THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 57, No. 5, May–June 1978
Printed in U.S.A.

Fault Coverage in Digital Integrated Circuits

By R. L. WADSACK

(Manuscript received October 10, 1977)

A theoretical expression is derived in this paper that evaluates the effectiveness of a set of logic tests for digital integrated circuits. The validity of the proposed figure of merit is examined with experimental data from CMOS integrated circuits. In addition, the importance of simulating the nonclassical stuck-open/stuck-on CMOS logic faults is also studied.

I. INTRODUCTION

The ever-growing complexity of digital integrated circuits places increasing emphasis upon the use of computerized design aids. Because no integrated circuit design is complete without an accompanying set of tests, one essential tool is the logic simulator.[1] The two principal reasons for logic simulation are (*i*) to verify the logic design and (*ii*) to develop the set of tests. A third purpose, related to the second, is that of diagnosis, i.e., identification of logic faults causing specific yield problems.

This paper will address itself to a study of the relation between fault coverage and measured yield and will consider specifically CMOS integrated circuits. The latter choice was made for two reasons. First, CMOS ICs are an attractive choice for many system designs. Second, CMOS ICs can possess nonclassical logic faults peculiar to MOS circuit elements: stuck-opens and stuck-ons.[2]

To verify the logical behavior of the IC, the test engineer usually begins with binary "vectors," or test patterns, that test the basic input/output logic functions of the circuit. For example, if the IC is a multiplexer, then multiplexing different data patterns is a natural starting point. Designing a set of vectors for high fault coverage generally represents a larger challenge than that of design verification. The difficulty arises because the logical structure of the IC must be tested and not just its generic properties, such as multiplexing. The major disadvantage of "behavioral" tests is that they are usually too lengthy.[3,4] Consequently, the simplest approach is to begin with a sequence of representative behavioral tests and then add to them the necessary "structural" tests to bring the fault coverage to the required level. In any event, the process of developing the digital tests should start as soon as the systems logic design is formulated and before the design reaches the mask layout phase.

II. FAULT COVERAGE AND MEASURED YIELD

Perhaps one of the most important questions for test vector development is: How much fault coverage is enough? The answer must obviously be related to the intrinsic yield of the IC under test. For example, if the yield is 100 percent, then any low-coverage vector set can be used, including none at all. On the other hand, if the yield is low, then there will be many defective ICs that can potentially masquerade as "good" devices if the fault coverage is poor. In the latter case, the lower the fault coverage the more probable it will be that a chip that tests "good" contains a fault.

In any event, the measured functional yield,* ym, is the sum of two components: the actual functional yield, y, and the yield of bad ICs tested "good," ybg. Thus, $ym = y + ybg$. This leads to a second question: How is ybg related to the fault coverage f? Consider the following definitions for the functional yield problem:

y = actual functional yield (good chips).
ym = measured functional yield.
$1 - y$ = yield of bad chips.
yi = yield of chips with i faults ($i = 1,2,3 \cdots$).
ybg = yield of bad chips that test good.
$ybg(i)$ = yield of bad chips, with i faults, that test good.
f = fault coverage ($0 \le f \le 1$).
yr = field reject rate due to functional defects.

First, assume that $ybg(i) = (1 - f)^i \cdot yi$. For $i = 1$, this implies $ybg\,(1) = (1 - f) \cdot y1$, which is quite reasonable because it represents the basic assumption behind most logic simulation. That is, a logic simulator considers only the population of all chips in which there is only one logic fault per chip. Second, if the fault coverage is f, then $(1 - f)$ is the fraction of all faults (chips) that will be undetected by the test sequence. For $i = n > 1$, the assumption amounts to stating that the probability of n faults being undetected is $(1 - f)^n$. This is true only if multiple faults are independent of one another.

* The yield of chips that are free of logic faults irrespective of their analog voltage/current behavior.

The second assumption is that $yi = y \cdot (1 - y)^i$ ($i = 1,2,3,\cdots$). This is the geometric distribution function and has been found to correctly describe the distribution of defective cells in static RAM chips.[5] In particular, the average yield y and the average number of fault-producing defects per chip, x_0, were measured and found to be related by the equation $y = 1/(1 + x_0)$. In the case of the RAM chips, $x_0 = 2.7$. For defect densities higher than, say, 5 per chip, a different distribution may be necessary.

Under the above assumptions,

$$ybg = \sum_{i=1}^{\infty} (1 - f)^i \cdot y \cdot (1 - y)^i$$

and

$$ybg = y\left[\frac{(1 - f)(1 - y)}{1 - (1 - f)(1 - y)}\right].$$

Therefore,

$$ym = \frac{y}{1 - (1 - f)(1 - y)}.$$

The field (or incoming inspection) reject rate yr is determined by the fraction of bad ICs that passed the functional test vector sequence, but that would have failed had the fault coverage been higher.

Therefore,

$$yr = ybg/ym,$$

which gives

$$yr = (1 - f)(1 - y).$$

(Note that in this context *undetectable* faults are not included in the statistical base for fault coverage. The most likely negative consequences of undetectable faults are long-term reliability problems or intermittents, not failures at incoming inspection.)

Inversely, for a given field reject rate (or "quality level") the fault coverage would be

$$f = 1 - \frac{yr}{1 - y}.$$

As an example, for an IC with a yield of 20 percent ($y = 0.2$), the fault coverage would have to be equal to or greater than 98.8 percent for a reject rate of 1 percent ($yr = 0.01$) or lower due to undetected logic faults.

III. FAULT COVERAGE AND LOGIC SIMULATORS

The percent fault coverage quoted for a set of test vectors is an important measure of their test effectiveness. Other things being equal, a vector sequence with 90 percent coverage is twice as likely to identify faulty ICs as one with only 45 percent coverage. Equally important, however, is the question: Ninety percent of what? Unfortunately, the answer is usually 90 percent of what faults the simulator simulates. Therefore, in comparing different simulations and quoted fault coverage, it is essential to know the kinds of faults that were modeled. Many simulators model only classical faults (stuck-at-1 and stuck-at-0). Others may treat only gate *output* stuck-at faults. It is common in the case of printed circuit board simulations to consider only the pin faults of each IC on the board. In the experimental results of the next section, all classical faults were modeled in addition to the relevant CMOS stuck-open and stuck-on faults.[2]

Second, to lower costs, simulations generally use only a "collapsed" set of faults. That is, several faults on different gates may cause the same faulted circuit behavior as viewed from the primary circuit outputs. Consequently, they are included in a single fault equivalence class with only one fault in that class being simulated. As an example, a chain of three inverters would have six physically distinct classical faults. However, after fault collapsing, only two faults would be simulated. The obvious drawback to fault collapsing is that it distorts the relation between the predicted and the observed number of failures.

An additional factor can alter the ratio of predicted to observed failures: the probability of and relation between physical faults and simulator faults. First, not all physical faults are equally probable. Second, an individual physical fault does not necessarily produce a single logical fault, i.e., one fault may map into two or more simulator faults or vice versa. In addition, the distribution function for physical defects produces many more chips with multiple faults than chips with only one fault. Obviously, this could pose a problem in the interpretation of failure data because fault simulators simulate only singly faulted circuits, not those with multiple faults. The effect of gross physical faults and the preponderance of chips with multiple faults is to cause a higher number of failures during the initial part of the test sequence compared to that predicted by the simulator.

The simulator can also contribute its own distortions. For example, it is clear that undetected faults that caused the simulation to oscillate may well cause an actual integrated circuit to fail during testing. In the same fashion, the simulator can be overly pessimistic with respect to other faults that produce or leave unknown states in the circuit (e.g., set/reset inputs to flip-flops). Conversely, the simulator may treat a particular fault as having been detected on a specific vector, but the fault causes the IC to fail on a following vector. This can occur because of differences between the discrete delays of the simulator and the actual delays present in the IC.

Table I — Circuit characteristics

Circuit	Inputs	Outputs	Gates
D flip-flop	4	2	15
Multivibrator	8*	6†	47
MUX	14	7	238

* Includes one as I/O and another for I/O control.
† Includes one as I/O.

The above are some of the more evident reasons why fault simulator results are only approximations to those actually measured on integrated circuits. This is true even if the simulator modeled all reasonable types of logic faults.

IV. EXPERIMENTAL RESULTS

Three CMOS integrated circuits were selected for studying the relation between fault coverage and yield. Two of these circuits were studied in some detail. The two circuits are (*i*) a dual D flip-flop functionally equivalent to the RCA CD4013A and (*ii*) a monostable/astable retriggerable multivibrator similar to the RCA CD4047A.

Table I gives the circuit characteristics of the circuits. The column labeled "gates" gives the gate count for each circuit with the convention that a node to which two or more transmission gates connect is counted as one logic gate. Table II summarizes the fault characteristics of each circuit. The circuits were modeled to include the nonclassical stuck-open/stuck-on CMOS faults.[2]

Although CMOS faults double the number of total faults, it is not obvious whether they should be counted on a 1:1 basis with classical faults. If the probability of occurrence of stuck-open/stuck-on faults is markedly different than that of SA0, SA1 faults, then a weighting factor different than unity should be used to determine the total number of "effective" faults. As a second consideration, classical faults are collapsible into equivalence classes, but the CMOS nonclassical faults are individualized to single gates.

Table II — Fault characteristics

Circuit	(1) Physical Gates	(2) Faults* Classical	(3) Faults* CMOS	(4) Faults* Total	(5) Total Faults per Gate
D-FF	15	38	38	76	5.1
MULTI	47	133	134	267	5.7
MUX	238	902	536	1438	6.3

* After fault collapsing.

Table III — Fault coverage results

Circuit	(1) No. of Gates	(2) Faults per Gate	(3) No. of Vectors	(4) Vectors per Gate	(5) Fault Coverage* Classical	(6) CMOS	(7) Total
D-FF	15	5.1	15	1.0	100%	100%	100%
MULTI	47	5.7	119	2.5	95	84	89
MUX	238	6.3	5549	23	95	90?	93?

* All faults, including undetectables.

4.1 The D flip-flop circuit

Not surprisingly, the D flip-flop circuit is 100 percent testable for both classical and CMOS logic faults (see Table III). Figure 1 shows the cumulative total fault coverage versus the fraction of the test vector sequence applied to the IC. In this case, 15 vectors were used to reach 100 percent coverage. Strictly speaking, the points in Fig. 1 should have been connected by step functions that rise to meet each datum point. For the sake of clarity, however, straight line segments running directly from one point to the other were used.

Figure 2 shows the relation between the total fault coverge f(total) and the classical fault coverage f(class). The total fault coverage falls below that for classical faults because of a characteristic lag in CMOS fault

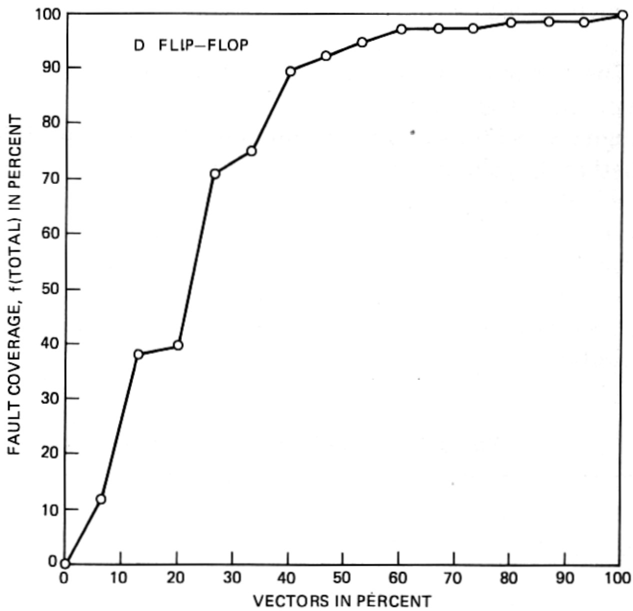

Fig. 1—The D flip-flop: total fault coverage vs. normalized vector number (15 vectors total).

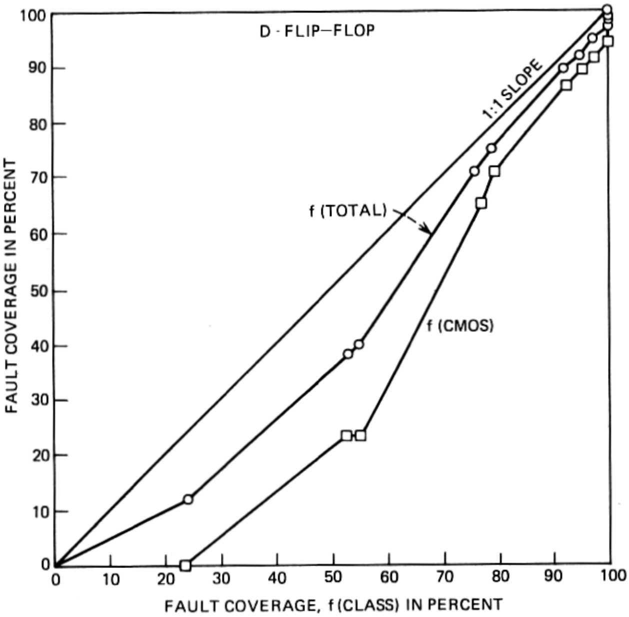

Fig. 2—The D flip-flop: total fault coverage and CMOS fault coverage as functions of the classical fault coverage for the vector sequence of Fig. 1.

coverage. The CMOS lag is more clearly shown by the second curve, marked *f*(CMOS), in Fig. 2. The lag is caused by the "history-dependence" of CMOS stuck-open faults that require at least two different vectors for detection.

Figure 3 shows a comparison between the simulation data and actual measurements for 11,150 ICs from 28 wafers. It is important to note that the curves of Fig. 3 are *reverse* cumulative distribution functions.[6] The independent variable is the vector number. Fifteen vectors were used to test this circuit. Vector 1 was the first in the sequence and is shown at the origin of the graph. The dependent variable is the running sum of the number of detected faults (or chip failures) beginning with vector 15 and proceeding to vector 1. Hence, the curves indicate the likelihood that a defective chip will fail at a specific vector.

In the simulation curve, the cumulative number of detected faults is shown. For the wafer data the cumulative number of functional, or logic test, failures is plotted. Reverse cdf's were chosen because they reveal the structure of the tail regions where the fault coverage is not changing as rapidly as it is near the beginning of the vector sequence. Of course, the tail of each curve illuminates most clearly the effects of ICs with single faults.

The simulation data are the same as those used for Fig. 1 in which all

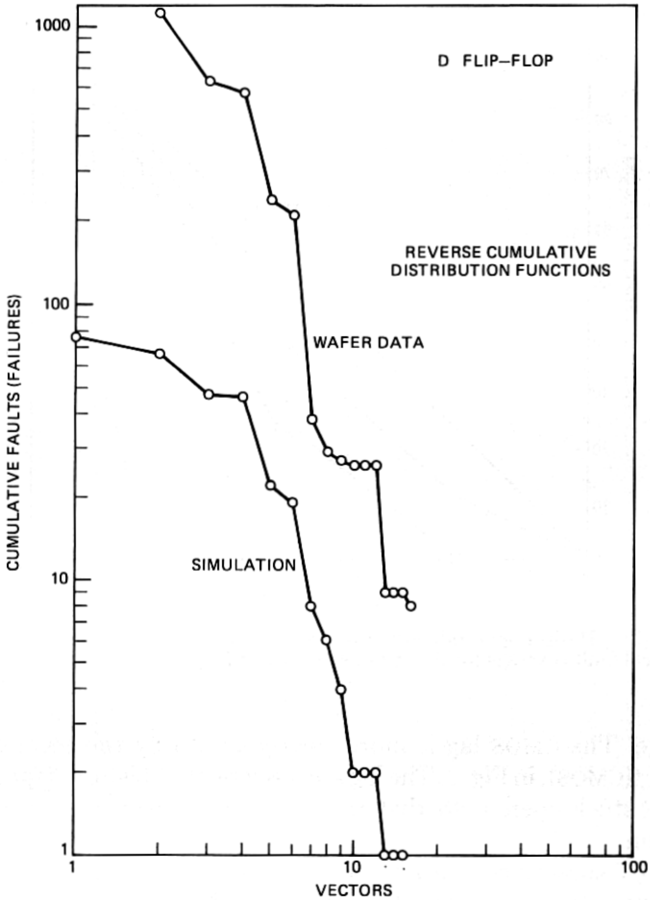

Fig. 3—The D flip-flop: reverse cumulative distribution functions for functional yield loss plotted as functions of the vector number. The simulation data are that of Fig. 1. The measured data points were obtained from 11,150 chips from 28 wafers.

faults, both classical and CMOS, are included. There is reasonable agreement between the "predicted" and the measured rcdf's. Only by coincidence would the two curves lie one upon the other because each is plotted in absolute numbers. Ideally, of course, they would be separated by a constant vertical displacement. The agreement is one of general shape. A specific quantitative comparison is treated later in this paper.

Naturally, near the beginning of the sequence there are more actual IC failures than indicated by the simulation. Recalling the factors discussed in Section III above, initial failures are probably caused by gross shorts, opens, and multiple faults. (All chips, however, were prescreened for contact failures.) The slight dip in the wafer data for vectors 7,8,9 is caused by a 2:1 overprediction of fault coverage. Vectors 7,8,9 detect, in a worst-case sense, set and reset faults which in a real IC are more likely to fail earlier in the vector sequence.

Only one fault is detected by vector 12, the data input transmission gate stuck-on. The failure rate on that vector was 17/11150 or 0.15 percent. The only other vector that detects a single CMOS fault is vector 15. The fault is a stuck-open in the master flip-flop feedback transmission gate. The failure rate was 1/11150, or 0.009 percent, quite low compared to the stuck-on fault. Strangely, there are 8 ICs (0.07 percent) that failed at an added vector 16 where the fault coverage is already at 100 percent. Vector 16 forces the set and reset inputs active at the same time. Although the behavior of the fault-free circuit is deterministic, no structural faults (classical, stuck-open, or stuck-on) remain in the circuit to be detected at that vector. The failures may have been caused by analog effects.

The predicted field reject rate yr, as a function of fault coverage f, can be computed from the data of Figs. 1 and 3. The procedure begins first with Fig. 3 where the measured yield is obtained as a function of the vector at which the sequence was truncated. Next, the fault coverage for the truncated vector set is established by reference to Fig. 1. Of course, if the entire untruncated vector set is applied to the IC, the cumulative fault coverage at the last vector is 100 percent and the measured yield ym should equal the true yield y.

The results of the above calculations are shown in Fig. 4. The theoretical relation, $yr = (1 - f)(1 - y)$, is the solid line. The two sets of data points represent the computed yr based, first, on all faults and, second, on only classical faults. The latter lie closer to the theoretical prediction.

Several conclusions can be drawn from the results of Fig. 4. The first is that the equation is a good estimator of the reject rate, but is somewhat pessimistic at high fault coverage. Second, the difference between the predicted and measured yr at high values of f can be explained by a proportionally larger actual fault coverage used as the abscissa (in each case). Finally, even though CMOS stuck-open/stuck-on faults were identified in some of the circuits, their relative probability seems to be much less than that of the more general classical stuck-at faults. In other words, a 1-to-1 weighting factor does not appear to be warranted.

4.2 The multivibrator circuit

The multivibrator was not 100 percent testable. The cumulative fault coverage for all faults is shown in Fig. 5. The occasional abrupt jumps in fault coverage are caused by sequential portions of the circuit. That

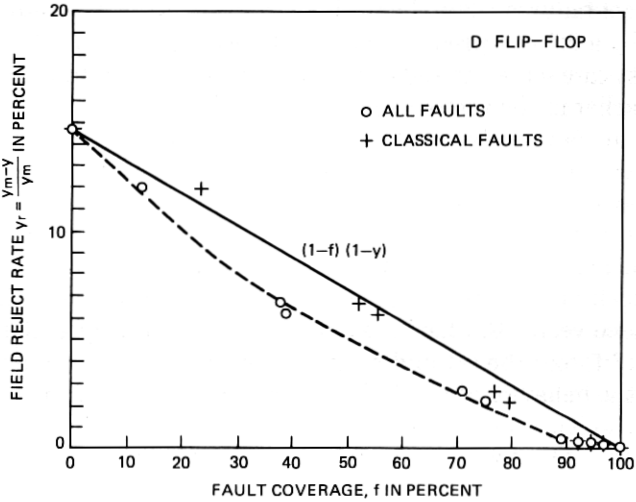

Fig. 4—The D flip-flop: field reject rate vs. total fault coverage as determined from the data of Figs. 1 and 3. The theoretical expression is indicated by the solid line.

is, several vectors are needed before a "fault effect" is generated at a gate and several more are necessary to propagate the fault to an output. When faults from that part of the circuit finally reach an output, there is a sharp increase in coverage.

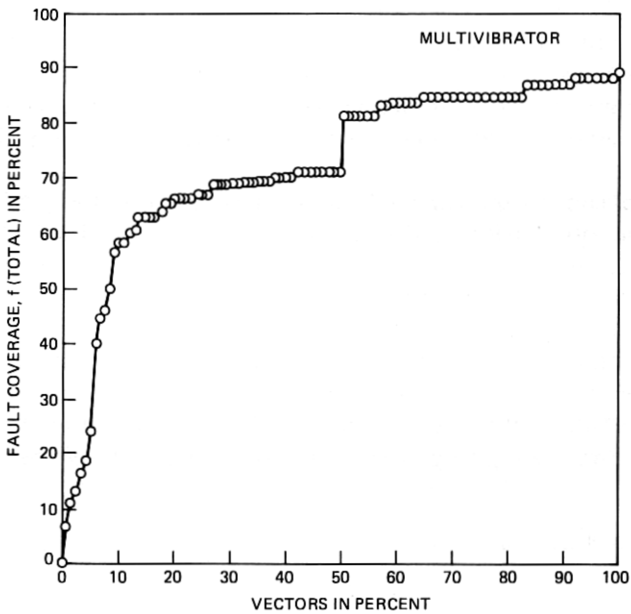

Fig. 5—The multivibrator: total fault coverage vs. normalized vector number (119 vectors total).

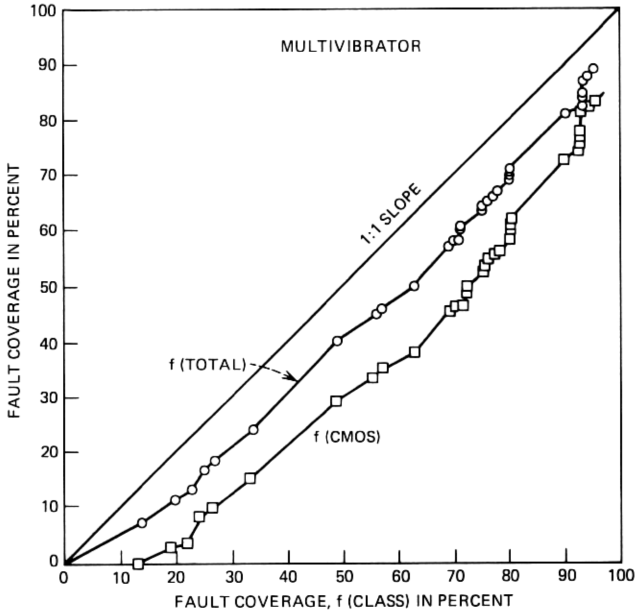

Fig. 6—The multivibrator: total fault coverage and CMOS fault coverage as functions of the classical fault coverage for the vector sequence of Fig. 5.

The final vector set provided a coverage of 89.1 percent, or 238 faults out of 267. The remaining 29 faults are all undetectable. There were two primary causes for the undetectable faults. The first was the presence of "asynchronous" circuit behavior (in the retrigger control section). The second was the use of two D flip-flops which had data inputs tied permanently to a fixed logic value (lack of controllability). Two undetectables occurred in a NOR latch: all CMOS latches formed by cross-coupled NOR or NAND gates have two undetectable faults.

Figure 6 shows the relation between the total fault coverage f(total) and the classical fault coverage f(class). Again the lag in CMOS fault coverage is evident. The total fault coverage reaches 89.1 percent. Classical coverage is 94.7 percent; CMOS coverage is 83.6 percent. Of the 29 undetectable faults, 22 were CMOS and 7 were classical.

Figure 7 compares simulation data with measurements taken from a single wafer. The wafer contained 418 chips which passed initial contact tests. Of those, 275 passed the logic tests for a gross functional yield of 66 percent. Again, the reverse cdf's are used to show the behavior in the tail regions near the end of the vector sequence. The overall agreement between the two curves is reasonable.

As in the above D flip-flop example, the field reject rate yr can be calculated for the multivibrator circuit from the corresponding data

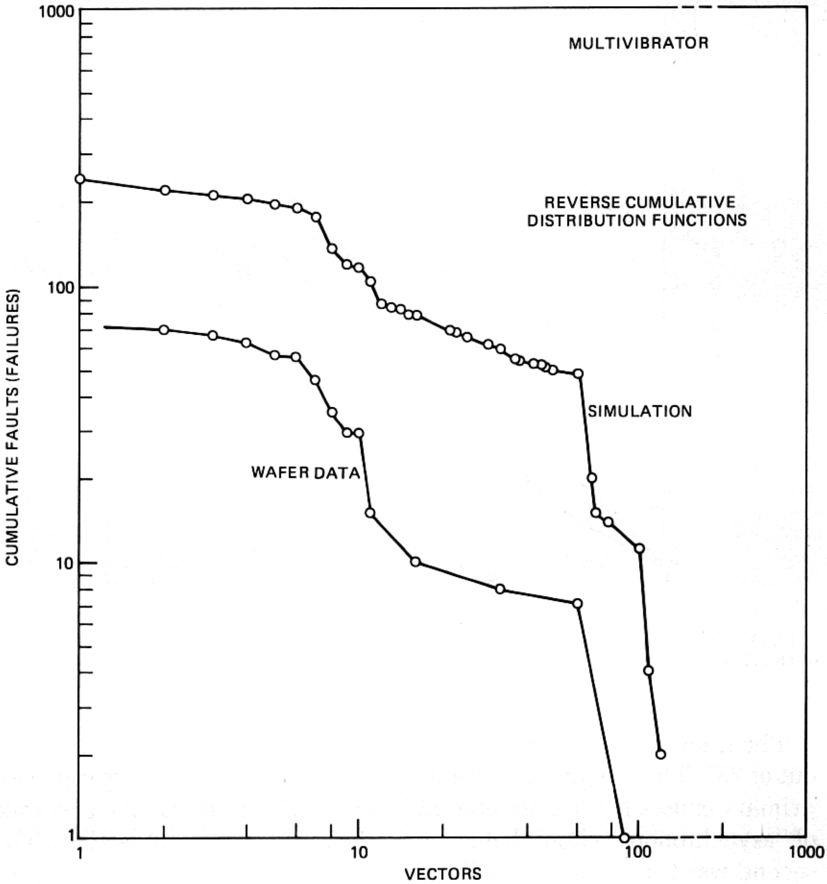

Fig. 7—The multivibrator: reverse cumulative distribution functions of the vector number. The simulation data are that of Fig. 5. The measured data points were obtained from 418 chips from one wafer.

(Figs. 5 and 7). The resultant points are shown in Fig. 8. The solid line is the predicted relation $yr = (1 - f')(1 - y)$, where f' is the fault coverage for all detectable faults.

The data of Fig. 8 suggest the same observations as for the D flip-flop: The actual fault coverage appears to be higher toward the end of the vector sequence than that predicted by the simulator. Also, the theoretical yr is best matched by the "classical faults only" data. Again, this indirectly implies that the relative frequency of CMOS faults is significantly less than that of the classical faults. In addition, for both the D flip-flop and the multivibrator the curves are similar above the 75 percent fault coverage point. In particular, each indicates that a coverage of 85 to 90 percent is needed to achieve a reject rate of 1 percent or less.

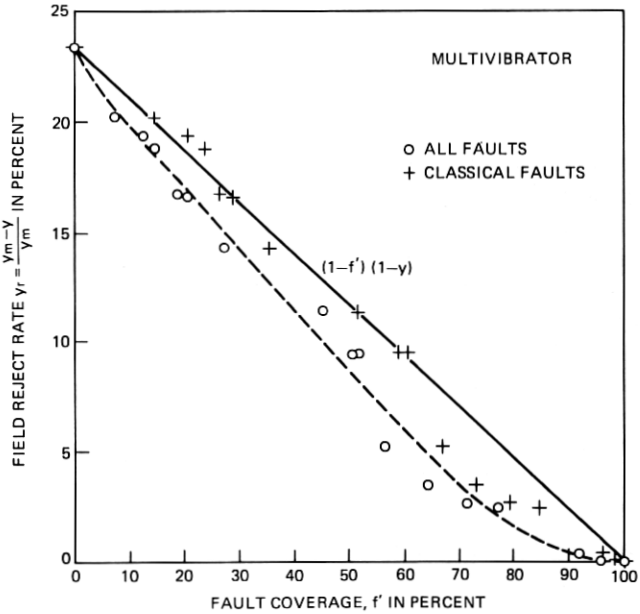

Fig. 8—The multivibrator: field reject rate vs. total (detectable) fault coverage as determined from the data of Figs. 5 and 7. The theoretical expression is indicated by the solid line.

On the other hand, the true yields for each are approximately equal (85 and 77 percent, respectively).

V. SUMMARY

The fault coverage results of the previous section have been summarized in Table III. To compare the difficulty of generating test vectors for one circuit versus another, a measure of circuit complexity is needed.[7] Gate count [column (1)] is a poor measure because it reflects only silicon area and not the interconnections that create the actual circuit. One potential measure of circuit complexity that indicates at least the magnitude of test vector generation is the ratio of the number of vectors to the number of gates [column (4)]. In that sense, the multivibrator is 2.5 times more complex than the D flip-flop, and the multiplexer (MUX) is 23 times more complex. Of course, a measure could be devised to incorporate the number of circuit inputs and outputs. However, for the three circuits the most dominant effect is that of the number of test vectors. The reader can readily interpret from Table III the magnitude of the test generation and fault coverage problems for large-scale silicon-integrated circuits with thousands of gates. In addition, for modern IC test equipment the number of circuit inputs and outputs generally does not affect the functional test time (as long as the number is less than the test set maximum).

Nevertheless, there are two major drawbacks to using "vectors/gate" as a measure. First, it is retrospective: Only after effort has been expended to develop the test vectors does the "complexity" become known. The second reason is that it depends somewhat upon the method or skill used to generate the test vectors themselves. In particular, the test vector sequences used for the example circuits are certainly not unique. Nor is it likely that any of them is optimal in the sense of being the least number for the same level of coverage. The basic problem is that there probably isn't any simple one-dimensional measure of circuit complexity that is useful for a broad spectrum of circuit types.

The relative frequency of CMOS stuck-open/stuck-on faults appears to be significantly less than that of the classical stuck-at-0/stuck-at-1 faults. On the other hand, CMOS nonclassical faults do occur. Perhaps the best approach to resolving this quandary would be a study of many different CMOS ICs. The investigation would use vector sets of high diagnostic capability to determine which kinds of logic faults are important.

Finally, the data presented in this paper support the reject rate (quality level) concept as an answer to the question, "How much fault coverage is enough?" However, the total economic picture obviously must take into account the cost of developing the vectors and the cost of using (applying) them. Only when all three of the above factors are considered can the cost of integrated circuit testing be properly judged.

REFERENCES

1. "LAMP: Logic Analyzer for Maintenance Planning," B.S.T.J., *53,* No. 8 (October 1974), pp. 1431–1555.
2. R. L. Wadsack, "Fault Modeling and Logic Simulation of CMOS and MOS Integrated Circuits," B.S.T.J., this issue, pp. 1449–1474.
3. A. K. Susskind, "Diagnostics for Logic Networks," IEEE Spectrum, *10,* No. 10 (October 1973), pp. 40–47.
4. W. G. J. Kreuwels, "Structural Testing of Digital Circuits," Philips Tech. Rev., *35,* No. 10 (1975), pp. 261–270.
5. R. L. Wadsack, unpublished work.
6. M. B. Wilk and R. Gnanadesikan, "Probability Plotting Methods for the Analysis of Data," Biometrica, *55,* No. 1 (1968), pp. 1–17.
7. J. Stephenson and J. Grason, "A Testability Measure for Register Transfer Level Digital Circuits," FTCS-6, Proceedings (1976), pp. 101–107.