Bridging Fault Diagnosis in the Absence of Physical Information

David B. Lavo* Brian Chess† Tracy Larrabee* F. Joel Ferguson*

Jayashree Saxena‡ Kenneth M. Butler‡

*Computer Engineering
University of California
Santa Cruz, CA 95064

†Hewlett-Packard Corp.
Integrated Circuit Business Div.
Palo Alto, CA 94306

‡Texas Instruments Inc.
P.O. Box 660199 MS 8645
Dallas, TX 75266-0199

Abstract

Effective bridging fault diagnosis requires reducing the $\binom{n}{2}$ number of bridging faults to a handful of candidates. A preliminary step can reduce the $O(n^2)$ candidates to a manageable $O(n)$ candidates by using layout information to eliminate those bridging faults that are very unlikely to be shorted together. This step removes from consideration those faults that match the fault signature but are physically impossible. However, sometimes—perhaps due to issues of intellectual property or because the degree of information stored about a circuit changes over its lifecycle—the physical design of the circuit is not available, and the number of nodes is too large to explicitly consider all pairs. In this paper we present two ways to provide successful diagnoses without access to physical information. The second method produces optimal diagnoses under our ranking criteria. Either technique can be used in conjunction with information extracted from the physical design to allow for diagnoses of much larger circuits than previously possible.

1 Introduction

The purpose of fault diagnosis is to identify the location of a fault so that the cause of the fault can be categorized. This is necessary for several steps in the manufacturing process: the initial system debug, the ramp to volume production, the yield improvement phase, and volume manufacturing testing. As IC manufacturing technology becomes more complex and feature sizes continue to shrink, the manual search and categorization of defects is becoming exceedingly expensive and time-consuming. The expense of manual search makes it crucial for automated diagnosis tools to pinpoint the location of a defect to only a few locations.

Because bridging faults are believed to be a common defect type [9, 16], bridging fault diagnosis is crucial to manufacturing diagnosis and debug. A circuit with n nodes has $\binom{n}{2}$ possible bridging faults; explicit consideration of all such faults is infeasible. There have been two approaches to this *candidate-space problem*: the first approach assumes that all $\binom{n}{2}$ possible bridging faults for a circuit need to be considered [7]. Building an $\binom{n}{2}$-sized fault dictionary is prohibitively expensive, so this line of research has focused on algorithms that continuously eliminate large portions of the candidate space based on the observed fault signature (without building a dictionary) [5, 6]. Once a bridging fault is removed from the candidate space it is no longer considered.

A major weakness of this approach is that if the bridging fault's behavior is not well-characterized, it is likely to be removed from the diagnosis. A major strength is that the physical design of the circuit is not necessary for diagnosing potential bridging faults.

The second approach uses the physical design of the circuit to eliminate bridging faults between lines that are extremely unlikely to be shorted together due to the physical location of the nodes comprising the bridging fault [2, 8]. If the two nodes are never closer than some minimum distance, or if there is another node separating them that would also be involved in the bridge, then that bridging fault is not considered[12]. Errors in understanding and predicting the bridging fault behavior are tolerated by finding the best match to the observed signature.

An opportunity arose for the UCSC physical design-based diagnosis software to be tested on Texas Instruments chips. However, Texas Instruments was not able to supply the physical design of the chip to UCSC due to intellectual property constraints. Therefore, the diagnosis procedure had to begin with $\binom{n}{2}$ bridging faults in a subcircuit with 20,000 nodes. We needed to pare down the candidate space in a way that would not inadvertently remove from consideration a bridging fault if its behavior varied slightly from the expected. In effect, this would eliminate what we see as the greatest weaknesses of the published $\binom{n}{2}$ approaches while keeping their greatest strength.

In this paper we present two solutions to the candidate-space problem that are tolerant of deviations in expected fault behavior. The first of our methods is intuitively natural, but yields worse results than the second. Our second method implicitly considers all $\binom{n}{2}$ faults and produces an optimal diagnosis, given our ranking criteria, in the absence of physical information. Section 2 explains our solutions to the candidate-space problem, and Section 3 provides an analysis of the success of our experiments.

2 Bridging fault diagnosis algorithms

Most commercial automated diagnosis tools rely on the stuck-at fault model as a basis for fault diagnosis. However, it has been shown repeatedly that the stuck-at fault model does not accurately reflect the behavior of current-generation silicon defects [3, 9–11, 14–17]. While a more realistic fault model provides better diagnoses, a bridging fault diagnosis algorithm that uses the stuck-at fault model as a foundation does not require a sacrifice of performance or a change in existing design flows.

The use of stuck-at fault information to perform bridging fault diagnosis [5, 8, 13, 14] is based on a somewhat simplified view of bridging fault behavior: In order for a bridging fault to be detected, the two bridged nodes must have opposite values in the fault-free circuit, but in the presence of the fault both bridged nodes will have the same value (one of the two nodes will dominate). This means that any vector that detects a bridging fault will detect one of the four stuck-at faults associated with the two nodes. Stanford [14] and UCSC [8] have produced bridging fault diagnosis methods where the signatures for all four associated stuck-at faults are concatenated: the bridging fault signature must be included in the resulting *composite signature*. Figure 1 shows how a candidate fault behavior is compared against an observed faulty behavior. Note that because composite signatures are meant to over-predict bridging fault behavior (they are inclusive), we expect mispredictions, but non-predictions are much less likely in a successful candidate.

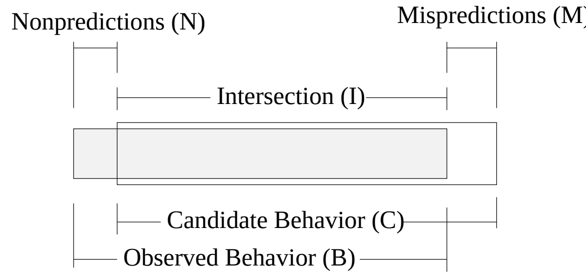

Figure 1: Overlapping of candidate behavior (unshaded) and observed behavior (shaded).

2.1 Candidate list reduction via stuck-at preselection

The first procedure for reducing the candidate space is based on the empirical observation that diagnosis with stuck-at candidates has traditionally proved to be effective in identifying at least one circuit node involved in a bridging fault. Knowing one involved node in a bridging fault reduces the search space: given a set of d candidates for the first node, only dn candidates need be considered.

We ran a set of experiments on the ISCAS-85 [4] circuits to verify the efficacy of stuck-at diagnosis in identifying one of a pair of bridged nodes. These trials simulated and diagnosed the top 10% of realistic bridging faults (from 160 for the C432 to 5379 for the C7552). Carafe [12] identified the most likely bridging faults, and our fault simulator, Nemesis, simulated them (taking into account the Byzantine Generals Problem for bridging faults [1]).

We then ran these simulated bridging fault behaviors through a standard stuck-at fault diagnosis procedure. Different weightings of misprediction and nonprediction penalties were used; Table 1 reports the results from the most successful—equal weightings. The ten candidates with the lowest combined number of mispredictions and nonpredictions constitute the stuck-at diagnosis. Table 1 shows the percentage of size-ten diagnoses that contain at least one of the two nodes involved in the bridging fault. A stuck-at diagnosis of size ten contains one of the two involved nodes in a bridging fault roughly 90% of the time[1].

To produce a bridging fault candidate list from a stuck-at diagnosis, the first diagnosis procedure pairs each member of the stuck-at diagnosis with every other node in the circuit. It then compares the composite bridging fault signature created for each pair to

[1] It is interesting to note that the second node in the bridging fault is rarely included in the diagnosis [8].

Circuit	% Success
C432	98.1
C499	83.9
C880	98.8
C1355	83.8
C1908	92.1
C2670	96.0
C3540	99.5
C5315	97.8
C6288	88.4
C7552	94.1

Table 1: Stuck-at diagnosis on bridging fault behaviors (d=10): percentage of diagnoses where at least one of the nodes is in the diagnosis.

the observed behavior, and the top-scoring D bridging fault candidates make up the final diagnosis.

Given an initial stuck-at diagnosis size of d, this procedure reduces the exhaustive list of size $\binom{n}{2}$ by $\binom{n-d}{2}$, the number of pairs that do not contain at least one node from the stuck-at diagnosis. This means that

$$\binom{n}{2} - \binom{n-d}{2} = dn - \frac{d(d+1)}{2} \qquad (1)$$

composite signature constructions and comparisons must be performed. Unfortunately, this reduction in candidate-space is achieved at the expense of potential successful diagnoses: the percentage of diagnoses where at least one of the nodes is in the diagnosis, as given in Table 1, sets an upper limit on the eventual success rate of the bridging fault diagnosis. Since only the top d nodes are paired, bridging faults that do not involve one of these nodes will not be considered.

Table 2 first reports the result of running our diagnosis software [8, 13] using a realistic fault list: it gives the number of realistic bridging faults in the candidate fault list, and the percentage of successful diagnoses of size 10 using the realistic candidate list. In contrast, the number of composite signatures produced via stuck-at preselection is generally significantly smaller than the number of realistic faults considered, but stuck-at preselection produces fewer successful diagnoses. Note that for the unsuccessful diagnoses, even increasing the final diagnosis size D to 100 is not often helpful; if one of the two bridged nodes is not included in the original stuck-at preselection, a successful diagnosis can never be performed—regardless of the size of the final diagnosis.

2.2 Candidate list reduction via node preselection with scoring thresholds

The second procedure for overcoming the candidate-space problem is based on the observation that diagnosis systems set limits, or thresholds, on the candidates that qualify for the final diagnosis: Either the top-scoring D candidates are included in a diagnosis, or only those candidates that score at or above a threshold I_t are included. In either case, the candidates are sorted by their scores, and only a small number of the $\binom{n}{2}$ candidates are retained. Performing the scoring and sorting intelligently can drastically reduce the number of candidates considered.

Because we know that composite bridging fault signatures are constructed with deliberate overprediction, we consider containment, or intersection with the observed behavior, to be the primary indicator of candidate goodness; the best candidate is the one with the greatest intersection [13]. In the case of intersection ties, we consider the number of required vectors matched and the amount of misprediction, in that order. Because our ordering is lexicographic, we can reduce our scoring criteria to the primary parameter: a candidate with greater containment than another is always ranked higher for a particular diagnosis. Given this fact, it is a simple matter to set a threshold for a diagnosis: either the top D candidates, by size of intersection, are reported, or all candidates that have an intersection greater than or equal to a threshold, I_t, make up the diagnosis.

Having reduced the scoring to a single primary parameter, there is an opportunity to reduce the candidate search space significantly—if the elements that contribute to this parameter can be reduced to a manageable number. To do this, consider again the construction of a composite bridging fault signature from four stuck-at signatures, as illustrated in Figure 2.

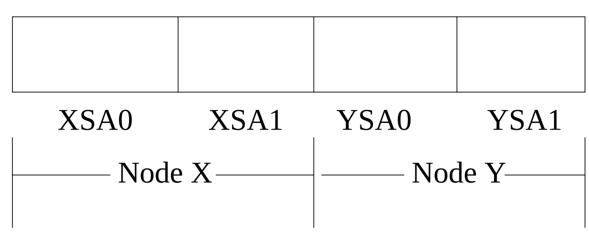

Figure 2: The composite signature is constructed from four stuck-at signatures or two node signatures.

Note that while there are four stuck-at signatures involved, there are only two nodes, each of which can be characterized as the union of its stuck-at-0 and

Circuit	Realistic List		Stuck-at Preselection		
	Comp. Sigs	% Success ($D = 10$)	Comp. Sigs	% Success ($D = 10$)	% Success ($D = 100$)
C432	1,595	98.1	1,615	74.7	97.5
C499	2,781	100.0	2,145	83.9	83.9
C880	3,275	98.8	2,665	95.4	98.2
C1355	4,422	98.9	4,005	82.9	83.1
C1908	4,744	91.3	3,355	75.2	89.2
C2670	13,710	94.8	6,885	81.8	92.5
C3540	16,459	96.2	7,655	89.0	96.0
C5315	40,430	97.7	13,175	97.0	97.0
C6288	21,911	99.8	18,715	88.4	88.4
C7552	53,789	94.7	16,800	83.0	89.9

Table 2: Number of composite signatures built for realistic faults, percentage of successful realistic diagnoses ($D = 10$), number of composite signatures, and percentage successful stuck-at preselection diagnoses for final diagnosis sizes of $D = 10$ and $D = 100$.

stuck-at-1 signatures. We call such a union of the stuck-at-0 and stuck-at-1 signatures for a single node a *node signature*. A composite bridging fault signature is the union of two node signatures. Note also that while there are $\binom{n}{2}$ possible composite bridging fault signatures for an n-node circuit, there are only n node signatures. Using this set of node signatures as a reduced search space, the second procedure attempts to eliminate the vast majority of low-scoring composite bridging fault candidates that could be built from these n nodes.

Consider first the scenario in which a scoring threshold, I_t, is set for a diagnosis; this scoring is a minimum acceptable intersection for a composite bridging fault candidate. The key observation behind the second procedure is that for a bridging fault candidate to have an intersection I_t with the observed behavior, one of its component node signatures must have an intersection of at least half of I_t. Any possible bridging fault candidate without at least one of its two node signatures scoring at least half of I_t cannot appear in the final diagnosis, and therefore need not be constructed or compared.

In order to produce a bridging fault candidate list, we construct the set of n node signatures, and we perform a diagnosis of the observed behavior in which the intersection of each node candidate is measured. The node candidates are then sorted by this intersection value, and composite bridging fault signatures are constructed only for pairs where the sum of the intersections for the two nodes is at least I_t.

As an example, Figure 3 shows the result of ranking a set of node signatures according to their intersection. If all composite bridging faults in the final diagnosis are required to contain at least 95% of the observed faulty behavior, then the procedure need only consider node pairs where one of the two nodes is from the set comprised of A, B, and C (the set with greater than 47.5% intersection), because no pairing created from the remaining nodes can possibly contain 95% of the observed faulty behavior. Composite bridging fault signatures should be created for the five node pairs A-B, A-C, A-D, B-C, and B-D.

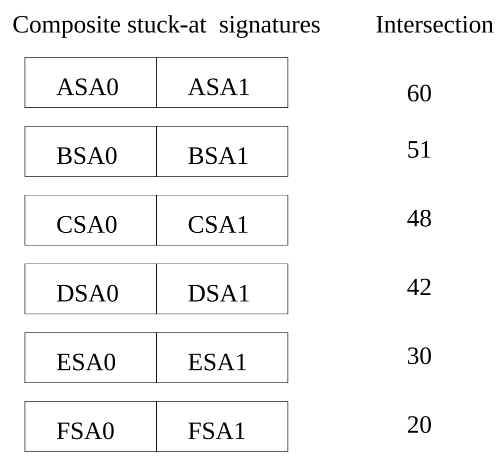

Figure 3: Several node (or *composite-stuckat*) signatures are ranked by their comparison with the observed behavior.

For this method, if there are k node candidates with an intersection of at least half of I_t, at most

$$\binom{n}{2} - \binom{n-k}{2} = kn - \frac{k(k+1)}{2} \qquad (2)$$

comparisons and constructions will be performed.

This equation is the same as Equation 1 (with all instances of d replaced by k), but unlike Equation 1, this is a loose upper bound, and we expect to actually create many fewer candidates. For the example in Figure 3, $n = 6$ and $k = 3$ giving a bound of 12, but only 5 composite bridging fault signatures will be created. This procedure guarantees that all composite bridging fault candidates with $I \geq I_t$ will be constructed.

In order to get an idea of the magnitude of k, and an upper bound on the success rate, we diagnosed the same behaviors from the ISCAS-85 circuits mentioned above, but we used node candidates instead of stuck-at candidates. Instead of a diagnosis size d, a scoring threshold I_t was set, and the average number of qualifying node candidates, or k, was recorded. Table 3 gives results analogous to those presented in Table 1: the percentage of the node diagnoses (for $I_t = 95$) that contain at least one of the two bridged nodes. Note that the upper bound for successful diagnosis using this method is much higher than for the previous method.

	Node $I_t = 95$	
Circuit	Success Bound	k
C432	100.0	15.0
C499	100.0	41.2
C880	100.0	8.6
C1355	100.0	102.5
C1908	99.6	31.0
C2670	99.9	26.0
C3540	100.0	26.1
C5315	99.9	9.2
C6288	99.7	79.1
C7552	99.5	16.2

Table 3: Bound on success rate and average values of k for node preselection with a scoring threshold of $I_t = 95$.

This approach can be modified to limit the number of composite signatures to at most D candidates. In this case, we proceed in exactly the same fashion, except the value of I_t is dynamic rather than static: it is the current score of the D^{th} composite bridging fault candidate, or $I(D)$. (Before D candidates have been constructed and scored, assume $I(D) = 0$.) Note that this procedure is guaranteed to produce the same diagnosis, using our scoring criteria, as that we could have achieved via explicitly considering all $\binom{n}{2}$ candidates.

Table 4 shows that, as expected, the diagnoses for node preselection with scoring are much more successful than the diagnoses for stuck-at preselection. In particular, the diagnoses of size 100 are very successful. The number of composite signatures created is generally less than or equal to the number of realistic faults created by Carafe for the same circuits.

3 Analysis

The most interesting and useful feature of the second method, node preselection with scoring thresholds, is the guarantee it provides about the candidates it considers: Any two-node bridging candidate that can be constructed from the stuck-at fault list that could score at or above a desired threshold will be constructed and scored. This is true despite the fact that many fewer than $\binom{n}{2}$ candidates will be constructed and matched.

If the diagnosis algorithm scores the correct candidate above this threshold, we are guaranteed to construct and rank the correct match during the diagnosis. This is not true of the first method of stuck-at preselection, because the initial stuck-at diagnosis may fail to identify one of the bridged nodes. Knowing this, the success rate of the stuck-at preselection method can be projected from the success rate of the node preselection method by the following relation:

$$S_S(D) \approx S_N(D) \cdot S_s(d) \tag{3}$$

where $S_S(D)$ is the success rate of the stuck-at preselection method for a final diagnosis size D, $S_N(D)$ is the success rate of the node preselection method for the same diagnosis size, and $S_s(d)$ is the success rate of the initial stuck-at diagnosis in identifying one of the bridged nodes.

Given that node preselection will report all qualifying candidates, two issues remain: what is a proper threshold to set, and what is the proper final diagnosis size?

If bridging fault behaviors were perfectly predictable, then the strictest intersection threshold, $I_t = 100$ would guarantee consideration of the correct match. Acknowledgment of the Byzantine Generals Problem for bridging faults, non-zero bridge resistance, and the possibility of other sources of noise in the behaviors argue for a lower threshold. For this work $I_t = 95$ was chosen as an informed but arbitrary value; adjustments to this threshold value remain a topic for further research.

The choice of diagnosis size is perhaps more complicated. The results given in Table 4 indicate that the standard choice of 10 candidates may not succeed in all situations. In examining those diagnoses that

Circuit	Node Preselection			
	Comp. Sigs	% Success	Comp. Sigs	% Success
	($D = 10$)		($D = 100$)	
C432	879	76.0	1,117	99.4
C499	4,162	97.5	4,268	99.4
C880	624	96.0	1,042	98.8
C1355	17,529	98.9	21,035	99.2
C1908	3,518	79.6	4,573	96.8
C2670	5,085	83.1	6,611	96.4
C3540	3,281	89.0	4,271	96.2
C5315	2,489	97.0	3,302	97.7
C6288	33,603	99.7	40,324	99.7
C7552	4,537	84.0	5,211	94.7

Table 4: Number of composite signatures and percentage successful node preselection diagnoses for fixed diagnosis sizes of 10 and 100.

did not succeed, we found a strong correlation between the number of failing bits (observed errors) in a diagnosed behavior and the success of the diagnosis: the node preselection method performed poorly when the number of failing bits was small for the test set.

Given this fact, it seems obvious that to maintain a consistent confidence level in the diagnosis results, the number of candidates reported for a given behavior will have to depend upon the number of failing bits, or amount of failure information, available to the diagnosis program. For large diagnosis sizes, then, the diagnosis may be more suitable as a reduced candidate list for further fault discrimination than for actual physical failure analysis. Such further discrimination could include the creation and application of distinguishing test vectors or current measurements, or some probabilistic evaluation such as inductive fault analysis.

One final point is that either of our two methods could be used in conjunction with a realistic fault list in order to minimize the number of composite signatures created (the use of node preselection would guarantee no degradation in diagnoses, so it would seem the better choice). This means that these procedures make bridging fault diagnosis feasible for circuits for which there is no physical information, and they make bridging fault diagnosis workable for much larger circuits for which there is a realistic fault list.

4 Conclusion

This paper presents two techniques for reduction of the candidate space involved in bridging fault diagnosis without the aid of physical design information. The first uses the intuitive idea of an initial stuck-at fault diagnosis to attempt to identify one of the bridged nodes. The second technique identifies all candidates that can have an arbitrary intersection threshold with the behavior, providing an optimal diagnosis under our scoring criteria. Both techniques only consider and construct $O(n)$ candidates, about as many as previously reported when using realistic fault lists. Either technique could be used to minimize the construction of composite signatures even when beginning with a realistic fault list.

The results from the two techniques indicate that the second is much more successful at reporting the correct match, regardless of the final diagnosis size. In the best case, it can report the correct match in a small number of candidates; in the worst case, it can provide a larger number of candidates, with a high confidence of including the correct match, for possible further discrimination. The performance of the technique appears to depend heavily on the number of failing bits in the behavior to be diagnosed, a topic which we continue to investigate.

Acknowledgements

This work was supported by the SRC under contract 96-DJ-315 and by the NSF under grants MIP-9158491 and MIP-9158490. The authors thank Marvin Karlow of Texas Instruments.

References

[1] J. M. Acken and S. D. Millman. Fault model evolution for diagnosis: Accuracy vs precision. *Proceedings of the Custom Integrated Circuits Conference*, pages 13.4.1–13.4.4, 1992.

[2] R. C. Aitken and P. C. Maxwell. Better models or better algorithms? on techniques to improve fault diagnosis. *Hewlett-Packard Journal*, pages 110–116, February 1995.

[3] C. C. Beh, K.H. Arya, C. E. Radke, and K. E. Torku. Do stuck fault models reflect manufacturing defects? In *Proceedings of International Test Conference*, pages 35–42. IEEE, 1982.

[4] F. Brglez and H. Fujiwara. A neutral netlist of 10 combinational benchmark circuits and a target translator in fortran. In *Proceedings of the IEEE International Symposium on Circuits and Systems*, 1985.

[5] S. Chakravarty and Y. Gong. An algorithm for diagnosing two-line bridging faults in combinational circuits. *Proceedings of Design Automation Conference*, pages 520–524, 1993.

[6] S. Chakravarty and M. Liu. Algorithms for IDDQ measurement based diagnosis of bridging faults. *Journal of Electronic Testing: Theory and Applications*, 3(4):377–386, December 1992.

[7] S. Chakravarty and P. J. Thadikaran. Which set of bridging faults should test compilers target? Technical report, Department of Computer Science, SUNY Buffalo, February 1996.

[8] B. Chess, D. B. Lavo, F. J. Ferguson, and T. Larrabee. Diagnosis of realistic bridging faults with single stuck-at information. In *Proceedings of International Conference on Computer-Aided Design*, pages 185–192. IEEE, 1995.

[9] F. J. Ferguson and J. P. Shen. Extraction and simulation of realistic CMOS faults with inductive fault analysis. In *Proceedings of International Test Conference*, pages 475–484. IEEE, 1988.

[10] J. Galiay, Y. Crouzet, and M. Vergniault. Physical versus logical fault models in MOS LSI circuits: Impact on their testability. *IEEE Transactions on Computers*, C-29(6):527–531, June 1980.

[11] C. F. Hawkins, J. M. Doen, A. W. Righter, and F. J. Ferguson. Defect classes—an overdue paradigm for CMOS IC testing. In *Proceedings of International Test Conference*, pages 413–425. IEEE, 1994.

[12] A. Jee and F. J. Ferguson. Carafe: An inductive fault analysis tool for CMOS VLSI circuits. In *Proceedings of the IEEE VLSI Test Symposium*, pages 92–98, 1993.

[13] D. B. Lavo, T. Larrabee, and B. Chess. Beyond the Byzantine Generals: Unexpected behavior and bridging fault diagnosis. In *Proceedings of International Test Conference*, pages 611–619. IEEE, 1996.

[14] S. D. Millman, E. J. McCluskey, and J. M. Acken. Diagnosing CMOS bridging faults with stuck-at fault dictionaries. In *Proceedings of International Test Conference*, pages 860–870. IEEE, 1990.

[15] V. V. Nickel. VLSI—the inadequacy of the stuck at fault model. In *Proceedings of International Test Conference*, pages 527–531. IEEE, 1980.

[16] J. P. Shen, W. Maly, and F. J. Ferguson. Inductive fault analysis of MOS integrated circuits. *IEEE Design and Test of Computers*, 2(6):13–26, December 1985.

[17] R. L. Wadsack. Fault modeling and logic simulation of CMOS and MOS integrated circuits. *Bell System Technical Journal*, 57(5):1449–1474, May-June 1978.