ANALYSIS OF ACTUAL FAULT MECHANISMS IN CMOS LOGIC GATES*

Glenn R. Case
Sandia Laboratories
Albuquerque, New Mexico 87115

ABSTRACT

An analysis of failure modes in CMOS logic gates is presented. An example 3-input NAND gate is analyzed in detail and the ramifications of its failure modes are discussed.

Introduction

Testing and inspection of an integrated circuit (a microelectronic chip) is usually accomplished by visually inspecting the chip and electrical testing.[1,2] While different methods for electrical testing have been proposed,[3] the mainstay of most fabrication shops is to apply a number of predetermined input signal combinations to the chip (called a test sequence) and compare the measured output signals with an expected response. If a difference is noted, the chip can definitely be tagged as being faulty. If no difference is noted, the chip is presumed good, although the only definitive statement that can be made is that the probability of the chip being good is high.

Test sequences are usually developed using the "stuck-at" model originally proposed by Armstrong,[4] but discussed in many other references as well.[5,6] The objective of test sequence generation (whether done manually or using an automated algorithm) is to establish that a high percent of the singly occurring faults can be detected by the sequence. A designer typically attempts to detect 90 to 95% of the singly occurring faults.

The purpose of this investigation is to examine the ways a gate fabricated using a particular CMOS technology can actually fail, and to estimate a probability for the various failure modes. Prior knowledge that a certain gate is much more likely to fail one state as opposed to another can benefit the designer in that a more realistic test sequence may be developed. A second purpose of this investigation is to suggest methods for generating more efficient test sequences.

This paper will review the stuck-at model with the view of demonstrating how this model represents a subset of the general failure model. Next the CMOS technology is reviewed particularly from the standpoint of failure mechanisms. A detailed analysis of fault mechanisms in a representative CMOS gate is next performed. Finally, the results of the analysis are examined to see how such information could be incorporated into a test generation/evaluation scheme.

Review of the Stuck-At Model

A gate can be said to have failed if the response to input signals deviates from the gate's expected performance. Consider a simple two input gate as shown in Fig. 1. With two levels of logic, there are four input state possibilities: Input 1 = 0, Input 2 = 0 on through 1,0; 0,1; and 1,1.

If the correct and faulty response of the gate are considered, the gate can respond in 1 of 16 different ways (in general, 2**(2**N) where N is the number of inputs), which are indicated in Fig. 2. For convenience, the responses are numbered 0-15. The proper response for the AND gate is state number 1

Fig. 1. General Two Input Logic Gate

I_1	0	1	0	1	
I_2	0	0	1	1	
	0	0	0	0	- 0
	0	0	0	1	- 1
	0	0	1	0	- 2
	0	0	1	1	- 3
	0	1	0	0	- 4
	0	1	0	1	- 5
	0	1	1	0	- 6
	0	1	1	1	- 7
	1	0	0	0	- 8
	1	0	0	1	- 9
	1	0	1	0	- 10
	1	0	1	1	- 11
	1	1	0	0	- 12
	1	1	0	1	- 13
	1	1	1	0	- 14
	1	1	1	1	- 15

Fig. 2. All Possible Responses to a General Two Input Gate

(0001); for a NAND gate, the correct response is number 14 (1110), etc. In general, the number of deviate truth tables is 2**(2**N) - 1. (A deviate truth table is a response state other than the correct response.) The stuck-at model accounts for 2*N+2 of these in general, and in the case of AND, NAND, OR, and NOR gates, N+2 since the effect of, for example, sticking the input of an AND gate at 0 is the same as the gate sticking at 0.

In the example shown in Fig. 2 for a NAND gate, the proper response would be number 14. The four stuck-at faults map into four of the response states as follows:

Gate s-a-0	responds with	0000, no. 0.
Gate s-a-1	responds with	1111, no.15.
In 1 s-a-1	responds with	1100, no.12.
In 2 s-a-1	responds with	1010, no.10.

It should be noted that most of the actual faults map into these particular deviate responses. For the CMOS example to be analyzed later, a large percentage of the faults do in fact map into the "stuck-at" response although some do not. Any new technology should be examined to determine the degree of accuracy which can be associated with the fault model. A five input gate, for example, has 4.29 x 10^9 possible deviate truth tables; the stuck-at model accounts for only 7 of the states if the gate is an AND or OR gate.

*This work was supported by the United States Energy Research and Development Administration.

Analysis of a 3-Input NAND Gate Using CMOS Technology

Because CMOS is one of the primary technologies used in industry, a detailed analysis has been performed on the actual faults which can occur. Since the structure of all CMOS cells is similar, a representative 3-input NAND gate cell was chosen for the analysis. As noted in the previous section, the number of alternate responses or truth tables becomes very large as the number of inputs to a gate increases. The three input gate was chosen for the detailed study because it represents a reasonable compromise between circuit and logic complexity. However, the conclusions may not apply to all CMOS cells.

Before examining a faulted gate, the normal operation of the gate will be reviewed. The schematic for the three input CMOS NAND gate is shown in Fig. 3. It is noted that there are three p type transistors on the top of the gate in parallel and three n type on the bottom in series. To have a convenient way of referring to these transistors in subsequent discussion, the convention of pD or nD for "p device" or "n device" will be used followed by a parenthesis indicating which input they are associated with. Thus, pD(A) refers to the top p device associated with input A.

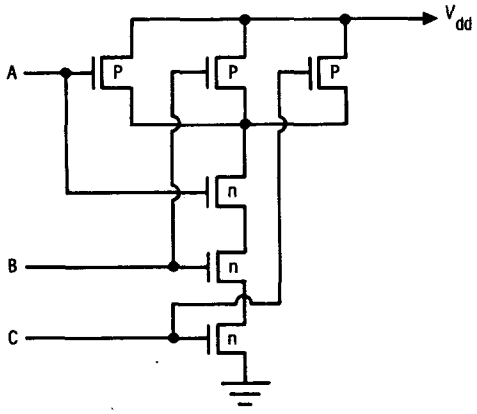

Fig. 3. Three Input NAND CMOS Gate Schematic

CMOS is very tolerant of supply voltage Vcc. Our analysis will consider 10 volts as an average value. The transfer curve for a typical CMOS inverter is shown in Fig. 4. Note that even over a considerable range of temperature, the cell has a very high gain. Any input voltage below 3.5 volts insures the output will be a logical 1, and any voltage input greater than 6.5 volts insures the output will be a logical zero.

For a logical 0 input, the corresponding p channel transistor is conducting and the n channel device is off; the output is a logical 1. If all inputs are 1, the bottom string of transistors are all on, the top devices are all off, and the output is a logical 0. This is summarized in Fig. 5.

To fabricate a CMOS cell seven masking steps are required. The basic cross sectional structure for a simple CMOS inverter is shown in Fig. 6. Note that a p well is required in the n substrate to form the n devices. The actual masks for the cell of Fig. 3 are shown in Fig. 7. The steps (corresponding to the masks) are as follows:

(1) p well diffusion for the n-channel devices

(2) p^+ diffusion for the guard band and the source and drain for the p-channel devices

(3) n^+ diffusion for the guard band and the source and drain for the n-channel devices (negative mask shown)

(4) Gate oxide windows

(5) Metallization contact openings in oxide

(6) Metallization paths

(7) Areas for bonding pads

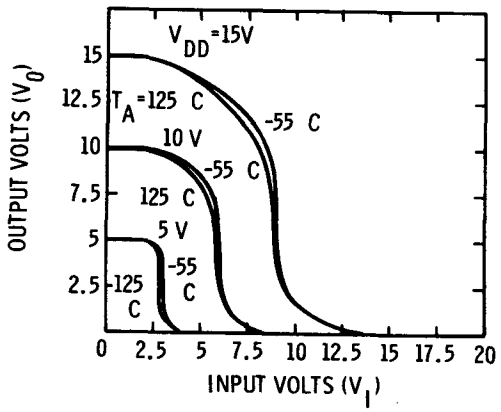

Fig. 4. Transfer Characteristic of a COS/MOS Logic Gate (From Ref. 2 - Used With Permission)

INPUT A	0	1	0	1	0	1	0	1
INPUT B	0	0	1	1	0	0	1	1
INPUT C	0	0	0	0	1	1	1	1
UNFAULTED OUTPUT	1	1	1	1	1	1	1	0

Fig. 5. Normal Response of a 3-Input NAND Gate

Failure Modes for the CMOS Gate

Failures in a CMOS cell can arise from one of three major mechanisms.

(1) Photolithography errors which can cause source-drain shorts, oxide pinholes or missing features.

(2) Diffusion errors produced by contaminants which result in shorted regions.

(3) Metallization errors resulting in either an open metal line or a short between lines.

Although each defect is assumed to cause a single fault, a fault can affect more than one transistor or interconnection on a chip. The usual assumption, however, is that if a test sequence can detect a single failure, multiple failures will also be detected. Experience has shown this to be a good rule of thumb. The analysis to follow assumes that a single failure has occurred.

The failure mechanisms cited above can precipitate many types of failures on the chip, but if single failures only are considered, the following types are presented:

(a) A transistor acting as an open or short circuit

(b) An open input to a gate causing the input to drift to a s-a-1 or s-a-0 condition (undefined)

(c) An open input to a transistor gate causing that

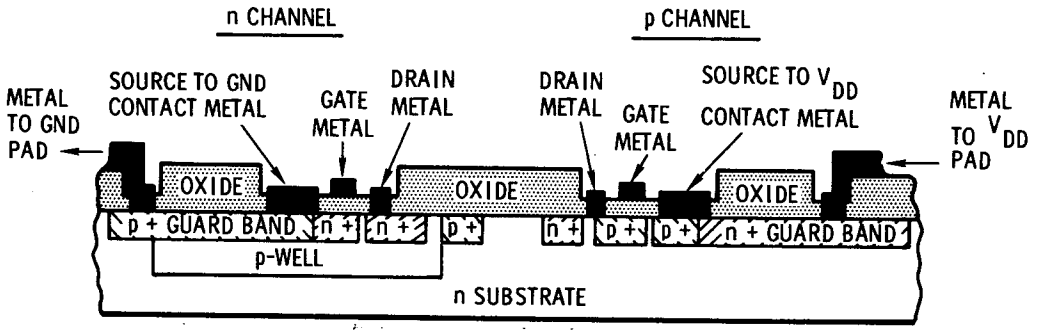

Fig. 6. Cross Section of a CMOS Inverter (From Ref. 2 - Used With Permission)

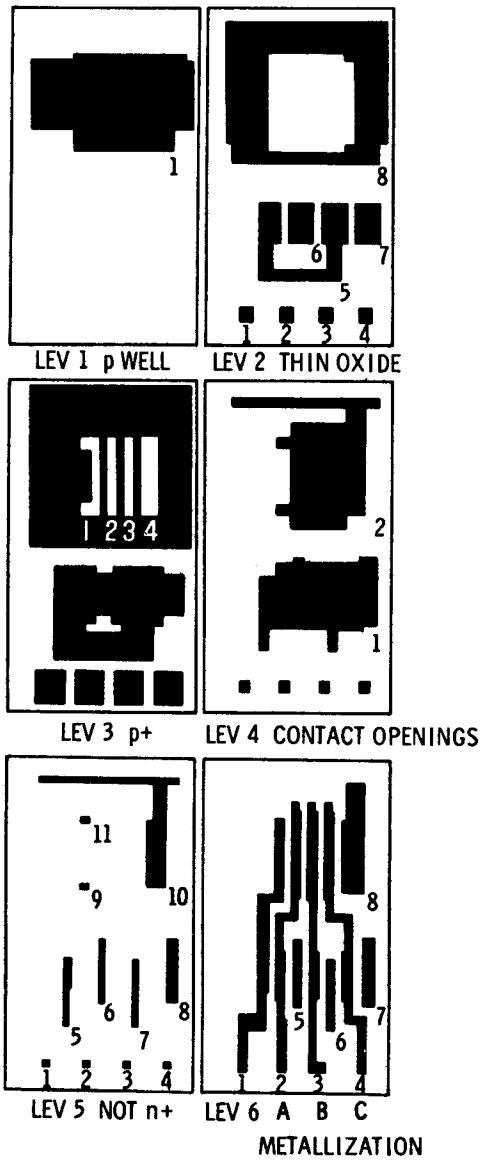

Fig. 7. Masks For Making 3-Input NAND Gate

device to drift to a permanent open or short condition. (This is treated electrically exactly the same as (a)).

(d) Shorts between gate inputs or shorts from input to output.

Some of the failures outlined above will map into a unique truth table. Input A s-a-1, for example, maps uniquely into the truth table 1111 1100. In other cases, however, the response of the gate to certain inputs will either be an open circuit or be represented by a string of conducting transistors from Vcc to ground. In either case, the output is undefined in the sense that it can act either as a 0 or a 1. In this case, the response will be represented by an * for the undefined response. As an example 1111 1*11. If the likelihood of the * going to 0 is the same as 1, no further notation will be used. If it is estimated that the probability of a 1 is different from 0.5, that value will be noted in parenthesis. For example 1111 1*(.857)10.

In the next section, the actual failure modes will be considered in detail. Certain failures, however, are repeated many times and will be considered generally in this section so that only a reference is required in the next section.

pD(A,B, or C) shorted will affect the gate only when all inputs are 1. In this case, there will be a path to Vcc when all three bottom n devices are on, resulting in an undefined output. This results in truth table 1111 111*.

If one of the p devices is open, the result will be an open circuit for a certain combination of inputs. This will have the effect of "holding" the charge on the gate of the succeeding stage at its previous value. If for example, pD(A) were open, then the response would be 1111 11*0. The probability is higher that the gate would be switching to the * from 1 rather than 0 since 6 input combinations will yield a 1 whereas only one will yield a 0. The 1 will then be weighted 6/7 = .857 and the table will then be 1111 11*(.857)0. Using this reasoning for the other p devices the responses are summarized below.

pD(A) open results in response 1111 11*(.857)0.
pD(B) open results in response 1111 1*(.857)10.
pD(C) open results in response 111*(.857) 1110.

Any of the n devices open, presents an open circuit on the output anytime the input is all 1's. The only existing output state, however, is the 1 state and so the gate effectively is stuck at 1. It can be argued that the gate can always have all 1's applied or that if the all 1 state persists for a long time the output could drift to 0. These possibilities are considered to be so unlikely that the assumption of s-a-1 does not appear to be unjustified. Hence,

nD(A,B,orC) open results in response 1111 1111.

Any of the n devices shorted will have the effect of providing a continuous path from Vcc to ground for a certain combination of inputs. The output will be undefined. The results are summarized in the following:

nD(A) shorted results in response 1111 11*0.
nD(B) shorted results in response 1111 1*10.
nD(C) shorted results in response 111* 1110.

Relative Frequency of Failures

In the analysis to follow, it is necessary to make assumptions. In all cases, the basis of the assumption is explained. Other investigators might make different assumptions, however, the following analysis was performed several times with differing assumptions and with only a small change in the overall result. For that reason it is believed that the final results are very indicative of the failure modes which can be expected.

We begin by looking again at the major failure mechanisms cited earlier. From experience in the device fabrication laboratory at Sandia as well as experience at Bell Telephone Laboratories,[7] estimates were made and are indicated below.

Photolithography errors, 7.5/cm^2 of chip area. The three manifestations of these errors are assumed equally likely and are therefore apportioned a weight of 2.5 each.

Oxide Pinholes	2.5
Missing Features	2.5
Source Drain Shorts	2.5

Metal errors, 4/cm^2 of chip area. Open and shorts are assumed equally likely with the weighting of 2 each.

Metal Shorts	2.0
Metal Opens	2.0

Diffusion errors, 2/cm^2 are given a weight of 2.

The final analysis of probabilities will normalize the failure probabilities to add to 1.0. For the moment, the relative weights will be used. When analyzing each failure mechanism, the ways the mechanism can affect the chip will be considered and a subweight (normalized to 1.0) assigned. Thus, if a given mechanism can affect, say, six transistors with approximately an equally probability, the subweight of 1/6 will be assigned.

In some cases, a third factor is used to separate faults which occur within a cell and those occurring on the remainder of the chip, e.g., between cells. The general weights assigned to failures outside the cell are 0.4 and those within the cell are 0.6. This conforms to the general experience obtained using the standard CMOS cell technology to design custom chips based on apportionment of chip area.

A detailed analysis follows. The faults are first numbered.

Oxide pinholes (overall weight of 2.5) are assumed to cause an input to stick at 1 or 0 with equal probability. Since any input s-a-0 results in the cell s-a-1, this condition is subweighted 3/6 and the input s-a-1 faults are weighted 1/6 each.

(1)	Gate s-a-1, response	1111 1111	subweight 3/6	3/6 x 2.5 = 1.25
(2)	Input A s-a-1	1111 1100	1/6	1/6 x 2.5 = .41667
(3)	Input B s-a-1	1111 1010	1/6	1/6 x 2.5 .41667
(4)	Input C s-a-1	1110 1110	1/6	1/6 x 2.5 .41667

Missing features (overall weight 2.5) outside the cell area are assumed to cause an open interconnect. This will result in a cell input stuck at 1 or 0 with equal likelihood. Within the cell a missing feature is assumed to cause a transistor to act as a short or open with equal probability.

Outside the cell is weighted 0.4 which multiplied by 2.5 gives 1.0.

(5)	Gate s-a-1	1111 1111	3/6	3/6 x 1. = .5
(6)	Input A s-a-1	1111 1100	1/6	1/6 x 1. = .16667
(7)	Input B s-a-1	1111 1010	1/6	1/6 x 1. = .16667
(8)	Input C s-a-1	1110 1110	1/6	1/6 x 1. = .16667

Inside the cell is weighted 0.6 which multiplied by 2.5 gives 1.5.

(9)	pD(A,B,C) shorted	1111 111*	3/12	3/12 x 1.5 = .375
(10)	nD(A,B,C) open	1111 1111	3/12	3/12 x 1.5 = .375
(11)	pD(A) open	1111 11*(.857)0	1/12	1/12 x 1.5 = .125
(12)	pD(B) open	1111 1*(.857)10	1/12	1/12 x 1.5 = .125
(13)	pD(C) open	111*(.857) 1110	1/12	1/12 x 1.5 = .125
(14)	nD(A) short	1111 11*0	1/12	1/12 x 1.5 = .125
(15)	nD(B) short	1111 1*10	1/12	1/12 x 1.5 = .125
(16)	nD(C) short	111* 1110	1/12	1/12 x 1.5 = .125

Source drain shorts caused by photolithography errors - overall weight 2.5.

(17)	pD(A,B,C) short	1111 111*	3/6	3/6 x 2.5 = 1.25
(18)	nD(A) shorted	1111 11*0	1/6	1/6 x 2.5 = .416667
(19)	nD(B) shorted	1111 1*10	1/6	1/6 x 2.5 = .416667
(20)	nD(C) shorted	111* 1110	1/6	1/6 x 2.5 = .416667

Metal shorts have an overall weight of 2.0.

In this area there are an infinitude of possibilities of possible shorted connections. The investigation will be restricted to two conditions: the shorting of two input leads together and the short of an input lead to an output. Considering the length of metal lines running outside the cell, the former will be weighted 0.8 and the latter 0.2.

To determine these effects, the SPICE[8] code was used using the model of Poon at BTL.[9] The NAND gate under investigation was modeled and placed into a test circuit for analysis.[10] Three similar gates were connected to the three inputs and a similar gate placed on the output. It was found that the effect of shorting two input lines was to reduce the speed of the cell, but not its response. Accordingly, the response of 1111 111* was concluded as proper. For the cell input shorted to the output the response as shown was indicated by the analysis.

(21)	Any two inputs shorted	1111 111*	0.8	0.8 x 2 = 1.6
(22)	In A shorted to the output	11*1 *101	0.2/3	0.2/3 x 2 = .13333
(23)	In B shorted to the output	1*11 *011	0.2/3	0.2/3 x 2 = .13333
(24)	In C shorted to the output	1**0 1111	0.2/3	0.2/3 x 2 = .13333

Metal opens, overall weight of 2.0. Outside the cell an open metal line is assumed to cause an input to stick at 1 or 0. Within the cell the assumption is that an individual transistor will act as though it were shorted or opened. This analysis is the same as for the photolithography metal open with different weights.

Outside the cell defects are weighted 0.4 which multiplied by 2.0 gives 0.8.

(25)	Gate s-a-1	1111 1111	3/6	3/6 x .8 = .4
(26)	Input A s-a-1	1111 1100	1/6	1/6 x .8 = .13333
(27)	Input B s-a-1	1111 1010	1/6	1/6 x .8 = .13333
(28)	Input C s-a-1	1110 1110	1/6	1/6 x .8 = .13333

In the cell defects are weighted 0.6 which multiplied by 2.0 gives 1.2.

(29)	pD(A,B,C) short	1111 111*	3/12	3/12 x 1.2 = 0.3
(30)	nD(A,B,C) open	1111 1111	3/12	3/12 x 1.2 = 0.3
(31)	pD(A) open	1111 11*(.857)0	1/12	1/12 x 1.2 = .1
(32)	pD(B) open	1111 1*(.857)10	1/12	1/12 x 1.2 = .1
(33)	pD(C) open	111*(.857) 1110	1/12	1/12 x 1.2 = .1
(34)	nD(A) short	1111 11*0	1/12	1/12 x 1.2 = .1
(35)	nD(B) short	1111 1*10	1/12	1/12 x 1.2 = .1
(36)	nD(C) short	111* 1110	1/12	1/12 x 1.2 = .1

Diffusion Errors - Overall weight of 2.0.

Four distinct types of failures will be considered relative to diffusion errors. They are assumed to be equally likely.

P^+ Shorts.

A short of any of the top p-channel devices to the supply will cause the cell to stick at one.

(37) Gate s-a-1 1111 1111 1/4 1/4 x 2 = .5

P^+ opens will have the effect of eliminating one of the p devices.

(38) pD(A) open 1111 11*0 1/4 x 1/3 1/12 x 2 = .16667
(39) pD(B) open 1111 1*10 1/4 x 1/3 1/12 x 2 = .16667
(40) pD(C) open 111* 1110 1/4 x 1/3 1/12 x 2 = .16667

N^+ shorts cause the effect of shorting one of the n type transistors.

(41) nD(A) short 1111 11*0 1/4 x 1/3 1/12 x 2 = .16667
(42) nD(B) short 1111 1*10 1/4 x 1/3 1/12 x 2 = .16667
(43) nD(C) short 111* 1110 1/4 x 1/3 1/12 x 2 = .16667

N^+ open causes an open n-channel device.

(44) nD(A,B, pr C) open 1111 111* 1/4 x 2 = .5

Fault Likelihood Analysis

In the previous section a total of 44 faults were considered and weighted. These data were fed to a code called WEIGHT[10] which expands the truth tables and tabulates the probability weights. E.g., 1111111* expands to 11111111 and 11111110 and the undefined mapping into a 1 or 0. The weights are normalized to add to 1.0 and a sort performed to rank them in descending order. The results of these analyses are shown below in Table I.

TABLE I.
WEIGHT PROGRAM RESULTS

Response Table	Name if Applicable	Normalized Weight
1111 1111	Gate s-a-1	.39196
1111 1110	Normal response	.30283
1111 1100	Input A s-a-1	.09102
1111 1010	Input B s-a-1	.09102
1110 1110	Input C s-a-1	.09102
1000 1111		.00268
1010 1111		.00268
1011 0011		.00268
1011 1011		.00268
1100 1111		.00268
1110 1111		.00268
1111 0011		.00268
1111 1011		.00268
1101 0101		.00268
1101 1101		.00268
1111 0101		.00268
1111 1101		.00268

It can be seen that the stuck-at faults predominate. The gate s-a-0 does not appear since all three n-channel devices would have to short at once. The normal response appears about 1/3 of the time even if faults are present. Another interesting result is that the gate s-a-1 is roughly four times as likely as any of the inputs stuck at 1. Finally, it is noted that other response tables besides the stuck-at fault is only 3.2% likely to happen. These are due to a gate input shorting to the output.

Improved States Applied Analysis

The concept of "states applied" analysis has been previously reported.[11,12] The basic idea behind a states applied analysis is that certain faults can only be detected if certain applied states are present. For example, in the case of the three input NAND gate, the gate s-a-1 can be detected only if the all 1 state is applied to the input. Any other applied state will see no difference in response between the faulted as opposed to normal gate. Below in Table II, the conventional applied state table is given for a 3-input NAND gate.

TABLE II.
STUCK-AT RESPONSES FOR 3-INPUT NAND

State no.	0 1 2 3	4 5 6 7	
Input A	0 1 0 1	0 1 0 1	
Input B	0 0 1 1	0 0 1 1	
Input C	0 0 0 0	1 1 1 1	
Normal response	1 1 1 1	1 1 1 0	
Gate s-a-1	1 1 1 1	1 1 1 1	detectable by state 7
Gate s-a-0	0 0 0 0	0 0 0 0	detectable by states 0-6
Input A s-a-1	1 1 1 1	1 1 0 0	detectable by state 6
Input B s-a-1	1 1 1 1	1 0 1 0	detectable by state 5
Input C s-a-1	1 1 1 0	1 1 1 0	detectable by state 3

If in the course of running the test sequence, it was noted that states 0,2,3,5 and 6 had been applied, the analysis would conclude that the gate s-a-0 and each of the inputs s-a-1 were potentially detectable or 80% of the faults were potentially detectable.

This can be modified by including the fault likelihood given for each of the applied states using the results of Table I.

State 0	0.00 (no failure mode produces a response table which differs from normal due to applied state 0)
1	.01316
2	.01316
3	.13398
4	.01316
5	.13398
6	.13398
7	.55814

The conventional states applied analysis gave a score of 80%. Under the modified system the score would be only 41% since state 7 is not applied and the state with the highest score is number 7.

Conclusions

This analysis has shown that the "stuck-at" model is basically valid, but the fault likelihoods for the technology involved should be considered. In the example used for analysis, the likelihood of a gate s-a-1 is four times as likely as any gate input s-a-1. Also, the interesting result is noted that in some failure modes, the gate will still perform its intended function.

It is also noted that certain states applied as a result of running the test sequence are much more valuable from a diagnosis point of view than others. In our example state 0 has no value and state 7 has a value of over 50%.

Acknowledgments

The author wishes to express his thanks to Mr. D. S. Hill for his help in preliminary analysis of the fault mechanisms and his help in setting up the SPICE code.

References

1. Gwyn, C. W., "Computer Aided Design of MOS Integrated Circuits," CUBE Symposium Proceedings, 1974 (Available through National Technical Information Service).

2. RCA COS/MOS Integrated Circuits Manual, RCA Solid State Division, Somerville, New Jersey 08876 (Available from RCA - price $2.50).

3. Fike, J. L., "Predicting Fault Detectability in Combinational Circuits - A New Design Tool?" 12th Design Automation Conference Proceedings.

4. Armstrong, D. B., "On Finding a Nearly Minimal Set of Fault Detection Tests for Combinational Logic Nets," IEEE Trans. on Electronic Computers, Vol. ED 15, Feb. 1966.

5. Verma, J. P., Selove, D. M., and Tessier, J. N., "Automatic Test-Generation and Test-Verification of Digital Systems," 11th Design Automation Workshop Proceedings, 1974.

6. Friedman, A., and Menon, P., (F. Kup, Editor), Fault Detection in Digital Circuits, Prentice Hall, 1971.

7. Murphey, B. T., "Cost-Size Optima of Monolithic Integrated Circuits," Proc. IEEE, Vol. 52, (1964), pp. 1537-1545.

8. Nagel, L. W., and Pederson, D. O., SPICE (Simulation Program with Integrated Circuit Emphasis, Electronics Research Laboratory, College of Engineering, University of California, Berkeley, California.

9. Private Communication from H. C. Poon, Bell Telephone Laboratories.

10. Case, G. R., Analysis of Actual Fault Mechanisms in CMOS Logic Gates, Sandia Laboratories Report, SAND75-0621.

11. Case, G. R., "SALOGS - A 6600 Program to Perform Digital Logic Simulation and Fault Diagnosis," CUBE Symposium Proceedings, 1974 (Available through National Technical Information Service).

12. Case, G. R., "A Statistical Method for Test Sequence Evaluation," 12th Design Automation Conference Proceedings, 1975.