# Mining IC Test Data to Optimize VLSI Testing

**Tony Fountain**
San Diego Supercomputer Center
University of California, San Diego
La Jolla, CA 92037

**Thomas Dietterich**
Computer Science Dept.
Oregon State University
Corvallis, OR 97331

**Bill Sudyka**
Hewlett Packard Co.
Corvallis, OR 97330

**Abstract**

We describe an application of data mining and decision analysis to the problem of die-level functional test in integrated circuit manufacturing. Integrated circuits are fabricated on large wafers that can hold hundreds of individual chips ("die"). In current practice, large and expensive machines test each of these die to check that they are functioning properly (die-level functional test; DLFT), and then the wafers are cut up, and the good die are assembled into packages and connected to the package pins. Finally, the resulting packages are tested to ensure that the final product is functioning correctly. The purpose of die-level functional test is to avoid the expense of packaging bad die and to provide rapid feedback to the fabrication process by detecting die failures. The challenge for a decision-theoretic approach is to reduce the amount of DLFT (and the associated costs) while still providing process feedback. We describe a decision-theoretic approach to DLFT in which historical test data is mined to create a probabilistic model of patterns of die failure. This model is combined with greedy value-of-information computations to decide in real time which die to test next and when to stop testing. We report the results of several experiments that demonstrate the ability of this procedure to make good testing decisions, good stopping decisions, and to detect anomalous die. Based on experiments with historical test data from Hewlett Packard Company, the resulting system has the potential to

improve profits on mature IC products.

## 1. INTRODUCTION

Modern computer-integrated manufacturing lines provide many opportunities for applying data mining techniques. These lines contain many sensors and computer-controlled devices, so the information needed for intelligent control is available and control decisions can be implemented easily. Furthermore, in most current computer-integrated manufacturing lines, the supply of available sensor data far outstrips the ability of the existing control systems to digest and apply it. Consequently, the combination of data mining – to analyze data to build and update probabilistic models – and decision-theoretic control – to make decisions based on those models – can have a huge financial impact in reducing costs, increasing throughput, and raising profits.

In this paper, we report our experiences with one such application. We developed a decision-theoretic controller for one phase of the VLSI integrated circuit manufacturing process. The controller manages the die-level functional test (DLFT) process with the goal of maximizing the expected utility of the overall manufacturing process. We applied our expertise in VLSI testing to develop an influence diagram for the decision-making process. As a result of a related project at Hewlett-Packard Company, a detailed cost model had already been constructed. The remaining part of the influence diagram – the probabilistic model describing patterns of IC failures – was learned automatically from historical data. Our final system chooses actions via one-step greedy value-of-information, and, in simulation, it achieves substantial profit increases over the current control method.

The remainder of this paper is organized as follows. First, we describe the integrated circuit manufacturing process and the decision-making problem to be solved. Then we present our influence diagram and its probabilistic model of IC failures. The third section of the paper describes the training procedure and our experimental methods. The fourth section reports the results of several

experiments to understand and evaluate the behavior of the system and its various components. We summarize our conclusions in the final section.

## 2. IC MANUFACTURING AND TEST

An integrated circuit (IC) is an electronic circuit in which a number of devices are fabricated and interconnected on a single chip of semiconductor material. According to current manufacturing practice, integrated circuits are produced en masse in the form of processed silicon wafers. While still in wafer form the ICs are referred to as dice, an individual IC is called a die. The process of cutting the dice from wafers and embedding them into mountable containers is called packaging. Figure 1 is a simplified schematic of the major IC manufacturing steps.

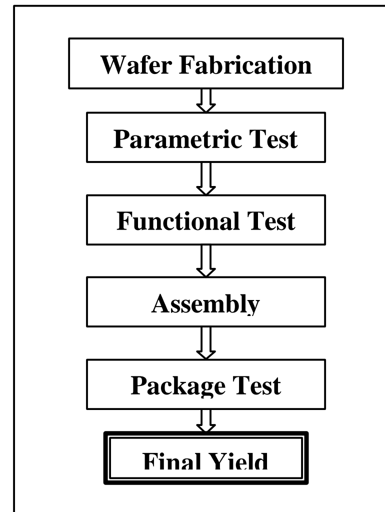


Figure 1: Major IC Manufacturing Steps

Brief descriptions of the manufacturing steps are provided below.

1. Fabrication:

A typical wafer is fabricated through a series of more than 100 process steps (Van Zant, 1997; Zorich, 1990). Virtually all of the processing is automated, but there are still many potential sources of failure that lead to defective wafers. An entire processing step may fail, in which case all of the die on a wafer will be bad. Many steps involve creating uniform thin layers on the wafer (e.g., by placing a drop of liquid material in the center and then spinning the wafer), and failures in this process can lead to radially-symmetric

patterns of failed die (e.g., in the center, in rings around the center, and most commonly, near the edges of the wafer). The wafers can also be scratched during automated handling, which creates linear failure patterns and edge defects. Finally, defects in the silicon substrate and in the applied materials, as well as dust particles, can create spatially uniform patterns of die failures.

2. Wafer parametric test:

Parametric tests measure physical and electrical parameters of the wafer such as electrical conductivity and behavior of individual sample components (transistors, capacitors, resistors). Although the above diagram depicts parametric testing as a distinct stage following wafer fabrication, in reality, parametric tests are performed throughout the fabrication process.

3. Die-level functional test (DLFT)

Functional testing typically occurs once the wafers are completely fabricated and the dice are completely formed and functional. Functional tests measure the operational quality of the individual dice. A large and expensive robotic machine presses electrical probes onto the die contacts, input signals are fed to the circuit, and output signals are measured. Functional testing simulates normal and abnormal operating conditions (e.g., high, normal, and low voltage tests). The conventional approach to DLFT is exhaustive wafer test, i.e., all dice on all wafers undergo DLFT. If a die fails the functional test, an ink dot is placed on it, so that it will not be packaged later. The decision to place the ink dot is called the "inking decision". After inking, the wafers are typically shipped to a separate location for packaging.

4. Packaging

To convert the wafers into packaged ICs, the wafers are cut into individual dice by a high-precision diamond saw. The resulting chips are mounted into packages, electrical contacts are bonded in place, and then a protective covering is added.

5. Package test

Once the ICs are packaged, they are tested again to ensure that the packaging process was successful. Package tests

usually repeat many of the functional tests that were performed during DLFT. The test results from package testing are used to decide which ICs to sell.

This brief summary shows that Die-Level Functional Test (DLFT) is not essential to the quality of the final product, because any failures will be detected during Package Test. Hence, the main purpose of DLFT is to reduce costs by avoiding packaging defective dice. A secondary purpose of DLFT is to provide rapid feedback to the manufacturing process by detecting and diagnosing faulty manufacturing processes. The challenge for a decision-theoretic approach is to reduce the amount of DLFT (and hence reduce its cost) while not appreciably increasing the costs of packaging and manufacturing.

## 3. DECISION-THEORETIC WAFER TEST

The goal of our project was to replace the exhaustive test policy with a decision-theoretic policy that decides in real time which die to test and when to stop testing. Because the DLFT is performed by a robotic tester, it can be reprogrammed to test the die in any desired order based on the results of previous tests. The decision-theoretic policy seeks to maximize overall profit by combining a probabilistic model of the spatial distribution of die failures with a utility model of the costs of the IC manufacturing process.

Figure 2 shows the influence diagram that we developed to model the manufacturing process. It contains a set of nodes $\{F_i, f_i, I_i, p_i, V_i\}$ for each die $i$ on the wafer. According to this model, the first step in the process is to generate a value for the variable $w$ at random according to $P(w)$. This latent variable $w$ is called the "wafer class", and it models the spatial correlations among the failures of individual die as a finite mixture model, as discussed in more detail below. The next step in the model is to choose a die to be tested. This is indicated by choosing one of the decision variables $F_i$ and setting its value to 1. The results of all previous functional tests are available when test $F_i$ is chosen, although the diagram does not show this. The result of the functional test for die $i$ is $f_i$, which is distributed according to $P(f_i|F_i, w)$.

After testing $F_i$, another die $F_j$ can be chosen and test result $f_j$ observed, and so on.

At some point, the testing process is terminated by setting all of the remaining $F_i$ to 0. The next step is to choose which die to ink. This is indicated by choosing values for all of the decision variables $I_i$ $(i = 1,\ldots,n)$. Depending on the inking decision $I_i$ and the functional test result $f_i$, the result of the final package test, $p_i$, is observed according to probability $P(p_i|I_i, f_i)$. Finally, we have included a wafer disposition decision $D$, which represents the decision to either package the (non-inked) die on the wafer or to scrap the entire wafer.

The utility of die $i$ is represented by $V_i$, and it includes the cost of the functional test (indicated by the arrow from $F_i$), the cost of packaging and package test (indicated by the arrow from $I_i$), and the selling price of the IC (indicated by the arrow from $p_i$, since the IC can only be sold if it passes package testing). The total utility of the wafer is represented by $V$, and it includes the sum of the $V_i$'s and also the cost of cutting up the wafer and shipping the die to the packaging facility (indicated by the arrow from $D$). The utility model is summarized in Table 1.

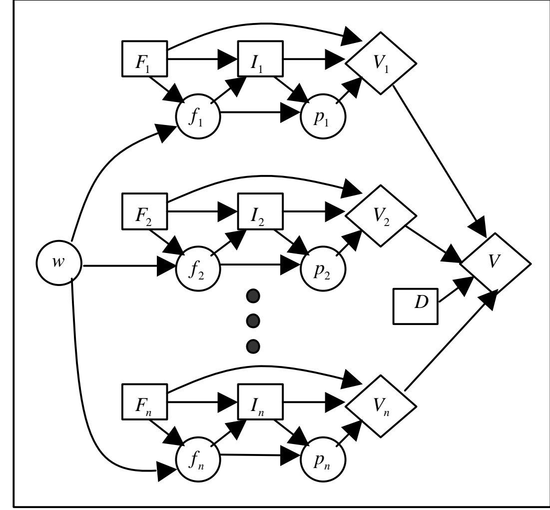


Figure 2: Wafer Test Influence Diagram

$W_j \equiv$ wafer$_j$
$N_d(W_j) \equiv$ # of dice on wafer$_j$
$N_f(W_j) \equiv$ # of functional tests on wafer$_j$
$N_p(W_j) \equiv$ # of package tests on wafer$_j$
$N_k(W_j) \equiv$ # of dice packaged from wafer$_j$
$Y(W_j) \equiv$ package yield of wafer$_j$
$c_f \equiv$ cost of single functional test
$c_p \equiv$ cost of single package test
$c_k \equiv$ cost of packaging a single die
$c_h \equiv$ single wafer handling costs
$v_k \equiv$ value of a single good package
value of package yield $\equiv V(Y(W_j)) = Y(W_j) * v_k$
total cost of functional tests $\equiv C_f = N_f(W_j) * c_f$
total cost of package tests $\equiv C_p = N_p(W_j) * c_p$
total cost of packaging $\equiv C_k = (N_k(W_j) * c_k) - c_h$
wafer profit $\equiv V(Y(W_j)) - C_f - C_p - C_k$

Table 1: Wafer Test Utility Model

Our cost model does not capture three important costs. First, the cost (or

benefit) of detecting fabrication problems is ignored. A reduction in die testing may reduce packaging costs, but increase fabrication costs because fabrication problems are not diagnosed as quickly. Second, the capital costs of purchasing and maintaining the die testing machines is ignored. If die-level testing can be significantly reduced, then fewer machines are needed, and these capital costs can be saved. Third, for some IC products, the current exhaustive testing policy becomes a rate-limiting bottleneck. A decrease in the number of dice tested per wafer means that more wafers can be tested, and therefore, wafer starts, throughput, and profits per unit time can increase.

To understand the probabilistic model of Figure 2, it is helpful to separate it out from the rest of the influence diagram (see Figure 3). This shows that we are modeling the die failures as being conditionally independent given a latent class variable $w$. This is the standard naïve Bayes' belief network that has already proven useful in diagnostic systems (Henrion, 1990) and learning and discovery systems (Dietterich, 1997, Cheeseman, Self, Kelly, Taylor, and Stutz, 1988). Each die $i$ corresponds to a fixed spatial location on the wafer, so if a particular location (e.g., near the edge) is prone to frequent failure, its value for $P(f_i=0|w)$ will be large. In the experiments reported below, we set $w$ to have four possible values, which gives us a mixture model with four multinomial components.

Note that unlike the applications of the naïve Bayes model in supervised learning, the class variable $w$ is not observed. Also note that unlike in the applications of naïve Bayes to clustering in Autoclass, we are not particularly interested in the structure of the classes that are learned. We simply view it as a convenient representation of the joint distribution $P(f_1, f_2, \ldots, f_n)$ of failures of the die on the wafers. We considered employing more sophisticated models of die failure, including models of "blobs" and "scratches". However, an analysis of the spatial statistics of die failure showed no evidence for such spatially-local patterns (Fountain, 1998).

To acquire the probabilities $P(w)$ and $P(f_i|w)$, we applied the EM algorithm to fit this naïve Bayes network to historical data from a mature IC product manufactured by Hewlett-Packard.

The probabilities *P(pi|fi)* were set to be identical for all die, and they were acquired from domain experts rather than from data. They represent the probability that a packaged die will fail the post-packaging functional test given that it passed (or failed) the die-level functional test.

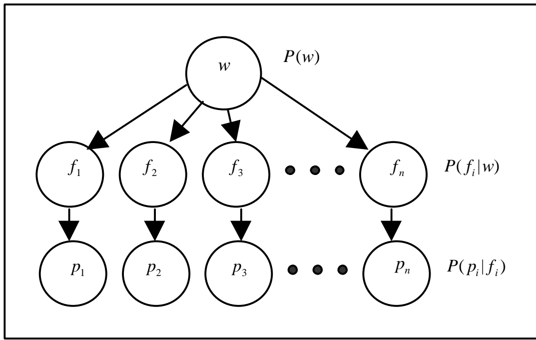


Figure 3: Wafer Test Belief Net

To apply this model to compute a DLFT policy, we perform an iterative one-step value of information (VOI) computation. Initially, all of the functional test decision nodes $F_i$ are set to 0 (no test). The die inking decisions, $I_i$, and the wafer disposition decision, $D$, are then made to maximize expected utility. Call this the "termination utility", $U_{term}$, because it is the utility of terminating functional testing and carrying out inking and packaging. The functional testing process then proceeds as follows. For each die that has not yet been tested, the expected utility $U_i$ of testing that one die and then making the inking and wafer disposition decisions is computed. If $\max_i U_i > U_{term}$, then test $i$ is performed, $f_i$ is observed, and the termination utility $U_{term}$ is recomputed. Otherwise, testing terminates, and the inking and disposition decisions are made.

One-step greedy VOI does not in general yield the optimal policy for a sequential decision problem. However, we expect it to perform very well in this particular problem because we are assuming that each testing action does not alter the wafer (e.g., by causing other die to fail). In addition, the inking decisions can be made independently, and the total utility is additive. Below, we will test experimentally how well greedy VOI works.

## 4. METHODS

Hewlett-Packard provided a data set for a mature IC product. The data consisted of the test results from 2400 wafers. We split this data into two separate data sets. Set 1 was used during model development and debugging. It consisted of 1200 wafers: 600 for

training and 600 for testing. Set 2 was used for a final test of the system, and it also consisted of 1200 wafers: 600 for training and 600 for testing. The wafers are grouped into "lots" of 24 wafers, which are kept together (in a cassette) during the manufacturing process. Because there is a strong possibility that wafers within a lot share the same defects, we divided the wafers according to entire lots, so each training data consisted of 25 lots and each testing set of 25 lots. The lots were kept in chronological order.

Tables 2 and 3 provide summary statistics for these two data sets. Each wafer contains 209 dice.

| Total Number Good Dice | 102288 |
| --- | --- |
| Total Number Dice | 125400 |
| Yield | 0.8157 |

Table 2: Test Wafer Statistics (Data Set 1)

| Total Number Good Dice | 101516 |
| --- | --- |
| Total Number Dice | 125400 |
| Yield | 0.8095 |

Table 3: Test Wafer Statistics (Data Set 2)

We applied the Expectation-Maximization (EM, Dempster, Laird, and Rubin, 1976) algorithm to train the naïve Bayes stochastic model. To determine the number of values of the latent variable *w*, we fit models with 1, 2, 4, 8, 12, 16, 20, and 24 values to the training data from Data Set 1 and measured the log likelihood of the wafers in the corresponding validation set. The best validation set log likelihood was achieved with a model containing 4 classes. For this model, the EM algorithm converged after no more than 50 iterations, which required less than three minute of CPU time. To obtain our final probability model for the decision-theoretic tester, we trained a 4-class model on the training data from Data Set 2.

## 5. RESULTS

We performed four experiments to address the following questions:

1. How well does the decision-theoretic approach perform compared to exhaustive testing, no testing, and optimal testing?

2. How well does greedy value of information (VOI) perform in deciding when to stop testing?

3. How does the system respond to abnormal wafers? Is the approach sensitive to process problems?

4. How robust is the system with respect to changes in utility parameters? Do changes in utility parameters result in rational responses from the system?

Each of these questions is addressed in turn.

### 5.1 PERFORMANCE

We compared the performance of the decision-theoretic testing policy ("DT") with three other policies: (a) the current exhaustive test approach ("Exhaustive"), (b) a policy that performed no tests and packaged all die ("Package All"), and (c) an optimal testing policy ("Oracle") that performs no testing but packages only those die that would have passed the functional test. The oracle policy provides an upper bound on the best that any implementable policy could do. We measured the net profit on the test set from Data Set 2.

| | Total Profit | Number Tested | Number Packaged |
|---|---|---|---|
| Exhaustive | 1184550 | 125400 | 102288 |
| Package All | 1226598 | 0 | 125400 |
| DT | 1229531 | 8210 | 121560 |
| Oracle | 1278600 | 0 | 102288 |

Table 4: Four Test Policies on Data Set 1

| | Total Profit | Number Tested | Number Packaged |
|---|---|---|---|
| Exhaustive | 1174900 | 125400 | 101516 |
| Package All | 1215211 | 0 | 125400 |
| DT | 1218309 | 5194 | 122292 |
| Oracle | ? | ? | ? |

Table 5: Four Test Policies on Data Set 2

The results from the model tests are presented in Tables 4 and 5 (for the test data in Data Set 1 and Data Set 2, respectively). The results show that the current exhaustive testing policy is the worst, and the DT policy is the best of the three implementable policies. Indeed, the DT policy achieves 96% of the profit that can be realized by the Oracle, and it gives a 3.8% improvement in profit over exhaustive testing.

The Package All policy produces almost as much profit as the DT policy, so a reasonable question is what advantage the DT policy has over Package All. The answer is that on wafers with high yield there may be little benefit. The problem with Package All is that process problems will not be detected until package test results become available. This can be a problem, because often packaging is performed at a location (e.g., Asia) far from the wafer fabrication plant (e.g., US). In these cases the delay in feedback and the costs of shipping and handling make the Package All policy risky. Furthermore, after packaging, the physical position of each die on the wafer is no longer known, so the spatial information provided by die-level test is lost. This spatial information is valuable for diagnosing fabrication problems. One of the benefits of the DT policy is that for good wafers it can produce profits comparable to those produced with a Package All policy, but for bad wafers, it can detect process problems while the wafers are still at the fabrication plant

## 5.2 EVALUATION OF THE GREEDY VOI STOPPING CRITERION

The decision-theoretic policy relies on a greedy value of information (VOI) computation to decide when to terminate testing. To determine how well this heuristic works, experiments were performed in which greedy VOI stopping was compared to the optimal stopping point. To determine the optimal stopping point, the decision-theoretic policy was modified to continue testing past the past the point of non-positive VOI until all dice were tested. Then the history of testing decisions was analyzed to find the moment at which the profit would have been maximized had the system stopped then. Profit includes the costs for functional tests up to that point and the

rewards obtained by making package decisions at that point.

| | Profit | Tested | Packaged |
|---|---|---|---|
| Exhaustive | 1184550 | 125400 | 102288 |
| VOI | 1229531 | 8210 | 121560 |
| Optimal Stop | 1231970 | 11206 | 119264 |

Table 6: Optimal Stopping: Results (Data Set 1)

Table 6 summarizes the results of this experiment on the test data from Data Set 1. The table shows that greedy VOI Stopping tests less than Optimal Stopping (performing only about 73% as many tests) and packages more (about 2% more packages). So Optimal Stopping spends a bit more on functional testing in order to reduce the number of bad dice packaged. Despite this difference, greedy VOI Stopping performs very well and realizes over 99% of the profit achieved by Optimal Stopping.

### 5.3 DETECTING PROCESS PROBLEMS

An interesting question for any testing policy is how it responds to abnormal wafers. Although the decision-theoretic approach was targeted towards a stable mature product, process problems are common, and abnormal wafer test results are often the first symptoms of such problems. An important question for a testing policy is how well can it recognize abnormally bad wafers?

To explore this issue, we discuss six wafers from Data Set 1 in detail. The first two wafers are typical “good” wafers with yields over 80%. The next two wafers are “bad”, with yields of less than 10%. The final two wafers are “mediocre”, with yields of 66% and 65%. The test wafers are described in Table 7. The simulation results are presented in Table 8.

| WID | NGD | Y |
|---|---|---|
| 1 | 178 | 0.85 |
| 2 | 180 | 0.86 |
| 3 | 17 | 0.08 |
| 4 | 2 | 0.01 |
| 5 | 138 | 0.66 |
| 6 | 118 | 0.56 |

Table 7: Test Wafer Statistics (WID = wafer id number, NGD = number of good dice on wafer, Y = yield)

| WID | NT | NP | CP | VOIP |
|---|---|---|---|---|
| 1 | 9 | 205 | 2068.25 | 2158.25 |
| 2 | 9 | 206 | 2093.25 | 2185.50 |
| 3 | 202 | 25 | 55.75 | 43.75 |
| 4 | 202 | 10 | -131.75 | -143.75 |
| 5 | 57 | 182 | 1568.25 | 1583.25 |
| 6 | 13 | 202 | 1318.25 | 1277.00 |

Table 8: Wafer Test Results (NT = number of die tested, NP = number packaged, CP = exhaustive test profit, VOIP = VOI test profit)

In Table 8, current profit is the profit realized under the exhaustive test policy. VOI profit is the profit realized under the decision-theoretic policy.

To visualize the testing behavior, wafer test maps are presented below for each wafer. These maps show the good dice, the dice that were tested according to the selective test policy, and the dice that were packaged according to this policy. In each map, green (light) encodes true, so green represents good dice, tested dice, and packaged dice. Red (dark) encodes bad dice, untested dice, and unpackaged dice.

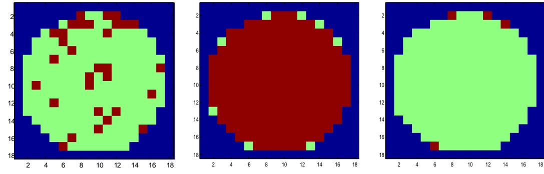

Figure 7: Detecting Process Problems: Wafer 1

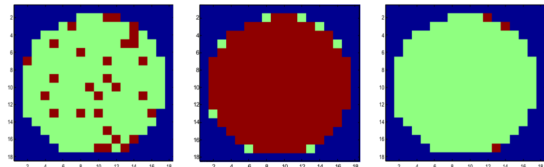

Figure 8: Detecting Process Problems: Wafer 2

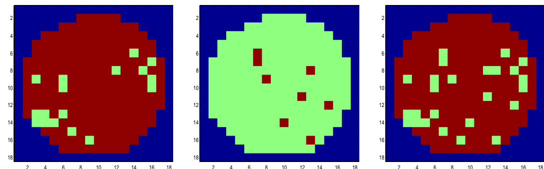

Figure 9: Detecting Process Problems: Wafer 3

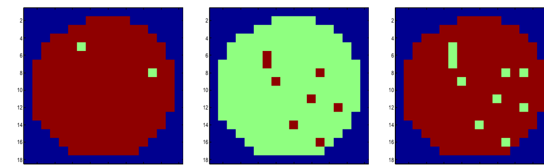

Figure 10: Detecting Process Problems: Wafer 4

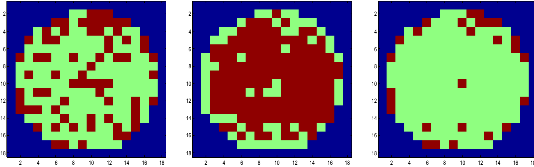

Figure 11: Detecting Process Problems: Wafer 5

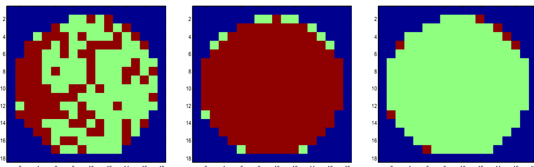

Figure 12: Detecting Process Problems: Wafer 6

Figures 7 and 8 correspond to good wafers, Figures 9 and 10 bad wafers, and Figures 11 and 12 mediocre wafers.

The analysis of these six wafers shows that the decision-theoretic policy is responsive to abnormal wafers. There is a direct relationship between wafer yield and the number of functional tests performed. The higher the yield, the fewer tests. This means that, for a stable and mature product, the system tests only a small fraction of the total dice. However, the system is sensitive to abnormal yields, which indicate process problems. On such wafers, the system tends to test more thoroughly. For wafers with extremely low yield, the system responds by testing almost all dice. This means that a minimal amount of resources are expended on good wafers, yet bad wafers are detected.

The reasons for this highly-desirable behavior are not entirely clear. One explanation is that because our IC product was a mature, high-yield product, the learned stochastic model is expecting to see good wafers. So when it encounters a bad wafer, its predictions concerning the untested wafers become uncertain (near 0.5), and it must do more testing to make good inking decisions. If our product had been one where half of the wafers were good and the other half very bad, then a bad wafer would not have been surprising, and the decision-theoretic policy would only perform enough tests to be confident of which kind of wafer it had. Then it would proceed to the inking decisions.

This analysis suggests that the output from the decision-theoretic approach could be fed into statistical process control (SPC) methods that routinely monitor for process problems. A straightforward extension to the current

SPC system would be to replace actual test measures with predicted measures. So, for example, rather than setting control limits around the actual functional test results, the control limits could be set around the predicted functional test results. Thus, the decision-theoretic approach provides dual benefits. First, it greatly reduces the requirement for testing resources. Second, it satisfies the requirement for prompt detection of process problems.

## 5.5 ROBUSTNESS TO CHANGES IN UTILITY PARAMETERS

One benefit of decision-theoretic methods is that changes in utility parameters should result in rational changes in performance without explicit re-engineering the learned models or control structures. To verify this, we experimented with changes to two of the utility parameters:

- Cost of performing a single functional test,
- Cost of packaging a single die.

For each of these parameters, a series of tests was performed in which the parameter of interest was swept through a range of values and performance on a testing scenario was measured. For these tests, we applied the stochastic model trained on the 600 training wafers from Data Set 1 to test 48 test-set wafers. Performance was measured by the number of functional tests performed, the number of dice packaged, the number of false positives (i.e., bad die packaged), and the number of true negatives (i.e., bad die not packaged).

### 5.5.1 Changes to Package Cost

In the first set of tests, the cost to package a single die was manipulated. Let $c_k$ represent the normal package cost. Then consider the effects of cutting the package cost in half ($.5c_k$) and of doubling the package cost ($2c_k$). The results are summarized in Table 9.

| | NT | NP | NFP | NTN |
|---|---|---|---|---|
| $.5c_k$ | 240 | 9854 | 1568 | 178 |
| $c_k$ | 796 | 9672 | 1386 | 360 |
| $2c_k$ | 2484 | 9312 | 1026 | 720 |

Table 9: Robustness Tests: Changes to Package Cost (NT = number of die tested, NP number packaged, NFP = false positives, NTN = true negatives)

The results show that when it is relatively inexpensive to package dice,

the system packages more and tests less. As the package cost increases, false positives become more expensive, so more tests are performed to reduce this risk. Thus, with respect to changes in package cost, the system performs rationally by adjusting its testing and package decisions to maximize expected profits.

### 5.5.2 Changes to Functional Test Cost

A second set of tests was performed in which the functional test cost was manipulated. Let $c_f$ represent the current cost of a single functional test. Then consider the effects of setting the functional test cost at $.1c_f$, $.67c_f$, $c_f$, $1.33c_f$, $1.67c_f$, $2c_f$, and $10c_f$. The results are summarized in Table 10.

| | NT | NP | NFP | NTN |
|---|---|---|---|---|
| $.1C$ | 10032 | 8286 | 0 | 1746 |
| $.67C$ | 1509 | 9485 | 1199 | 547 |
| C | 796 | 9672 | 1386 | 360 |
| $1.33C$ | 321 | 9835 | 1549 | 197 |
| $1.67C$ | 240 | 9854 | 1568 | 178 |
| $2.0C$ | 169 | 9912 | 1626 | 120 |
| $10C$ | 0 | 10032 | 1746 | 0 |

Table 10: Results of Changes to Functional Test Costs (NT = number of die tested, NP = number packaged, NFP = false positives, NTN = true negatives).

This table shows that the system behaves rationally by adjusting its testing and packaging decisions to reflect changes in cost parameters. When the package cost is increased, the system tests more to avoid wasting resources by packaging bad dice. On the other hand, when the functional test cost is increased, the system tests less and packages more. If the test cost is set sufficiently low, then the system tests all dice. If the test cost is set sufficiently high, then the system tests none of the dice. In none of the test scenarios was it profitable to miss a good die, so the number of true positives was always equal to the total number of good dice, and the number of false negatives was always zero. This behavior is the result of two factors. First, given the quality of the wafers in the training set, all dice had a reasonable prior probability of being good. Second, the value of a good package was sufficient to justify packaging all dice based on these prior probabilities.

## 6. CONCLUDING REMARKS

The experiments in this paper have demonstrated that manufacturing data can be mined to produce effective decision-theoretic control methods. Furthermore, these methods can produce substantial improvements in the die-level functional test stage of VLSI IC manufacturing. Specifically, the experiments have shown the following:

1. The decision-theoretic policy produced more net profit than either the exhaustive test policy or the a policy of performing no die-level testing.

2. The greedy VOI stopping criterion produced near-optimal stopping behavior.

3. The decision-theoretic policy is able to detect abnormal wafers, and it responds by testing them more thoroughly. Hence, although the cost model does not reflect the value of die-level test for detecting fabrication problems, the decision-theoretic policy still detects these problems well. Existing process control statistics could be based on the predictions of the stochastic model.

4. The decision-theoretic policy is robust to changes in testing costs and packaging costs.

These experiments demonstrate that the decision-theoretic approach to die-level functional test has the potential to reduce testing costs, increase wafer starts, and improve the bottom line. In addition, the method is easy to implement – the EM training was straightforward and efficient, and the cost model can easily be changed to reflect changes in market conditions. We believe that in this and many other computer-integrated manufacturing applications, data mining and decision-theoretic methods have an important role to play.

**References**

Cheeseman, P.; Self, M.; Kelly, J.; Taylor, W.; Freeman, D.; and Stutz, J. 1988. Bayesian classification. In *Proceedings of the Seventh National Conference on Artificial Intelligence*, San Mateo, CA:Morgan Kaufmann 607-611.

Dempster, A. P.; Laird, N. M.; and Rubin, D. B. 1976. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society* B 39:1-38.

Dietterich, T. G. 1997. Machine-Learning research: four current directions. *AI Magazine*, Menlo Park, CA: AAAI Press 18(4): 97-136.

Fountain, T. 1998. *Just enough die-level functional test: Optimizing IC test via machine learning and decision theory.* Doctoral dissertation, Department of Computer Science, Oregon State University.

Van Zant, P. 1997. *Microchip fabrication: a practical guide to semiconductor processing, third edition.* New York: McGraw-Hill.

Zorich, R. 1991. *Handbook of Quality Integrated Circuit Manufacturing*. San Diego, CA:Academic Press.