An Area-Efficient Reconfigurable Binary Tree Architecture

Chung-Han Chen and Nian-Feng Tzeng

The Center for Advanced Computer Studies
University of Southwestern Louisiana
Lafayette, Louisiana 70504

Abstract

The VLSI layouts of most fault-tolerant binary tree architectures are based on the classical H-tree layout, resulting in low area utilization and likely an unnecessarily high manufacturing cost simply due to the waste of a significant portion of silicon area. In this paper, we present an area-efficient approach to the reconfigurable binary tree architecture. Area utilization and interconnection complexity of our design compare favorably with other known approaches. In the reliability analysis, we take into account for the first time the fact that accepted chips (after fabrication) are with different degrees of redundancy initially, so as to obtain results which better reflect real situations.

1. Introduction

The binary tree architectures have received significant attention for parallel and hierarchical computing applications recently. It has been observed that many problems, such as sorting, searching, dictionary, and divide-and-conquer problems, can be solved with the binary tree architectures effectively. With advances in VLSI/WSI technology, recent research interest has centered around realizing a whole binary tree in a single chip/wafer. The WSI implementation can further improve the performance and reduce the manufacturing cost by eliminating the needs of individual chips packaging and external connections.

A major problem associated with high density VLSI/WSI chips is its low yield. To overcome this problem, several fault-tolerant binary tree architectures have been proposed in the literature recently, including the SOFT approach [1], the Cluster-Proof approach [2] and the Modular approach [3]. In large area VLSI/WSI implementation, however, the SOFT approach and Modular approach may not be desirable because they fail to survive most of clustering defects, a commonly found defect distribution. The Cluster-Proof approach, on the other hand, deals with the yield improvement for large VLSI/WSI systems where clustering defects are likely. Compared with others, the Cluster-Proof scheme provides better yield and reliability, but at the cost of higher redundancy in links and switches. The need of a large amount of switches and links limits the application of the Cluster-Proof approach (e.g., only 1-bit serial links are suggested in [2]). Furthermore, the Cluster-Proof binary tree architectures basically follow the classical H-tree layout [4]. Such layout exhibit poor area utilization, particularly for large sized systems. As a result, the manufacturing cost tends to be high.

This work was supported by the NSF under Grant MIP-8807761.

Reliability of manufactured chips is another essential factor that has to be considered in system design and production. All previous reliability studies concentrated merely on the analysis of fault-free systems, assuming that all added spares are available and can be used to replace operational faults. Clearly, such analyses fail to reflect the fact that the number of spares available for reliability enhencement may vary from one chip to another, depending on how many spares have been employed to substitute fabricating defects. Reliability estimation able to capture this fact is highly desired.

In this paper, we present a new scheme for the fault-tolerant binary tree architecture that leads to improved area utilization when compared with the Cluster-Proof approach. In the reliability analysis, we consider for the first time the fact that accepted chips (after fabrication) are with different degrees of redundancy initially. We show that our scheme is superior to the previous approaches in terms of yield, interconnection complexity and area utilization.

2. The Proposed Binary Tree Architecture

In our scheme, the entire binary tree is divided into an upper subtree and many lower subtrees. Each lower subtree together with its spares is embedded in a rectangular array, in a more efficient manner than the H-tree layout strategy. It has been shown [3] that a k-level binary tree (without spares) can be effectively embedded in a $2^{k/2} \times 2^{k/2}$ (if k is even) or a $2^{(k-1)/2} \times 2^{(k+1)/2}$ (if k is odd) rectangular array. An example of embedding a 6-level binary tree in an 8X8 rectangular array is shown in Fig. 1. Our approach is to construct a desired fault-free logic array from a larger host array through reconfiguration (so as to achieve fault/defect tolerance), and then map the embedded binary tree onto the logic array. The H-tree layout is then followed to construct the upper subtree, with each lower subtree as a "big" leaf node.

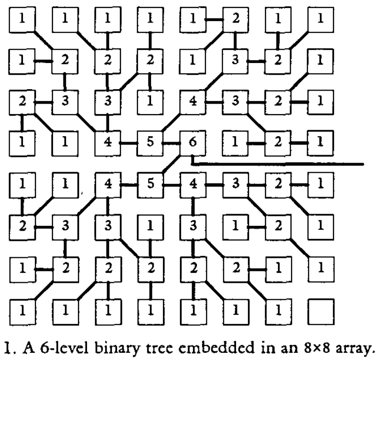
Fig. 1. A 6-level binary tree embedded in an 8×8 array.

It is essential in our approach to select an appropriate reconfiguration algorithm for constructing the fault-free logic array. Among others, the algorithms proposed by Youn and Singh [5] seems a good candidate because it provides the highest fault-tolerant capability with reasonable interconnection complexity, even in the presence of clustering faults. Youn and Singh suggested a grid method by considering that an imaginary grid array is laid over the host array. The desired array can be constructed by assigning fault-free nodes to suitable imaginary grid points. The assignment starts with either the topmost row or the leftmost column of the imaginary grid array, based on the pattern of failed nodes. A fault-free node from the adjacent rows or columns in the host array is assigned to a grid in the selected row or column. The remaining available nodes can be used later for the assignment of the next row or column of imaginary grids. The same strategy is applied to the rest of the array until all grids are assigned. A complete reconfiguration algorithm can be found in [5], and we adopt this algorithm for reconfiguring subtrees in our design.

As shown in [5], an $N \times N$ logic array can always be reconfigured successfully from an $(N+1)\times(N+1)$ host array with maximally $2N+1$ faulty nodes, regardless of the exact locations of the faults. Fig. 2 illustrates how an 8×8 logic array is obtained from a 9×9 host array in the presence of 17 ($= 2N+1$) faulty nodes. The tree embedding pattern given in Fig. 1 is then mapped onto the logic array of Fig. 2. The actual interconnection for the reconfigured 6-level binary tree is given in Fig. 3. It is observed that, for a small tree (with, say, six levels or less), two horizontal and two vertical interconnection tracks respectively between adjacent rows and adjacent columns are sufficient for reconfiguring the tree.

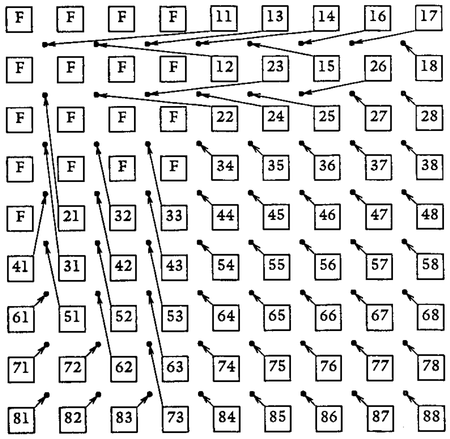

Fig. 2. Reconfiguring an 8×8 array out of a 9×9 host array [5]. (The number of a PE denotes its mapped location.)

The upper subtree can be laid out using the H-tree strategy, with each lower subtree considered as a "big" leaf node. Various fault-tolerant schemes may be employed for the upper subtree defect/fault tolerance. Clustering defects are usually tolerable in the upper subtree because its nodes are relatively widely dispersed. In this paper, we assume that all nodes (except for those "big" leaf nodes) in the upper subtree of both our approach and the Cluster-Proof approach are duplicated. This assumption simplifies the yield and reliability analysis, and makes possible the comparison between the two approaches.

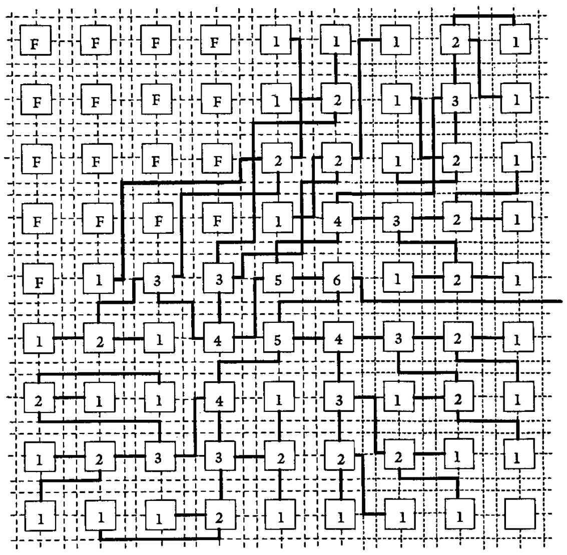

Fig. 3. Constructing a 6-level binary tree in the mapped logic array (depicted in Fig. 2) following the embedding shown in Fig. 1.

It should be noticed that since the Cluster-Proof approach is superior to the Modular and the SOFT approaches as illustrated in [2], we compare our scheme only with the Cluster-Proof approach, in terms of yield, area utilization and interconnection complexity. The fault-tolerance of the lower subtrees in both schemes are achieved by means of global spares. Therefore, these two approaches would have the same fault-tolerant capability. Area utilization here is defined as the ratio of the actual area occupied by the nodes (including spares) to the area required for the binary tree layout. Area for switches and links is assumed negligible to simplify our comparison. This assumption, in fact, favors the Cluster-Proof approach because it involves more links and switches. For an n-level binary tree, area utilization, U, is expressed as

$$U = \frac{2^n - 1 + S}{A}, \tag{1}$$

where S is the number of spare nodes and A is the silicon area required for the layout. For the Cluster-Proof approach, area A is given by

$$A = \begin{cases} (2^{(n+1)/2} - 1)^2, & \text{if } n \text{ is odd} \\ (2^{(n+2)/2} - 1)\times(2^{n/2} - 1), & \text{if } n \text{ is even.} \end{cases} \tag{2}$$

In our approach, area A depends on the size of the host array for lower subtrees. If the lower subtree involves k levels and is embedded in an $a \times b$ array ($a \geq b$), the area required for an n-level tree is

$$A = \begin{cases} ((a+1)2^{(n-k-1)/2} - 1)\times((b+1)2^{(n-k+1)/2} - 1), & \text{if } n-k \text{ is odd} \\ ((a+1)2^{(n-k)/2} - 1)\times((b+1)2^{(n-k)/2} - 1), & \text{if } n-k \text{ is even.} \end{cases} \tag{3}$$

The comparison results of 10-level binary trees are listed in Table 1. From the table, it can be seen that our approach consistently requires a much less number of switches than the Cluster-Proof scheme. Although more redundant links are needed in our scheme for systems with 4- and 5-level lower subtrees, it should be noted that a link in our approach is shorter (see Fig. 3), and thereby introduces less area, implying that the total area for links may be comparable. When 6-level lower subtrees are employed, it becomes clear that the proposed approach is superior to the Cluster-Proof approach both in redundant switches and in redundant links. Area utilization is an important index that reflects the manufacturing cost. As illustrated in Table 1, our approach always provides higher area utilization than its counterpart because its lower subtrees are laid out more efficiently.

Table 1

Comparison of the proposed approach and the Cluster-Proof approach

size of lower subtrees	spares per lower subtree	scheme	no. of switches	no. of links	area utilization
4-level	4	proposed	10807	15549	73.2%
		C-P	11831	4029	68.7%
5-level	8	proposed	10519	15363	79.6%
		C-P	19863	8029	67.1%
6-level	17	proposed	10503	15597	86.1%
		C-P	38167	17181	67.1%

3. Yield and Reliability Analysis

Yield Analysis

To evaluate the yield enhancement of the binary tree architecture, we assume that the system consists of identical nodes and each node/spare takes a unit silicon area. Let Y_1 denote the yield (the probability of being defect-free) of a single node, and each node be defective independently. The interconnect area of a system is considered as *kill area* with the yield of Y_0. For an n-level binary tree architecture comprising $N = 2^n-1$ nodes and S spares, yield of a chip with exact i faulty nodes, $Y(i)$, is

$$Y(i) = Y_0 C_i \binom{N+S}{i} Y_1^{N+S-i}(1-Y_1)^i, \tag{4}$$

where C_i is the coverage factor, which is the probability that a system reconfigures successfully in the presence of i defective nodes. In fact, the coverage factor is the ratio of the number of all tolerable fault patterns to the number of all possible fault patterns. The coverage factors of our scheme and the Cluster-Proof approach are identical and can be calculated by an enumeration approach. By taking the area utilization and the extra area needed for redundancy into account, the equivalent yield (Y_e) is given by

$$Y_e = \frac{N}{N+S} \times U \times \sum_{i=0}^{S} Y(i), \tag{5}$$

where U is the area utilization given in Eq. (1).

Although the Cluster-Proof approach provides the same $Y(i)$, our scheme has higher equivalent yield because of its better area utilization (see Table 1), leading to a lower manufacturing cost. The equivalent yields of a 10-level binary tree, as a function of Y_1, for the proposed scheme is shown in Fig. 4, where Y_0 is assumed to be 0.9. It can be seen that a system with larger lower subtrees has higher equivalent yield due to more global spares and higher area utilization. The price paid for higher Y_e, of course, is more complicated interconnection and reconfiguration process.

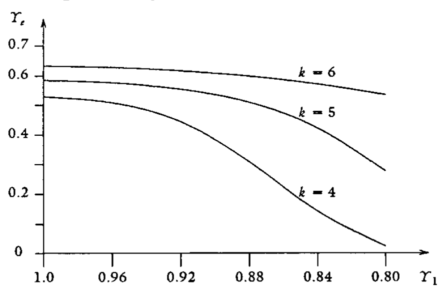

Fig. 4. Equivalent yield for 10-level binary trees with k-level lower subtrees (the number of spares follows Table 1). ($Y_0 = 0.9$)

Reliability Analysis

The use of the coverage factor makes it possible to analyze the system reliability by means of the Markov model. Unlike previous reliability studies in which chips are assumed defect-free, this analysis considers the fact that an accepted chip may have used spares to replace manufacturing defects, and the number of spares available for tolerating operational faults may thus vary from chip to chip. The Markov model of a system with i initial manufacturing defects is shown in Fig. 5. The state in the model, say state (k), means that the system is operational in the presence of k faults (including defects).

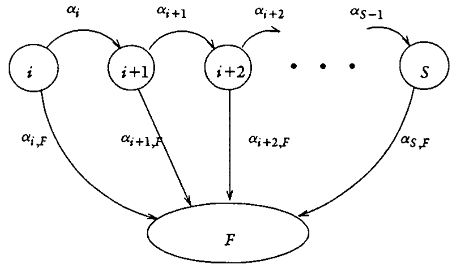

Fig. 5. A Markov model for binary tree architectures.

We denote α_k and $\alpha_{k,F}$ as the transition rates from state (k) to state $(k+1)$ and to the failed state (F), respectively. The transition rates depend on the failure rate of each node, and on the coverage factor of the system. The system can tolerate $k+1$ faulty nodes only when the system already covered k failed nodes and can successfully reconfigure in response to an additional fault. Let β_k denote the probability of successful reconfiguration with respect to an additional fault. We have $C_{k+1} = C_k \times \beta_k$, or alternatively

$$\beta_k = C_{k+1}/C_k. \tag{6}$$

Assume that the entire interconnection of a system is treated as a kill area with a failure rate of λ_0, each nonfaulty node has the same failure rate λ_1, and faults happen to nodes independently. The transition rates from state (k) to state $(k+1)$ and to the failed state (F) are given respectively by

$$\alpha_k = \beta_k(N+S-k)\lambda_1, \tag{7}$$

and

$$\alpha_{k,F} = \lambda_0 + (1-\beta_k)(N+S-k)\lambda_1. \tag{8}$$

Let $P_{i,k}(t)$ be the state probability, i.e.,

$$P_{i,k}(t) = \Pr\{\text{The system is in state } (k) \text{ at time } t \ / \text{ Its initial state is } (i)\}.$$

The differential equations for this Markov model are written as follows:

$$\frac{dP_{i,i}(t)}{dt} = -(\alpha_i + \alpha_{i,F})P_{i,i}(t), \tag{9}$$

and

$$\frac{dP_{i,k}(t)}{dt} = -(\alpha_k + \alpha_{k,F})P_{i,k}(t) + \alpha_{k-1}P_{i,k-1}(t), \tag{10}$$

where we have assumed that the initial state is (i) such that $P_{i,i}(0) = 1$ and $P_{i,k}(0) = P_{i,F}(0) = 0$. Using the Laplace transform technique, we can solve these state equations and obtain

$$P_{i,i}(t) = e^{-(\alpha_i+\alpha_{i,F})t}, \tag{11}$$

and

$$P_{i,k}(t) = \prod_{a=i}^{k-1}\alpha_a \sum_{j=i}^{k} \frac{e^{-(\alpha_j+\alpha_{j,F})t}}{\prod_{b=i,b\neq j}^{k}(j-b)\lambda}. \tag{12}$$

The reliability of chips with i initial faults (i.e., defects) is

$$R_i(t) = \sum_{k=i}^{S} P_{i,k}(t). \tag{13}$$

Since $Y(i)$ is the percentage of acceptance of chips with i initial faults, the average reliability of chips can be calculated by

$$R(t) = \frac{\sum_{i=0}^{S} Y(i)\, R_i(t)}{\sum_{i=0}^{S} Y(i)}. \tag{14}$$

The average reliability of a 10-level tree with 4, 5, and 6-level lower subtrees as a function of the mission time t is given in Fig. 6. The results are derived by assuming the failure rates $\lambda_0 = 0.2$ and $\lambda_1 = 0.1$ per unit time (e.g., 10^6 hours). It is observed that a system with larger lower subtrees has high reliability, which coincides with our argument. Again, the high reliability is achieved at the cost of increased reconfiguration complexity.

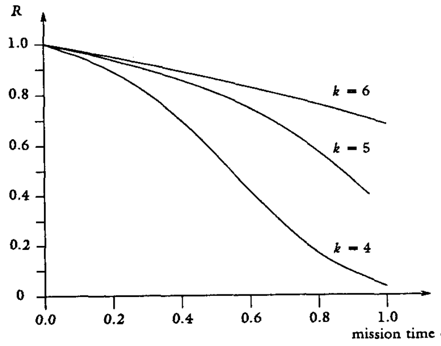

Fig. 6. Reliability of a 10-level reconfigurable binary tree with k-level lower subtrees ($Y_1 = 0.99$, $\lambda_0 = 0.2$, and $\lambda_1 = 0.1$).

We also observe that the yield Y_1 affects the average reliability of manufactured chips, which is an important point that has been neglected by previous researchers. Fig. 7 illustrates the average reliability of a 10-level binary tree for different values of Y_1. The curve with $Y_1 = 1$ represents the reliability of defect-free chips. It is shown that a system with a lower Y_1 not only has low equivalent yield (see Fig. 4), but also gives rise to low reliability of produced chips. This is because, for a lower Y_1, more spares have been used to replace the defective nodes during the fabricating process, and thereby the number of remaining spares available for tolerating operational faults decreases, making system reliability reduced.

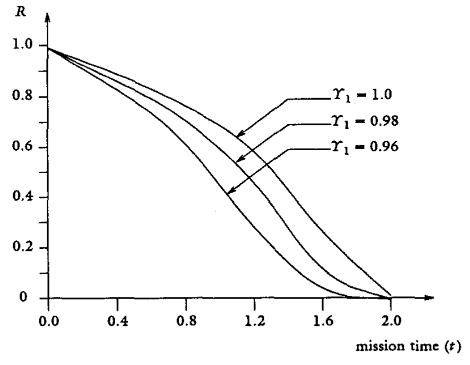

Fig. 7. Reliability of a 10-level reconfigurable binary tree for different values of Y_1 (with 6-level lower subtrees). (λ_0=0.2, λ_1=0.1)

4. Conclusions

An area-efficient approach to the design of reconfigurable binary tree architectures has been presented. We have shown that our approach is superior to the Cluster-Proof approach in terms of equivalent yield, interconnection complexity and area utilization. In the reliability analysis, we take into account the fact that accepted chips are with different degrees of available redundancy initially, and thus obtain results that better reflect actual situations. The developed analytical model for reliability is readily extended to other VLSI/WSI-based multiprocessor systems.

References

[1] M.B. Lowrie and W.K. Fuchs, "Reconfigurable Tree Architectures Using Subtree Oriented Fault Tolerance," *IEEE Trans. on Computers,* vol. C-36, Oct. 1987, pp. 1172-1182.

[2] M.C. Howells and V.K. Agarwal, "A Reconfiguration Scheme for Yield Enhencement of Large Area Binary Tree Architectures," *IEEE Trans. on Computers,* vol. C-37, Apr. 1988, pp. 463-468.

[3] H.Y. Youn and A.D. Singh, "On Implementing Large Binary Tree Architecture in VLSI and WSI" *IEEE Trans. on Computers,* vol. C-38, Apr. 1989, pp. 526-537.

[4] E. Horowitz and A. Zorat, "The Binary Tree as an Interconnection Network: Applications to Multiprocessor Systems and VLSI," *IEEE Trans. on Computers,* vol. C-30, Apr. 1981, pp. 247-253.

[5] H.Y. Youn and A.D. Singh, "An Efficient Channel Routing Algorithm for Defective Arrays" *IEEE Int'l. Conf. CAD,* Nov. 1989, pp. 432-435.