for the price of increasing the number of arithmetic operations per result element from two to $s + t$ where $s + 1$ is the number of digits and t the number of 1's in the binary representation of the window length w.

Runs with the vectorized algorithm were made on both a 2-pipe and a 4-pipe CYBER 205, using 32-bit as well as 64-bit data. For long result vectors we observed speedup factors of about 3 for 64-bit data with 2 pipes, about 5 for 64-bit data with 4 pipes and 32-bit data with 2 pipes, while 32-bit data on 4 pipes gave almost a factor of 10.

Appendix

A vectorized moving sum algorithm for the CYBER 205 written in CYBER 200 Fortran with vector extensions: $X(J; K)$ denotes $X(J)$ through $X(J + K - 1)$.

```
      SUBROUTINE MSUMV(N, W, A, B, G)
C
C     N    = LENGTH OF THE INPUT ARRAY "A."
C     W    = LENGTH OF SUM WINDOW—MUST BE ODD
C            AND NOT EXCEED "N."
C            WHEN EVEN IT WILL BE TREATED AS IF IT
C            HAD THE VALUE W+1.
C     A    = INPUT ARRAY OF LENGTH "N."
C     B    = OUTPUT ARRAY OF LENGTH AT LEAST
C            "N+W/2." THE FIRST "N" POSITIONS
C            WILL BE FILLED WITH THE MOVING SUMS
C            COMPUTED FROM "A." THE ARRAYS "A" AND
C            "B" MUST BE DISTINCT.
C     G    = SCRATCH ARRAY OF LENGTH AT LEAST
C            "N+W/2."
C
      INTEGER D, W, WT
      DIMENSION A(1), B(1), G(1)
C
C     PREPARE G(O) IN ARRAY B AS W/2 ZEROES FOLLOWED
C     BY THE ABSOLUTE VALUES OF A(1) THROUGH
C     A(N). THEN PREPARE G(1) IN ARRAY G.
C
      D      = W/2
      LEN    = N + D - 1
      B(1    ;D)     = 0.0
      B(1+D  ;N)     =VABS( A(1;N) ; B(1+D;N) )
      G(2    ;LEN)   = B(2;LEN) + B(3;LEN-1)
C
      JS       = 2
      JP       = 2
      LBIT     = AND  ( D , 1 )
      WT       = D / 2
      IF ( LBIT .NE. 0) GOTO 60
C
C     GENERATE THE NEXT G-VECTOR ON TOP OF
C     THE OLD ONE IN ARRAY G.
C
   40 CONTINUE
      G(JS;LEN-JP) = G(JS;LEN-JP) + G(JS+JP;LEN-JP)
      JP     = JP + JP
      LBIT   = AND  ( WT , 1 )
      WT     = WT / 2
      IF ( LBIT .EQ. 0) GOTO 40
C
C     HERE IF THE CURRENT BIT IN W WAS SET.
C     ADD THE LAST G-VECTOR INTO ARRAY B.
C
   60 CONTINUE
      B(1;N) = B(1;N) + G(JS;N)
      JS     = JS + JP
      LEN    = LEN - JP
      IF ( WT .GT. 0) GOTO 40
C
      RETURN
      END
```

Evaluation of On-Chip Static Interconnection Networks

PINAKI MAZUMDER

Abstract—**This correspondence evaluates three types of static interconnection networks for VLSI implementation. The criteria of evaluation have been selected from three orthogonal aspects—physical (chip area and dissipation), computational speed (message delay and message density) and cost (chip yield, operational reliability and layout cost). The main feature of this paper is to augment the selection criteria for the interconnection networks from the classical AT^2 metric and to provide results pertaining to realistic VLSI implementation.**

Index Terms—**Binary tree, cube connected cycles, static interconnection networks, two-dimensional meshes, VLSI implementation.**

I. Introduction

Over the past two decades many interconnection networks have been proposed in the literature for SIMD architectures. Extensive accounts of these networks and their performance evaluation have been reported in [1]–[4]. Now with the advent of submicron silicon technology, more than ten million devices can be integrated in the VLSI circuit [5] and it has opened a new vista in parallel processing. On-chip multiprocessing by several processors for executing special algorithms is envisioned to be the major application goal of the future generation computers using massive parallelism.

In order to justify the need for reevaluation of interconnection networks, it should be noted that in the design of earlier networks adapted for non-VLSI environment:

a) spatial distribution of the processors is not a constraint on the design,

b) signal propagation time is exclusively determined by the velocity of electromagnetic wave in the resistive medium and is negligibly small compared to the speed of operation. Thus, the length of the interconnecting wire is not a constraint on design,

c) cost of the system is directly proportional to the average number of links per node, and

d) fault tolerance capability of such networks is merely a topological property (i.e., whether alternate message transmission routes exist or not).

On the contrary, under a VLSI environment:

a) the spatial distribution of the processors play an important role on the total chip area,

b) the interconnecting wire behaves as transmission line having both resistive and capacitive components and signal propagation time is largely dependent on these values which are directly proportional to the length of the interconnection,

c) link per node does not have any direct relevance to design cost. The regularity of the networks topology decides the layout cost and size of chip decides the fabricational cost, and

d) fault tolerance has an additional role to play. To improve the device yield and thereby to reduce the overall cost, it is necessary to introduce redundant processors. The existence of long interconnects increases the chip failure probability, both at the time of fabrication and in normal operation.

This correspondence envisages to evaluate the performance of three types of static interconnection networks and to determine how these new constraints modify the performance of these interconnection networks in VLSI implementation. The choice of the networks

Manuscript received January 17, 1986; revised June 26, 1986. This work was supported by the Semiconductor Research Corporation under Contract 84-06-049.

The author is with the Coordinated Science Laboratory, University of Illinois, Urbana, IL 61801.

IEEE Log Number 8612061.

has been motivated from the facts that these networks have been widely pursued in the literature for designing many algorithms [6], [7] and all of them have optimal VLSI layouts with reference to AT^2 metric proposed by Thomson [8]. Also, these provide the insights to three topologically different classes of networks. The networks discussed are: two dimensional meshes, binary trees and cube connected cycles (CCC). These three networks conceptually belong to separate classes in the sense that if the interprocessor link is constrained to have constant length, their distribution in three dimensional space reveals somewhat planar, conical, and spherical surfaces, respectively. Thus the results of this evaluation can be easily extended to assess the performance of other networks, because most networks have one of these three topologies in three-dimensional space. Networks with wraparound connections like the ILLIAC IV describe a torroidal surface and have overall performance similar to two dimensional meshes [3].

The computational model used here to asymptotically evaluate the performance of the networks is similar to Thompson's [8] model and additionally accounts for device faults and chip yield. Faults are assumed to occur randomly and due to the occurrence of random defects the interconnects failure probability increases directly with its length [9] and the overall chip yield varies inversely with chip area [10]. The criteria of evaluation are enumerated from three orthogonal viewpoints, viz., area, speed, and cost. The speed of computation depends both on the topology of the networks and the presence of long interconnects. The number of links and the interconnection structure also decide its message traffic density and is an important measure to determine its bottleneck in communication. The cost aspects consider the fabrication cost and the replacement cost due to poor reliability of the networks. The manufacturing cost of the IC is related to the total chip area and the regularity of the layout [11].

The fault-tolerance capability largely decides the reliability of a working chip. Due to a host of causes, like electromigration, Kirkendall's effects, hot electron effects, etc., a processor or a link may fail during the normal use of the chip. Depending on the topology of the networks, the effect of failure of a single processor or a link will adversely affect the operation of the network. Normally the level of masking and processing associated with the interconnect is far more simpler than the processors and the reliability of the interconnect is higher than the reliability of the processor. So the reliability due to the processor failure and the interconnect failure are separated and different measures are used here. Since the probability of interconnect failure is directly proportional to its length, total length of the interconnects in the network is used as the measure of fault tolerance due to interconnect failure and is denoted by $R_l = x^{\lambda_l}$ where $x < 1$ and λ_l is the mean life of a wire of length l. The failure of a single processor will result in performance degradation because it may isolate one or more processors. The degradation in computing will be determined by the maximum number of connected processors (say N') in the network due to the occurrence of a single processor failure. The value of N' depends on the topology and the location of failed processor. The ratio of maximum value of N' to the original size of the network is defined as the degradation factor, δ and is used as a measure of fault tolerance. The reliability of the network can be improved by introducing redundant processors. The ratio of the reliability of redundant network to that of the nonredundant (original) network is defined as *reliability improvement factor* (RIF). Since the addition of redundant processors increases the chip area, the ratio of RIF to the number of redundant processors (denoted by ρ) is also used here as a measure of fault tolerance. Overall fault-tolerance capability of three networks will be graded as *High*, *Medium*, and *Low* by making a relative comparison of these three measures.

II. Evaluation of Networks

A. Two-Dimensional Meshes

The two-dimensional mesh network is shown in Fig. 1. Each processor is represented as a unit square and the interconnect length can be ignored. Thus the overall chip area is approximately equal to N and the chip yield is $O(1/N)$.

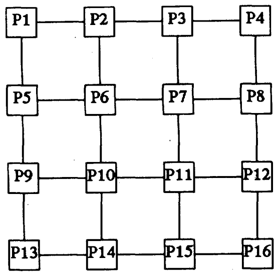

Fig. 1. A 4 × 4 mesh network.

In order to compute the delay in a $\sqrt{N} \times \sqrt{N}$ mesh, it should be noted that the message path length between two arbitrarily located processors at (i, j) and (k, l) within the square grids is given by the *city block distance* $d = |i - k| + |j - l|$. Thus, the average message path length from a source processor is a function of its location (i, j), assuming the lower leftmost processor is at (0, 0). If the source processor is at any of the locations (0, 0), $(n, 0)$, $(0, n)$ or (n, n) [where $n = \sqrt{N}$] then the average message path length is $\bar{d}_1 = n^{-2}[2 * 1 + 3 * 2 + \cdots + n * (n - 1) + (n - 1) * n + \cdots + (2n - 2)] = O(n) = O(\sqrt{N})$. If the source processor is at $(n/2, n/2)$, then the average message path length is $\bar{d}_2 = 4n^{-2}[2 * 1 + 3 * 2 + \cdots + (n/2) * ([n/2] - 1) * [n/2] + \cdots + (n - 1)] = O(n) = O(\sqrt{N})$. For the source processor in any other position it can be shown that the average message path $\bar{d}$ is $O(\sqrt{N})$ and satisfies the inequality $\bar{d}_2 < \bar{d} < \bar{d}_1$. Thus, the average delay between the processors is $\bar{D} = O(\sqrt{N})$.

The total number of links in the square mesh is equal to $2\sqrt{N}(\sqrt{N} - 1) = O(N)$ and the average message path is $O(\sqrt{N})$. Assuming all the N nodes issue messages simultaneously, the average message traffic density is then $\bar{M} = NO(\sqrt{N})/O(N) = O(\sqrt{N})$.

Since the average delay is $O(\sqrt{N})$ and the chip size is $O(N)$, the average chip dissipation is $O(N^{3/2})$.

The layout can be constructed hierarchically and a block of 4^k processors can be laid in kth step paying 2^{k+1} cost. Thus, a network of size N needs $\Sigma_{k=1}^{\log_4 N} 2^{k+1} = 2^{\log_4 N+2} - 4 = O(\sqrt{N})$ cost. The regularity factor is defined as the ratio of total number of links to the number of links actually laid and is $O(N)/O(\sqrt{N}) = O(\sqrt{N})$.

Since the network consists of nearest neighbor-type connections, interconnect reliability is $R_l \approx 1$. The failure of a single processor does not impair the performance of the networks drastically. Due to the presence of many parallel paths in the square grids, the failure of a single processor results into isolation of the failed processor only and does not impair the performance of the networks drastically. The degradation factor is thus $\delta = (N - 1)/N$. If $R_p = e^{-\lambda_p t}$ is the functional reliability of each processor in the meshes, then the overall reliability of the network is $R_M = R_p^N$. This reliability can be sufficiently ameliorated by adding a redundant row and the overall network can be made to be $(\sqrt{N} - 1)$ fault-tolerant. The reliability of the redundant mesh network is $R_{rM} = (R_p^n + n(1 - R_p)R_p^{n-1})^n$ where $n^2 = N$. The reliability improvement factor, RIF_M due to the redundant processors is given by $\text{RIF}_M = R_{rM}/R_M$ such that $\rho = (1/(2n - 1))(1 + n(R_p^{-1} - 1))^n$.

The delay can be improved for mesh networks if the processors belonging to each column and each row are connected hierarchically as binary trees. Such networks are known in the literature as orthogonal tree networks [12] and mesh of trees [13]. The average delay for such networks reduces to $O(\log N)$ but the chip area increases to $O(N \log^2 N)$. The overall performance thus does not improve. On the contrary, the presence of long interconnects of length $O(\sqrt{N})$ actually increases the average delay to $O(\sqrt{N})$. Moreover, these networks suffer from many practical limitations as poor yield (due to large chip size), poor regularity (due to presence of mesh and trees combined), $O(N \log N)$ crossovers, long interconnects, etc.

B. Binary Tree Networks

The complete binary tree networks of N processors needs at most $O(N \log N)$ layout area corresponding to $O(N)$ leaves and $O(\log$

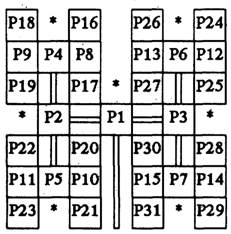

Fig. 2. *H* tree layout.

N) height of the tree. A better layout which needs optimal $O(N)$ area can be constructed using the concept of an H diagram, originally proposed by Marihugh and Anderson [14] as a graphical approach to logic design. Horowitz and Zorat [15] have constructed the algorithms for the generation of such a layout and the modified network is henceforth referred as an H tree. The H tree layout of the binary tree is shown in Fig. 2. The total area for a complete binary tree of N processors can be computed from the following recursive relationship

$$A(N)=[2\sqrt{A(\lceil N/4 \rceil)}+1]^2 \quad \text{with } A(1)=1.$$

Assuming $N = 2.4^k - 1$, it can be shown that $A(N) = 4^{k+1} - 2^{k+2} + 1 = 2N - 2.82\sqrt{N+1} + 3$.

The longest wire in the layout is of size $\sqrt{N}/2$ and the total length of wires in the layout is given by the recurrence relation

$$L(N)=4L(\lceil N/4 \rceil)+\sqrt{N} \quad \text{with } L(7)=1$$

which gives a solution $L(N) = O(\sqrt{N} \log N)$.

The worst case delay occurs when a message is propagated between the leaves through the root of the tree. Assuming $O(l)$ delay for both metal wire (without driver) and polysilicon wire (with interspersed driver) [16], the worst case delay can be given by $D_{\max}(N) = 2 * [2 \Sigma_{j=1}^{k} (2^{k-j} - 1) + 2k + 1] = 2\sqrt{2(N+1)} - \log (N + 1) - 2 = O(\sqrt{N})$. But this delay is smaller than in a mesh network, since only 2 log N intermediate processors are visited as opposed to $2\sqrt{N} - 1$ in a mesh network. The average message delay between root and other processors can be given by

$$\bar{D}=\sum_{j=0}^{2k-1} \sum_{i=1}^{\lfloor j/2 \rfloor} 2^{j-2k}(2(2^{k-i}-1)+(j \bmod 2)(2^{k-\lfloor j/2 \rfloor -1}-1)+\log j)$$

$$=O(\sqrt{N}).$$

To calculate the message traffic density on a link, consider $N - 1$ time units during which $N(N - 1)$ messages are generated and each node on the average will have sent a message to each of the others. Let $h = \log (N + 1)$ be the height of the tree network, denoted as T_h, such that $|T_h| = N$. A subtree of T_h at level k from leaves is shown as T_k such that $|T_k| = 2^k - 1$. A link between level k and level $k + 1 \leqslant h$ will be used to transmit messages between i) all the nodes of the left and the right subtrees each of size $|T_k|$ and ii) one subtree of size T_k connected by the link and $(N - 2T_k)$ nodes of the tree, T_h (Fig. 3). Thus, the message density per unit time at a link between level k and level $k + 1$ is $M(k) = (2/(N - D)[|T_k|^2 + (N - 2|T_k|)\cdot|T_k|] \approx (2^h - 2^k)(2^k - 1/2^{h-1} - 1)$. Since $M(k)$ is a monotonically increasing function of k, the maximum congestion occurs at the link between the root and its sons (i.e., $k = h - 1$) which can be obtained by solving $\partial M(k)/\partial k = 0$. The total number of messages that pass through these links is $M(h - 1) = 2^{h-1} \approx N/2 = O(N)$.

The layout can be constructed hierarchically and each level of embedding needs $7 \times O(1)$ cost and the overall connection cost for a network of N processors is equal to $O(\log_4 N)$. The total number of links in a binary tree is equal to $N - 1$. Thus the regularity factor is $O(N/\log N)$.

The fault tolerance capability of the network due to the failure of a single processor depends on the location of the processor. If the external communication is done solely through the root, its failure will have total disastrous effect invalidating the usability of the IC. Since there is no parallel path for message flow, failure of any links will truncate the operability of the chip. If any processor other than the root fails, it will also reduce the performance of the IC by an amount depending on its location in the tree. The computational degradation that occurs due to a faulty processor or an interconnect at level i from the leaf nodes can be defined as the number of processors which are eliminated due to the fault at level i and is equal to $2^i - 1$. If the external communication is made through leaf nodes, then $\delta = (N - 1)/(2N)$. The network can be restored to function normally by replacing the defective processor by a redundant processor. Redundant processors can be placed in the extra space available within the chip and rerouting can be done by electrically programmable routing technique. Since only $N - 2.8\sqrt{N+1} + 3$ space is available for laying out the redundant processors, redundancy can be added for nodes till level 2 from the leaf nodes, i.e., level $h - 2$ from the root. Thus, the leaf processors and their fathers are not replicated and all other nodes in the tree are replicated at locations shown by * in Fig. 2. If R_p is the reliability of each processor then the overall reliability of the tree network without any redundancy is $R_T = R_p^N$. Clearly, $\lambda_l = O(\sqrt{N} \log N)$. If the redundancy is added as described above, then the reliability of the redundant network is $R_{rT} = R_p^{3N/4} \times (2R_p - R_p^2)^{N/4}$. The reliability improvement factor RIF$_T$ is given by RIF$_T$ $= R_{rT}/R_T = (2 - R_p)^{N/4}$ and $\rho = (4/N)(2 - R_p)^{N/4}$. Thus, the reliability of the tree network is poorer compared to the meshes.

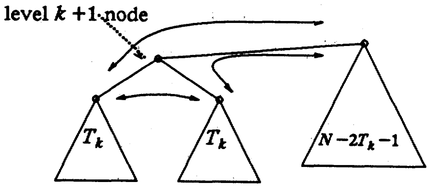

Fig. 3. Message flow through a level $k + 1$ node in a tree network.

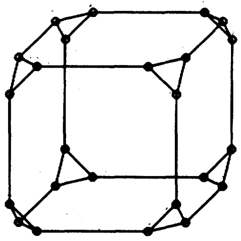

Fig. 4. Cube connected cycle topology for 3.2^3 processors.

C. Cube Connected Cycles

An m-dimensional cube connected cycle (CCC) is a network which can be derived from a Boolean hypercube of 2^m vertices by replacing each vertex with a cycle of m vertices. This was originally proposed by Preparata and Vuillemin [7] to ensure that the degree of each vertex is bounded to 3 and not to m as in an m-cube network. The topology of a 3-dimensional CCC with $3.2^3 = 24$ processors is shown in Fig. 4 and its optimal VLSI layout has been given in Fig. 5. It should be noted from Fig. 5 that the layout of an m-dimensional CCC can be made by laying out m C_i routes in an m-cube [17] on a grid graph and replacing the vertices of m-cube by 2^m cycles of size m. Clearly, the maximum height of the cycle is $\Sigma_{i=0}^{m-1} 2^i = 2^m$ and there are totally 2^m cycles. Thus, the total area required by an m-dimensional CCC is 2^{2m+1} assuming that the width of the edge is equal to the square root of processor area. Since $N = m2^m$, then $2^m = N/(\log N - \log m)$, i.e., $m \approx \log (N/\log N)$. Thus, the total chip area is approximately $O(N^2/\log^2 N)$.

Let the interprocessor link in the cycle be called ring link (vertical lines in Fig. 5) and the interprocessor link between two adjacent cycles be called vertex link (horizontal lines in Fig. 5). Using Wittie's algorithm for message routing, it can be shown that on the average a

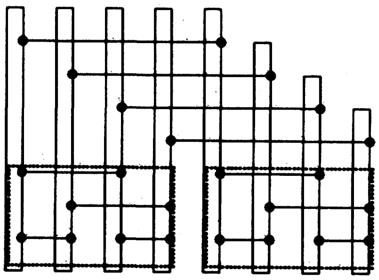

Fig. 5. Layout of CCC with 3.2^3 processors.

TABLE I
EVALUATION OF THREE STATIC INTERCONNECTION NETWORKS

Evaluation of Meshes, H-Tree and CCC for VLSI application with respect to Physical, Computational and Cost Aspects								
NETWORK STRUCTURE	EVALUATION CRITERIA AND PERFORMANCE							OVERALL RESULTS
	Computational Aspects		Physical Aspects		Cost Aspects			
	Average Message Delay	Average Message Density	Chip Area	Power Consumption	Chip Yield	Layout Regularity Factor	Fault Toler-ance	
MESHES	$O(\sqrt{N})$ 2	$O(\sqrt{N})$ 2	$O(N)$ 3	$O(N^{3/2})$ 3	$O\left[\frac{1}{N}\right]$ 3	$O(\sqrt{N})$ 2	High 3	18
H-TREE	$O(\sqrt{N})$ 3	$O(N)$ 1	$O(N)$ 2	$O(N^{3/2})$ 2	$O\left[\frac{1}{N}\right]$ 2	$O(\frac{N}{\log N})$ 3	Low 1	14
CCC	$O\left[\frac{N}{\log N}\right]$ 1	$O(\log N)$ 3	$O\left[\frac{N^2}{\log^2 N}\right]$ 1	$O\left[\frac{N^3}{\log^3 N}\right]$ 1	$O\left[\frac{\log^2 N}{N^2}\right]$ 1	$O(\log N)$ 1	Medium 2	10

message traverses $m/2$ vertex links and $(5m/4) - 2 + 2^{1-m}$ ring links if m is even (and an additional $1/(4m)$ ring links if m is odd). It may be noted that Wittie's average path length analysis by his routing algorithm was incorrect and the correct result is stated above. In a two-layered interconnect model using Manhattan layout [18], wires are either horizontal or vertical and cannot be both metal. Thus, in order to reduce the propagation delay, it is needed that the cycle links should be made of metal and the vertex link should be made of polysilicon (or diffusion). It may be noted that in an m-dimensional CCC, there are $m\ 2^{m-1}$ vertex links having a total interconnection length of $2^{m-1}\Sigma_{i=1}^{m} 2^i$ so that average vertex length is $(2^{m+1} - 2)/m$. Thus the average vertex delay is $4(2^m - 1)$ i.e., $O(N/\log N)$. The asymptotic average delay over metal ring link is $O(\log N)$. The worst case delay is due to message transmission between two processors at the opposite edges of the chip and is proportional to the perimeter of the chip, i.e., $O(N/\log N)$. Since each node has degree 3, the total number of links is equal to $3N$. If N messages are generated on the average in unit time, then the average message density is equal to $\bar{M} = (N/3N) \times O(m) = O(\log N)$.

Since the chip area is very large, the yield is $O(\log^2 N/N^2)$, which is very poor compared to the mesh and the tree networks. The average energy dissipation of the chip is $O(N^3/\log^3 N)$.

The layout can be partially constructed hierarchically. An $N(=m2^m)$-node CCC composes of two $(m - 1)2^{m-1}$-node CCC and 2^{m-1} connections as is evident from Fig. 5. Thus the layout cost to construct an N-node CCC is $2^m + m - 1 = O(N/\log N)$. Since the total number of connections in the layout is equal to $1.5N$, the regularity factor is $O(\log N)$ and is poor compared to the mesh and the tree networks.

The fault tolerance capability of CCC is good because there is always an alternate path to reroute the message like in the mesh. Thus the failure of a single processor will not have any drastic effects on the performance of the network and $\delta = (N - 1)/N$. But due to presence of many long interconnects and large chip area, the reliability is not as good as the meshes. Total length of interconnect within the chip is $O(N^2/\log^2 N)$ due to the presence of $N/\log N$ cycles of average height $N/\log N$. Thus $\lambda_l = O(N^2/\log^2 N)$. The reliability of the network due to processor failure can be given by $R_{\text{CCC}} = R_p^N$. If one redundant processor is added to each cycle, then the reliability improves to $R_{r\,\text{CCC}} = (R_p^m + m(1 - R_p)R_p^{m-1})^{N/\log N}$. The reliability improvement factor, RIF_{CCC}, is given by $\text{RIF}_{\text{CCC}} = R_{r\,\text{CCC}}/R_{\text{CCC}}$ and $\rho = (1/m)(1 + m(1/R_p - 1))^{N/\log N}$.

III. Comparison of Three Classes of Networks

From the analyses done in the previous section, it is evident that each network has certain strong aspects and certain weak aspects. It is difficult to relate all these aspects by a compact formula which can be utilized as a performance metric. A weak effort in this respect was originally done by Mead and Rem [19], [16] relating the area (A) and the speed (T^{-1}) and proposing the *rental time* of the chip AT as a metric. Thompson [8] has extended the concept for any arbitrary network by showing that AT^2 indicates a better performance metric. He has related the area and the speed of computation in a network through its *minimal bipartition width*, ω and have shown that a computational problem can be solved by exchanging information over ω such that the speed of computation is directly proportional to the cardinality of ω while the area of planar implementation of the network is directly proportional to the square of the size of ω. But this metric accounts for the lower bound of the chip area. Savage [20] has contended that A^2T reflects a better evaluation for certain computational problems like binary sorting. From all these contradictory claims, it is evident that it is virtually not possible to correlate all the criteria discussed here. An alternative strategy has been adopted here. The networks have been given credit points for each criterion depending on their relative merits and the total points have been used as a performance index for the network. The results of the evaluation with respect to different criteria have been shown in Table I. The credit points have been assigned on the basis of relative merits of the networks. From the values of total points, it can be seen that the two-dimensional mesh networks indicate overall better performance than the H tree and the CCC. This is in direct contrast to the results of Wittie [3], Siegel [21], [22], etc., who have concluded that fast networks like CCC, PSN, spanning bus hypercubes, etc., have better overall performance. It may be argued whether it is appropriate to give same weight to all the criteria. But it can be easily seen that the conclusion remains true even if different weights are ascribed to three orthogonal aspects (i.e., criteria having same aspect only have same weight).

IV. Conclusion

The basic conclusion emerging out of the analyses done here is that the cellular networks which have a similar structure to the mesh can be cost effectively implemented for VLSI implementation and are highly suitable for VLSI parallel processing. The penalty in delay can be offset by the gains of several criteria discussed in this correspondence. The faster topologies like CCC, PSN, tree, etc., do not provide an overall good performance because of long interconnects which introduce high delay and large chip area which reduces the chip yield. The conclusion is made here on the basis of the topological properties of the networks and not on actual implementation of a specific algorithm. Special algorithmic features can be exploited to improve the average message delay over a specific network topology.

Acknowledgment

The author wishes to thank Profs. J. Patel, D. Reed, and B. Wah for their comments on the initial version of this correspondence.

References

[1] K. J. Thurber, "Interconnection networks—A survey and assessment," *AFIPS Conf. Proc.*, vol. 43, pp. 909–919, 1974.

[2] G. A. Andersen and E. D. Jensen, "Computer interconnection structures: Taxonomy, characteristics and examples," *ACM Comput. Survey*, vol. 7, pp. 197–213, Dec. 1975.

[3] L. D. Wittie, "Communications structures for large networks of microcomputers," *IEEE Trans. Comput.*, vol. C-30, pp. 264–273, Apr. 1981.

[4] T.-Y. Feng, "A survey of interconnection networks," *IEEE Computer*, vol. 14, pp. 12–27, Dec. 1981.

[5] D. F. Barbe, *Very Large Scale Integration: Fundamentals and Applications*. Berlin: Springer-Verlag, 1980.

[6] M. J. Atallah, "Algorithms for VLSI networks of processors," Ph.D. dissertation, Johns Hopkins Univ., Baltimore, MD, 1983.

[7] F. P. Preparata and J. Vuillemin, "The cube connected cycles: A versatile network for parallel computation," in *Proc. 20th Ann. IEEE Symp. Foundations Comput. Sci.*, pp. 140–147, 1979.

[8] C. D. Thompson, "A complexity theory for VLSI," Ph.D. dissertation, Carnegie-Mellon Univ., Pittsburgh, PA, 1980.

[9] C. H. Stapper, "Modeling of integrated circuit defect sensitivities," *IBM J. Res. Develop.*, vol. 27, pp. 549–557, June 1983.

[10] R. B. Seeds, "Yield, economic and logistic models for complex digital arrays," *IEEE Int. Convention Rec.*, Mar. 1967, pp. 60–61.

[11] C. H. Sequin, "Managing VLSI complexity: An outlook," *Proc. IEEE*, vol. 71, pp. 149–166, 1983.

[12] D. Nath, S. N. Maheswari, and P. C. P. Bhat, "Efficient VLSI networks for parallel processing based on orthogonal trees," *IEEE Trans. Comput.*, vol. C-32, pp. 569–581, June 1983.

[13] F. T. Leighton, "A layout strategy for VLSI which is provably good," in *Proc. 14th Symp. Theory of Comput.*, 1982, pp. 85–98.

[14] G. E. Marihugh and R. E. Anderson, "The *H* diagram: A graphical approach to logic design," *IEEE Trans. Comput.*, pp. 1192–1196, 1971.

[15] E. Horowitz and A. Zorat, "The binary tree as an interconnection network: Applications to multiprocessor systems and VLSI," *IEEE Trans. Comput.*, vol. C-30, pp. 247–253, Apr. 1981.

[16] C. A. Mead and L. A. Conway, *Introduction to VLSI Systems*. Reading, MA: Addison-Wesley, 1980.

[17] K. Hwang and F. A. Briggs, *Computer Architecture and Parallel Processing*. New York: McGraw-Hill, 1984.

[18] C. Y. Lee, "An algorithm for path connection and its applications," *IRE Trans. Electron. Comput.*, vol. EC-10, pp. 346–365, Sept. 1961.

[19] C. Mead and M. Rem, "Highly concurrent structures with global communications," *IEEE J. Solid State Circ.*, vol. SC-14, pp. 455–462, Apr. 1979.

[20] J. E. Savage, "Planar circuit complexity and the performance of VLSI algorithms," in *Proc. CMU Conf. VLSI Syst. and Comput.*, 1981, pp. 61–67.

[21] H. J. Siegel, "Analysis techniques for SIMD machine interconnection networks and the effects of processor address masks," *IEEE Trans. Comput.*, vol. C-26, pp. 153–161, Feb. 1977.

[22] ——, "A model of SIMD machines and a comparison of various interconnection networks," *IEEE Trans. Comput.*, vol. C-28, pp. 907–917, Dec. 1979.

A New Built-In Self-Test Design for PLA's with High Fault Coverage and Low Overhead

ROBERT TREUER, VINOD K. AGARWAL, AND HIDEO FUJIWARA

***Abstract*—This correspondence presents a new built-in self-test design for PLA's, that has a lower area overhead and higher multiple fault coverage (of three types of faults: crosspoint, stuck, and bridging) than any existing design. This new design uses function independent test input patterns (which are generated on chip), compresses the output responses into a function independent string of parity bits (whose fault-free expected values are generated on-line with a simple circuit), and detects all single faults and more than $(1 - 2^{-(m+2n)})$ of all multiple faults where m and n represent the number of product terms and input variables, respectively.**

***Index Terms*—Built-in self test (BIST), fault coverage, fault models, output response compression, parity bits, programmable logic array (PLA), test pattern generation, VLSI design.**

Manuscript received June 20, 1985; revised December 27, 1985 and June 11, 1986. This work was supported by a Strategic Grant and Fellowship from the Natural Sciences and Engineering Research Council of Canada.

R. Treuer and V. K. Agarwal are with the Department of Electrical Engineering, McGill University, Montereal, P.Q. H3A 2A7, Canada.

H. Fujiwara was with the Department of Electrical Engineering, McGill University, Montreal, P.Q. H3A 2A7, Canada. He is now with the Department of Electrical and Communications Engineering, Meiji University, Tokyo, Japan.

IEEE Log Number 8612063.