 68130, May 1968.

[5] Jensen, P. A., "Quadded NOR logic," IEEE Trans. on Reliability, vol. 12, No. 3, 1963, pp. 22-31.

[6] Gurzi, K. J., "Estimates for the best placement of voters in a triplicated network,' IEEE Trans. on Elect. Computers' vol. EC-14, Oct. 1965, pp. 711-717.

[7] Knox-Seith, J. K. 'Improving the reliability of digital systems by redundancy and restoring organs,' SU-SEL-64-094 Tech. Report No. 4816-2, Stanford Electronics Laboratories, Stanford University, Aug. 1964.

[8] Dunning, M., B. Kolman, and L. Steinberg, "Reliability and fault masking in n-variable NOR trees,' Proc. 6th Annual Sym. on Switching Circuit Theory and Logical Design, IEEE Publication 16c13, pp. 126-142 Oct. 1965.

[9] Mathur, F. P. and A. Avizienis, "Reliability analysis and architecture of a hybrid redundant digital system: generalized triple modular redundancy with self-repair," SJCC, vol. 36, May 1970, pp. 375-383.

[10] Anderson, J. E. and F. J. Macri, "Multiple redundancy application in a computer," Proc. 1967 Annual Symposium on Reliability, 1967, pp. 553-562.

[11] Goldberg, J., K. N. Levitt, and R. A. Short, "Techniques for the realization of ultra-reliable spaceborne computers,' Final Report-Phase I, Project 5580, Stanford Research Institute, Menlo Park, Calif. Sept. 1966.

[12] Tryon, J. G., "Quadded logic," in Redundancy Techniques for Computing Systems, Wilcox and Mann, eds., Spartan Books, Washington, D.C. 1962.

[13] Klaschka, T. K., "Reliability improvement by redundancy in electronic systems. II. An Efficient new redundancy scheme - radial logic," Royal Aircraft Establishment Technical Report 69045, March 1969.

[14] Finkelstein, H. A., "An investigation into the extension of redundancy techniques," Coordinated Science Laboratory Report R-455, University of Illinois, Urbana, Illinois, Feb. 1970.

[15] Siewiorek, D. P., "An improved reliability model for NMR" SU-SEL-72-004 Technical Report No. 24, Digital Systems Laboratory, Stanford University, Dec. 1971,

[16] von Neumann, J., "Probablistic logics and the synthesis of reliable organisms from unreliable components,' Automata Studies, from Annals of Mathematics Studies, No. 34, Princeton University Press, pp. 43-99, 1956.

t-171 Brown, W. G., J. Tierney, and R. Wasserman, "Improvement of electronic computer reliability through the use of redundancy," IRE Trans. on Electronic Computers, Vol. EC-10, pp. 407-416, 1961.

[18] Teoste, R., "Design of a repairable redundant computer," IRE Trans. Electronic Computers, Vol. 11, pp. 643-649, 1962.

[19] Rhodes, L. J., "Effects of failure modes on redundancy," Proc. Tenth Natl. Symp. on Reliability and Quality Control, Washington,D.C., pp. 360-364, 1964.

[20] Longden, M., L. J. Page, and R. A. Scantlebury, "An assessment of the value of triplicated redundancy in digital systems," Microelectronics and Reliability, Vol. 5, Pergamon Press, pp. 39-55, 1966.

[21] Rubin, D. K,, "The approximate reliability of triply redundant majority-voted systems," IEEE Publication 16051, Digest of the First Annual IEEE Computer Conference, Chicago, pp. 46-49, Sept. 1967.

[22] Jensen, P. A., "The reliability of redundant multiple-line networks," IEEE Transaction on Reliability, Vol. 13, No. 1, pp. 23-33, 1964.

[23] Siewiorek, D. P., "On the rapid calculation of the reliability of serial triple-modular redundancy," Technical Note No. 5, Digital Systems Laboratory, Stanford University, Oct. 1970.

[24] Lyons, R. E. and W. Vanderkulk, "The uses of the triple-modular redundancy to improve computer reliability," IBM J. Res. Develop., Vol. 6, pp. 200-209, 1962.

[25] Esary, J. D. and F. Proschan, "The reliability of coherent systems," in Redundancy Techniques for Computing Systems, Wilcox and Mann, eds., Spartan Books, Washington, D. C., 1962.

[26] Roth, J. p., W, G. Bouricius, W. C. Carter, and P. R. Schneider, "Phase II of an architectural study for a self-repairing computer," SAMSO-TR-67-106, Nov. 1967.

[27] Siewiorek, D, P. and E. J. McCluskey, "Switch Complexity in Systems with Hybrid Redundancy," IEEE Transactions on Computers, C-22, 3, March 1973, pp. 276-282.