Modeling the Economics of Testing: A DFT Perspective

Pranab K. Nag, Anne Gattiker, Sichao Wei, R.D. Blanton, and Wojciech Maly
Carnegie Mellon University

Deciding whether and how to use DFT is difficult because the relationship of costs and benefits is far from being well understood. The tradeoff-modeling procedure presented here helps users fill in the missing links in the DFT decision-making process.

To DFT or not to DFT? The answer depends on who is asking and who is answering. Designers might view DFT as a necessary evil that prevents them from achieving the best possible design—“best” meaning high performance, minimum hardware complexity, and low power. From the test engineer’s perspective, “best” may be quite different—controllability, observability, and diagnosability being the desired attributes. From a manager’s or marketing perspective, “best” includes both sets of attributes—but in the form of costs incurred versus benefits gained. Likewise, for us “best” means an end product that maximizes the cost-to-benefit ratio, with the decision to use DFT ultimately based on its impact on profit.1

Decision-makers typically make test tradeoffs using models that mainly represent direct costs such as test generation time and tester use. Analyzing a test strategy’s impact on other significant factors such as test quality and yield learning requires an understanding of the dynamic nature of the interdomain dependencies of test, manufacturing, and design. Our research centers on modeling the tradeoffs between these domains. To answer the DFT question, we

- developed the Carnegie Mellon University (CMU) Test Cost Model, a DFT cost-benefit model;
- derived inputs to the model for various IC cases with different assumptions about volume, yield, chip size, test attributes, and so forth; and
- studied DFT’s impact on these cases.

We used the model to determine the domains for which DFT is beneficial and for which DFT should not be used. The model is a composite of simple cause-and-effect relationships derived from published research. It incorporates many factors affecting test cost, but we don’t consider it a complete model. Our purpose here is to illustrate the necessity of using such models in assessing the effectiveness of various test strategies.2

Test cost model

To develop our model, we first identified significant factors contributing to the cost of testing any IC. We then identified model components that DFT alters. The cost factors combined with the components give us the capability of estimating the DFT cost-benefit ratio.

Basic structure

An IC must go through several test phases: wafer and package test, burn-in, and so on. The CMU model describes a single phase in a complete IC-testing process. We obtain the cost for
other phases by appropriately modifying the model’s input values.

As Figure 1 shows, we assume that test cost consists of four main components—preparation cost, execution cost, test-related silicon cost, and imperfect-test-quality cost. Hence, we compute test cost $C_{\text{test}}$ (per good die) as

$$C_{\text{test}} = C_{\text{prep}} + C_{\text{exec}} + C_{\text{silicon}} + C_{\text{quality}}$$  \hspace{1cm} (1)

$C_{\text{prep}}$ captures fixed costs of test generation, tester program creation, and any design effort for incorporating test-related features (all non-recurring costs, including software systems). Test generation cost depends on die area and personnel cost. Tester program preparation is a function of test generation cost. Test-related design effort is a function of the extra silicon area required to incorporate testability-enhancing features such as scan and BIST.

$C_{\text{exec}}$ consists of costs of test-related hardware such as probe cards and cost incurred by tester use. Tester-use cost depends on factors including

- tester setup time,
- test execution time (as a function of die area and yield), and
- capital cost of the tester (as a function of die area and number of I/O pins) and capital equipment depreciation rate.

$C_{\text{silicon}}$ is the cost required to incorporate DFT features. We model this cost as a function of wafer size, die size, yield, and the extra area required by DFT. We model differences in attributes such as area, yield, and fault coverage between DFT and non-DFT designs by using appropriate multiplying or additive factors.

$C_{\text{quality}}$ is often ignored but is probably one of the most important test cost components. Another major cost component, albeit implicit (and not represented in Figure 1), is DFT’s potential impact on yield learning. Yield learning is the process by which causes of yield losses are discovered and corrected, resulting in progressive yield increases over time.

The equations we present here represent cause-and-effect relationships between various test, manufacturing, and design attributes. In some cases, we derived them in an ad hoc fashion from results of our prior work and from expert opinions. In other cases, we selected equations from previously published results.

**Test preparation.** We compute test preparation cost as

$$C_{\text{prep}} = (C_{\text{test,gen}} + C_{\text{test,prog}} + C_{\text{DFT,design}})/Y$$  \hspace{1cm} (2)

where $C_{\text{test,gen}}$ is the test-pattern generation cost, $C_{\text{test,prog}}$ is the tester-program preparation cost,
\( C_{\text{DFT\_design}} \) is the additional design cost for DFT, and \( Y \) is the yield. Each cost component is calculated per die, and dividing by \( Y \) ensures that \( C_{\text{prep}} \) is calculated per good die. (\( C_{\text{DFT\_design}} \) is 0 when the die does not contain DFT.)

We calculate test-pattern-generation cost (per die) as

\[
C_{\text{test\_gen}} = \left( R_{\text{person\_hour}} T_{\text{test\_gen}} \right) / V
\]

where \( R_{\text{person\_hour}} \) is the cost of one person-hour for test generation, \( T_{\text{test\_gen}} \) is test generation time, and \( V \) is manufacturing volume (the total number of ICs produced).

Total generation time is computed as

\[
T_{\text{test\_gen}} = K_{\text{test\_gen}} A
\]

where \( K_{\text{test\_gen}} \) is a constant multiplier that relates test generation time to the IC die area \( A \). This multiplier facilitates flexibility in the choice of inputs to the model. If we know the test generation time for a die of a given area, we can solve equation 4 for \( K_{\text{test\_gen}} \). Then, we calculate test generation times for varying die sizes on the assumption that test generation time increases exponentially with area.

We assume the cost associated with tester program preparation is proportional to the cost of test generation. The total cost of tester program preparation is therefore

\[
C_{\text{test\_prog}} = \beta_{\text{test\_prog}} C_{\text{test\_gen}}
\]

where \( \beta_{\text{test\_prog}} \) is a factor of less than 1 that models the difficulty of translating test vectors to the tester’s format.

**Test execution.** We calculate test execution cost (per good die) as

\[
C_{\text{exec}} = \left( C_{\text{hw}} + C_{\text{tester}} \right) / Y
\]

where \( C_{\text{hw}} \) is the cost per die of using hardware for testing (excluding the tester), and \( C_{\text{tester}} \) is the cost per die of using the tester.

To model the finite life of probe (test) cards, we calculate hardware cost as

\[
C_{\text{hw}} = \left( Q_{\text{probe}} \frac{V}{N_{\text{probe\_life}}} \right) / V
\]

where \( Q_{\text{probe}} \) is the unit price of a probe card, and \( N_{\text{probe\_life}} \) is the number of ICs a probe card can test.

We model tester time cost as the product of the cost of tester use per second and the average IC test time. We assume active-tester cost is greater than inactive-tester cost. The larger active-tester cost primarily reflects the cost of tester operators and electricity. Thus, tester-use cost is the average of active- and inactive-tester costs weighted by the tester utilization factor.

We calculate tester-use cost (per die) as

\[
C_{\text{test\_use}} = [R_{\text{act}} + R_{\text{inact}} (1 - \beta_{\text{util}})] T_{\text{test}}
\]

where \( R_{\text{act}} \) is the cost rate (dollars per second) for an active tester, \( \beta_{\text{util}} \) is the tester utilization factor (the percentage of time the tester is actually testing ICs), \( R_{\text{inact}} \) is the cost rate for an inactive tester, and \( T_{\text{test}} \) is the average time (seconds) required to test a single IC.

We calculate the rate for an active tester as

\[
R_{\text{act}} = R_{\text{inact}} (1 + \beta_{\text{act}})
\]

where \( \beta_{\text{act}} \) is the fractional increase in rate when a tester is actively used.

The cost rate for an inactive tester is

\[
R_{\text{inact}} = \left( Q_{\text{capital}} \beta_{\text{depr}} \right) / \left( T_{\text{sec\_per\_yr}} \right)
\]

where \( Q_{\text{capital}} \) is the tester price, \( \beta_{\text{depr}} \) is the annual depreciation rate, and \( T_{\text{sec\_per\_yr}} \) is the total seconds in one year.

We assume the tester capital cost is proportional to the pin count of the IC under test, implying that tester cost is a function of die area. A tester’s unit price is

\[
Q_{\text{capital}} = K_{\text{capital}} K_{\text{pins}} \sqrt{A}
\]

where \( K_{\text{capital}} \) is tester price per pin, and \( K_{\text{pins}} \) relates the number of pins to the IC die area \( A \). We calculate tester utilization as

\[
\beta_{\text{util}} = T_{\text{test}} V / \left( T_{\text{sec\_per\_yr}} \left( T_{\text{test}} V / T_{\text{sec\_per\_yr}} \right) \right)
\]

We assume test execution time is proportional to the square of the die area. We also assume the average test time depends on yield,
because test time is shorter for a failing die than for a good or defect-free die. This is true if testing terminates upon first failure. However, in some cases, the test cannot terminate immediately but must continue until a particular test-sequence portion completes. If a group of dies are tested together, in the worst case the entire test set must be applied. A constant multiplier in the equation facilitates flexible input-value assignment to reflect various scenarios.

Consequently, the average time required to test a single IC is

\[ T_{test} = T_{setup} + [Y + \beta_{fail}(1-Y)] K_{time} A^2 \]  

(13)

where \( T_{setup} \) is the setup time for an IC on the tester, \( \beta_{fail} \) is the average percentage of good-die test time required to test a defective IC, and \( K_{time} \) is a constant multiplier that relates test time to IC die area \( A \). We can extract both \( \beta_{fail} \) and \( K_{time} \) through regression analysis of historical data on test execution times of various products. If \( \beta_{fail} \) is less than 1, the entire test sequence need not be applied to a failing die.

**Imperfect test quality.** We calculate the cost of imperfect test quality as

\[ C_{quality} = C_{loss_perform} + C_{escape} + C_{loss_yield} \]  

(14)

where \( C_{loss_perform} \) is the loss in profit from performance degradation (because of added DFT circuitry), \( C_{escape} \) is the cost of test escape, and \( C_{loss_yield} \) is the cost of good dies being deemed faulty by imperfect testing. Here, we model only escape cost. For simplicity in modeling, we don’t consider performance- or yield-loss costs caused by the failure of good dies. Modeling performance-loss cost requires an in-depth study of relationships between circuit attributes and performance, especially in the presence of DFT circuitry. Modeling yield-loss cost requires careful collection and analysis of failure data.

The test escape rate is a function of fault coverage and the fault occurrence rate (or simply the yield) at a particular stage of test. Williams and Brown \(^3\) give the escape rate, the ratio of the defective ICs that pass testing to all the ICs that pass testing, as

\[ E = 1 - Y^{1-f} \]  

(15)

where \( f \) is fault coverage. The escape cost must be allocated to the good ICs. We calculate the cost contribution of escape for every good IC as

\[ C_{escape} = M_{cost} \alpha_{escape} E \]  

(16)

where \( M_{cost} \) is the manufacturing cost (including the processing cost for a given technology) of one tested good IC, and \( \alpha_{escape} \) is the multiplying factor representing the risk incurred by accepting a defective IC as good. In other words, the risk factor represents the economic penalty for allowing a defective IC to escape testing. Depending on the importance attached to escaping ICs, a manufacturer may consider \( \alpha_{escape} \) to be as much as 10 for each testing stage.

We express the IC’s manufacturing cost as the sum of the basic test cost and the fabrication cost:

\[ M_{cost} = C_{basic} + C_{fab} \]  

(17)

where \( C_{basic} \) is the first two terms of equation 1 \( (C_{prep} + C_{exec}) \) and \( C_{fab} \) is

\[ C_{fab} = \frac{Q_{wafer} R_{wafer}^{2} \beta_{waf_die} A}{\pi} \]  

(18)

where \( Q_{wafer} \) is the wafer cost, \( R_{wafer} \) is the wafer radius, and \( \beta_{waf_die} \) is the percentage of wafer area that can be divided into dies.

**DFT impact on test cost**

For ICs with DFT features, we modified the cost model to reflect the impact of DFT. The modifications fall into two categories: basic DFT impact and DFT impact on yield learning. We further break the first category into additive costs and alpha factors.

**Additive costs.** We first add the cost of silicon overhead to equation 1 and the cost of DFT-related design to equation 2. The cost of test-related silicon is the cost of the extra silicon required to implement DFT circuitry. The cost of extra silicon (per good die) due to the IC area increase is
where $A_{\text{DFT}}$ and $Y_{\text{DFT}}$ are the area and yield of the die with DFT, and $A_{\text{no DFT}}$ and $Y_{\text{no DFT}}$ are the corresponding parameters without DFT. (Equation 25 describes the relationship between $A_{\text{DFT}}$ and $A_{\text{no DFT}}$.)

In equation 19, the silicon unit area cost is the wafer manufacturing cost divided by the wafer area and the factor $\beta$. (The fraction does not equal 1 because the wafer is essentially circular and the dies are rectangular.) The equation also captures the detrimental effect of DFT silicon area on yield by using a “yielded” die area—die area divided by yield. Because we wish to consider only the increase in area and decrease in yield that are directly attributable to the DFT circuitry, we subtract the yielded die area of the nominal die (the same die without DFT) from the yielded area of the die with DFT.

We use Seed’s equation\(^4\) to model yield:

$$Y = \frac{1}{1 + AD} \quad (20)$$

where $D$ is defect density. Our model is not limited to this yield expression; several others would be suitable.\(^5\)\(^6\)

We calculate the cost of the extra design effort required for DFT as

$$C_{\text{DFT_design}} = \frac{K_{\text{die_design}}(A_{\text{DFT}} - A_{\text{no DFT}})}{V} \quad (21)$$

where $K_{\text{die_design}}$ is the design cost per unit area of the design.

**Alpha factors.** We alter four of the cost model’s most important quantities by introducing four multiplier factors, which we call alpha factors. These factors affect test generation time (equation 4), test execution time (equation 13), tester capital cost (equation 11), and area overhead for DFT (equation 25).

Assuming that test generation time is less for the DFT case than for the no-DFT case, we introduce test generation factor $\alpha_{\text{DFT_test_gen}}$. Thus, we change equation 4 as follows:

$$T_{\text{test_gen}} = \alpha_{\text{DFT_test_gen}} K_{\text{test_gen}} e^{A_{\text{DFT}}} \quad (22)$$

Similarly, we assume IC test time with DFT is less than without DFT. The second term of Equation 13 becomes

$$\alpha_{\text{DFT_t_time}} [Y + \beta_{\text{fail}}(1 - Y)] K_{\text{t_time}} A_{\text{DFT}}^2 \quad (23)$$

To model the possibility that the tester capital cost for ICs with DFT may be lower than the cost for those without DFT (because of different pin and speed requirements) we modify equation 11 with multiplier factor $\alpha_{\text{DFT_capital}}$:

$$Q_{\text{capital}} = \alpha_{\text{DFT_capital}} K_{\text{capital}} K_{\text{pins}} \sqrt{A_{\text{DFT}}} \quad (24)$$

Finally, we obtain the die area for the IC with DFT by using area overhead factor $\alpha_{\text{DFT_area}}$:

$$A_{\text{DFT}} = (1 + \alpha_{\text{DFT_area}}) A_{\text{no DFT}} \quad (25)$$

The DFT alpha factors can range from 0.0 to 1.0 depending on the characteristics of the die being tested.

**Yield learning.** So far, we’ve discussed DFT’s effect on test cost—the dollars added to the untested-product cost for testing. Now we focus on test-related profit benefits. Profit benefits cross the design, test, and manufacturing domain boundaries and cannot be modeled as additions to or subtractions from test costs. Two key potential profit benefits associated with DFT are an increased yield-learning rate and decreased time to market.

DFT-related increases in yield-learning rates result from an improved ability to perform defect diagnosis and feed the results back to the manufacturing process for defect reduction or to the design team for design-for-manufacturability enhancements. Time-to-market benefits, on the other hand, have several sources. One example is ease of test preparation. Equation 22 reflects the direct effect on test generation cost of relatively easy test preparation. However, easier test preparation may also help get the product to market faster. Yield learning also greatly influences time to market. For a product to be marketable, the yield must reach a viable value. Time to market is also a function of production volume, ramp rate, product demand, and time required for design...
and verification. In our current model, we do not consider time to market, but Lu and Wu discuss the topic.  

We assume that yield is a fixed value for the production period under consideration. In other words, we model the yield-learning curve using an average yield value, \( Y_{avg} \), such that

\[
Y_{avg} = \frac{1}{T} \int_0^T Y(t) \, dt
\]  

(26)

where \( T \) is the total time period under consideration, and \( Y(t) \) is the yield-versus-time function. Thus, we can model a greater yield-learning rate attributable to DFT as a larger \( Y_{avg} \) for the DFT case. This implies a lower average value for defect density in the DFT case than in the non-DFT case (see equation 20).

Next we introduce the defect-density parameters. For the DFT case, we calculate defect density as

\[
D_{DFT} = (1 - \alpha_{YLF}) D_{no\_DFT}
\]  

(27)

where \( \alpha_{YLF} \) is the yield-learning factor, which reflects the reduction in defect density, and \( D_{no\_DFT} \) is the defect density for the no-DFT case. We then use the equations presented earlier to calculate testing cost, using two different defect-density values corresponding to ICs with and without DFT.

DFT features such as full scan have great potential to positively affect diagnosis. Full scan reduces sequential automatic test generation to the combinational case that makes internal circuit nodes more accessible to testing. It is reasonable to expect increased diagnostic resolution, which localizes a larger fraction of possible faults to a smaller IC area. Because defect localization is one of the most time-consuming processes in modern semiconductor manufacturing, this increased resolution speeds up the failure-analysis cycle.

Currently, no known model relates DFT features to yield learning. The actual value of \( \alpha_{YLF} \) depends not only on the chosen DFT strategy, but also on the failure-analysis practices of a particular manufacturer. Despite the limited published knowledge of the relationship, in the next section, we use our model to explore a range of \( \alpha_{YLF} \) values to speculate about its impact on test cost.

Discussion. We have modeled several attributes as functions of die area. Design and test generation costs could perhaps be more accurately modeled as functions of circuit complexity, which in turn could be modeled as a proportional function of the number of transistors. The yield impact of extra circuitry is also naturally a function of the number of added transistors. However, for simplicity, we describe these relationships in terms of die area. We can thus simply assess the cost penalty for additional DFT-related transistors, both in terms of die area and yield.

Implicitly, we assume the design is not pad limited, so that the die area is proportional to the number of transistors. If the die were pad limited, the extra DFT circuitry might be cost free in terms of die area but would still adversely affect yield. We also assume the model is tuned to a particular technology node (for example, gate length). A different technology node would produce a different proportionality constant between the number of transistors and the die area.

Model parameters

We assumed various input values to perform simulation experiments with our test cost model. Essentially, the model uses three categories of parameters: independent variables, constants, and DFT alpha factors.

Independent variables

The key independent variables we used for our studies are production volume \( V \) and die area \( A_{no\_DFT} \). (Recall that the die area calculation for DFT designs is based on \( A_{no\_DFT} \) in equation 25.) We explored a spectrum of values for \( V \), ranging from small, ASIC-like volumes (100,000) to large, microprocessor-like volumes (100 million). For \( A_{no\_DFT} \), we explored die areas ranging from 0.5 cm\(^2\) to 4 cm\(^2\).

Constants

We knew the choice of values for the constants introduced earlier would greatly affect our study’s conclusions. Therefore, to ensure that these values reasonably reflected reality, we verified them through questionnaires circulated among industry experts.

Table 1 lists the nominal values for the con-
stants used in our study. Figures 2 and 3 (next page) illustrate how the values given in Table 1 translate to the key parameters of our model: test generation time, test execution time, and yield. Each of these key parameters is plotted as a function of die area.

DFT factors

DFT’s impact is difficult to estimate without specific attributes of the IC design. Moreover, there has been little prior work on modeling DFT-related design attributes. Hence, we adopted a range of what we think are reasonable values for the parameters in equations 22 through 25. For equations 22 through 24, the worst-and best-case values we investigated for the first three DFT factors were 0.5 and 0.05, respectively. The values we analyzed for area-overhead parameter $\alpha_{DFT\_area}$ in equation 25 were 0.13 and 0.05 for the worst and best cases. (The worst-case value 0.13 was based on our survey results.) We chose the wide (order-of-magnitude) range of values for the other alpha factors to explore the many situations that can arise in a real manufacturing scenario. We used fault-coverage values of 99% and 95%, respectively, for ICs with and without DFT. The range of risk factors ($\alpha_{\text{escape}}$ in equation 16) we explored was between 1 and 20. Finally, for yield-learning analysis, the yield-learning factor ($\alpha_{\text{YLFin}}$ in equation 27) ranged from 0 (no learning) to 0.6 (a high learning rate).

We did not vary the parameters listed in Table 1 because they are, by construction, independent of DFT-related parameters. In reality, interdependencies may exist between these

**Table 1. Constant values used in the CMU Test Cost Model.**

<table>
<thead>
<tr>
<th>Equation numbers</th>
<th>Constant</th>
<th>Value</th>
<th>Remark</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>$R_{\text{person_hour}}$</td>
<td>$50/hr.$</td>
<td></td>
</tr>
<tr>
<td>4, 22</td>
<td>$K_{\text{test_gen}}$</td>
<td>5,122 hr.</td>
<td>See Figure 2</td>
</tr>
<tr>
<td>5</td>
<td>$\beta_{\text{test_prog}}$</td>
<td>0.2</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>$Q_{\text{probe}}$</td>
<td>$1,000$</td>
<td>$N_{\text{probe_life}}$ 100,000 dies</td>
</tr>
<tr>
<td>9</td>
<td>$\beta_{\text{act}}$</td>
<td>0.25</td>
<td></td>
</tr>
<tr>
<td>10, 12</td>
<td>$\beta_{\text{depr}}$</td>
<td>0.40</td>
<td>First-year depreciation</td>
</tr>
<tr>
<td></td>
<td>$T_{\text{sec_per_yr}}$</td>
<td>31,536,000 s</td>
<td></td>
</tr>
<tr>
<td>11, 24</td>
<td>$K_{\text{capital}}$</td>
<td>$7,800$/pin</td>
<td>$2$ million for a 256-pin tester</td>
</tr>
<tr>
<td></td>
<td>$K_{\text{tens}}$</td>
<td>141/cm</td>
<td>100 pins for a 0.5-cm$^2$ die</td>
</tr>
<tr>
<td>13, 23</td>
<td>$K_{t_time}$</td>
<td>4.8 s/cm$^2$</td>
<td>See Figure 2</td>
</tr>
<tr>
<td>18</td>
<td>$Q_{\text{waf}}$</td>
<td>$1,300$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$R_{\text{waf}}$</td>
<td>100 mm</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$\beta_{\text{waf_die}}$</td>
<td>0.90</td>
<td></td>
</tr>
<tr>
<td>20, 27</td>
<td>$D$</td>
<td>0.222/cm$^2$</td>
<td>See Figure 3</td>
</tr>
<tr>
<td>21</td>
<td>$K_{\text{die_design}}$</td>
<td>$500,000$/cm$^2$</td>
<td>5 person-years for a 1-cm$^2$ die</td>
</tr>
</tbody>
</table>

**Figure 2. Test generation time and test execution time as functions of die size.**
and the alpha parameters. However, introducing second-order dependencies complicates extracting the model parameters from real-life data. We extracted most of the parameters in Table 1 fairly easily from accounting data and from published sources such as the *International Technology Roadmap for Semiconductors* (1997, 1999, and 2000).

Three constant parameters in Table 1, however, need special attention: $K_{\text{test\_gen}}$, $K_{\text{t\_time}}$, and $K_{\text{die\_design}}$. They can be difficult to extract because historical data is hard to find. If the numbers at a company using the CMU Test Cost Model are significantly different from ours, the company’s specific conclusions will also be different.

**Applying the model to DFT**

We can assess the test cost model’s relevance by seeing how it answers our opening question: To DFT or not to DFT? To answer, we must explore the domain of model parameters that cause profit loss or benefit when DFT is applied.

**Basic costs**

Industrial practice and prior publications suggest that die area and manufacturing volume are the most important factors in answering our question. Therefore, our investigations focus on the manufacturing-volume-versus-die-area plane. There, we look for the domain in which DFT is beneficial (that is, the die cost with DFT is less than the die cost without DFT) and the domain in which DFT provides no cost benefits. Figure 4 shows the result. We obtained this plot by assigning the best value to the four DFT alpha factors (the value that most favorably affects ICs with DFT): $\alpha_{\text{DFT\_test\_gen}} = \alpha_{\text{DFT\_t\_time}} = \alpha_{\text{DFT\_capital}} = \alpha_{\text{DFT\_area}} = 0.05$.

At the other extreme are the worst-case val-
ues for the DFT factors: \( \alpha_{DFT\_test\_gen} = \alpha_{DFT\_time} = \alpha_{DFT\_capital} = 0.50 \), and \( \alpha_{DFT\_area} = 0.13 \). This value assignment considerably reduces the domain in which DFT has an advantage. Figure 5 compares the best- and worst-case scenarios. The region between the two plotted curves is the domain in which DFT’s benefits are uncertain and the DFT question cannot be answered.

The uncertainty is caused by the fact that we assign the alpha factors a broad range of values, rather than modeling them as precise functions of product and test environment attributes. However, the uncertainty region covers only a portion of the considered domain. Despite a relatively large range of assumed DFT alpha factors, we can still find regions in which the answer to our key question is clear. Consequently, we reach these conclusions:

- The proposed cost model may be useful in the DFT planning process despite the uncertainty caused by an incomplete understanding of key test cost relationships.
- Decreasing the size of the uncertainty region (and consequently decreasing uncertainty in the DFT-application strategy) requires more research on DFT’s impact on test generation time, test execution time, tester complexity, and area overhead.

To demonstrate the sensitivity of the uncertainty-region boundaries to the model parameters, Figure 6 (next page) shows cases in which one of the four DFT alpha factors is modified from its original extreme value to its extreme opposite value. This has the effect of moving the best-case curve toward the worst-case curve and vice versa.

Table 2 (next page) shows all the combinations of parameter values used in computing the results (the inner two curves marked 1b and 2b in the figures) shown in Figures 6a through 6d (next page). The parameters for the outer two curves in each figure (marked 1a and 2a) are the same as in Figure 5 (extreme-case values) and are also listed in Table 2.

Figure 6a illustrates the sensitivity of the best- and worst-case boundaries to \( \alpha_{DFT\_test\_gen} \). As we expected, the impact of test generation effort on total test cost is considerable and cannot be ignored. On the other hand, a change in test execution time factor \( \alpha_{DFT\_time} \) has little effect on the boundaries, as Figure 6b shows.

Test execution cost is directly proportional to tester capital cost. Thus, any change in tester capital cost affects the boundaries between the DFT and no-DFT scenarios, as Figure 6c shows.

Finally, Figure 6d shows that area-overhead factor \( \alpha_{DFT\_area} \) has the greatest effect on the worst- and best-case boundaries. Area overhead also implies costs incurred by extra design effort, yield decrease, and silicon overhead. Thus, we
have much to gain by minimizing DFT overhead. (We varied $\alpha_{DFT\_area}$ over a smaller range of values than the other parameters to reflect typical industrial situations.)

Escape cost

The results just presented lead to the conclusion that for any set of reasonable assumptions, DFT is advantageous mainly for low volumes. However, the examples took into account only factors that explicitly contribute to cost. Now we study DFT’s impact on imperfect test quality (the fourth branch of the model shown in Figure 1).

To illustrate how $C_{quality}$ affects the answer to the DFT-or-no-DFT question, we assign worst-case values to the four DFT alpha factors. Figure 7 shows the impact of escape for a wide range of risk factors. The DFT/no-DFT boundary shifts dramatically with increasing risk factors. Even for $\alpha_{escape} = 10$, the domain where DFT should be used enlarges considerably. The

![Graphs showing the impact of escape cost on the DFT/no-DFT boundary.](image)

**Figure 6.** Sensitivity of the best-case and worst-case boundaries to test generation time factor $\alpha_{DFT\_test\_gen}$ (a), test execution time factor $\alpha_{DFT\_t\_time}$ (b), tester capital cost factor $\alpha_{DFT\_capital}$ (c), and area-overhead factor $\alpha_{DFT\_area}$ (d). The shaded areas represent the portions of the space that are no longer uncertain if the relevant alpha factor is moved from its original extreme value (curves 1a and 2a) to its opposite extreme value (curves 1b and 2b).

**Table 2.** Parameter values used in computations of the results shown in Figures 5 and 6. The alpha factors that we changed with respect to the reference case (Figure 5) are shown in boldface.

<table>
<thead>
<tr>
<th>Figure number</th>
<th>Curve</th>
<th>$\alpha_{DFT_test_gen}$</th>
<th>$\alpha_{DFT_t_time}$</th>
<th>$\alpha_{DFT_capital}$</th>
<th>$\alpha_{DFT_area}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>1a</td>
<td>0.05</td>
<td>0.05</td>
<td>0.05</td>
<td>0.05</td>
</tr>
<tr>
<td></td>
<td>2a</td>
<td>0.50</td>
<td>0.50</td>
<td>0.50</td>
<td>0.13</td>
</tr>
<tr>
<td>6a</td>
<td>1b</td>
<td>0.50</td>
<td>0.05</td>
<td>0.05</td>
<td>0.05</td>
</tr>
<tr>
<td></td>
<td>2b</td>
<td>0.05</td>
<td>0.50</td>
<td>0.50</td>
<td>0.13</td>
</tr>
<tr>
<td>6b</td>
<td>1b</td>
<td>0.05</td>
<td>0.50</td>
<td>0.05</td>
<td>0.05</td>
</tr>
<tr>
<td></td>
<td>2b</td>
<td>0.50</td>
<td>0.05</td>
<td>0.50</td>
<td>0.13</td>
</tr>
<tr>
<td>6c</td>
<td>1b</td>
<td>0.05</td>
<td>0.05</td>
<td>0.50</td>
<td>0.05</td>
</tr>
<tr>
<td></td>
<td>2b</td>
<td>0.50</td>
<td>0.50</td>
<td>0.05</td>
<td>0.13</td>
</tr>
<tr>
<td>6d</td>
<td>1b</td>
<td>0.05</td>
<td>0.05</td>
<td>0.05</td>
<td>0.13</td>
</tr>
<tr>
<td></td>
<td>2b</td>
<td>0.50</td>
<td>0.50</td>
<td>0.50</td>
<td>0.05</td>
</tr>
</tbody>
</table>
effect is even more pronounced for larger die sizes, for which inherently lower yield (equation 20) combined with high risk factors makes DFT justifiable even for high volumes. Thus, as many researchers have reported, the escape cost definitely should be considered when evaluating DFT’s cost-benefit impact.

Figure 7. Impact of test escape. We assigned worst-case values to the four alpha factors and varied $\alpha_{\text{escape}}$ from 1 to 15. The unshaded area represents the increasing region categorized “Apply DFT” as $\alpha_{\text{escape}}$ increases.

Yield learning

To study how yield learning affects test cost, we again assign worst-case values to the four DFT factors. Figure 8 shows the boundaries between the DFT and no-DFT regions for yield-learning-factor values ranging from 0.1 to 0.6. Even for a low value ($\alpha_{YLF} = 0.2$), the shift in the boundary favoring DFT is considerable. For
greater $\alpha_{42}$ values, DFT has a more dramatic impact on larger dies. Only small dies with an inherently high yield are unaffected by an increased learning factor. Again, the potential benefits of using DFT for yield learning significantly affect the location of the DFT/no-DFT boundary.

Our DFT-application results lead to the following conclusions:

- Researchers could (and should) minimize the uncertainty region by relating the four DFT alpha factors to IC design-and-test environment attributes to account for higher-order dependencies that may exist among these factors.
- Manufacturing volume is a key factor differentiating the no-DFT and DFT regions. Area overhead, high escape cost, and impact on yield-learning rate also strongly influence the DFT decision.
- Other factors such as test execution time, test generation cost, and tester capital cost seem to be less important, except when die size and volume are small.

The CMU Test Cost Model, like many other published models, rests on several important assumptions. The most arbitrary assumptions concern test generation cost and test execution time. We modeled both factors as functions of die area. In reality, die area may not significantly affect these attributes, because large, microprocessor-like designs may be I/O limited rather than transistor limited. However, design density is an important contributor to both design cost and manufacturing yield.  

Furthermore, to increase the validity of our results, we assumed a wide range of model parameter values to cover most real-life scenarios. Again, in reality, a situation may be quite different, especially with rapidly changing design styles and process technology.

We have shown that test tradeoffs depend on a test methodology’s effect on yield learning and test quality. Clearly, we need further research on the underlying principles that affect both factors. Yield learning is a complex process requiring understanding of how contamination and process variations lead to failures and how such failures can be diagnosed and corrected. In fact, a complete understanding of this relationship would provide a mechanism for modeling test escape, which would ultimately lead to a clear understanding of test quality. Thus, new research must focus on the time-dependent relationships among test, design, and manufacturing.

The advent of SoCs presents some new challenges in modeling test cost. Because SoCs contain different cores implemented in various design styles, they require a combination of different test strategies. One way to deal with this variety is to partition the SoC into components and apply the model to each component, using different model parameters. For example, embedded memories would have different yield and area equations than the random-logic parts. Another important area of research is the impact of time to market on the benefits of DFT.

Acknowledgments

We thank Phil Fisher and Sumit DasGupta for their discussion and support, and all the other members of the Sematech community who provided valuable feedback. We also thank the reviewers and editors for their suggestions and constructive criticism.

References


**Pranab K. Nag** is a project scientist in the Department of Electrical and Computer Engineering at Carnegie Mellon University, where he is a member of the Center for Silicon System Implementation. His research interests include design for manufacturability, yield modeling, economic tradeoff modeling, and information system architectures for IC design, test, and fabrication. Nag has a BTech in electrical engineering from the Indian Institute of Technology, Kharagpur, and an MS and a PhD, both in electrical and computer engineering, from Carnegie Mellon University.

**Sichao Wei** is a software engineer. His research interests include test and cost modeling. Wei has an MS in electrical and computer engineering from Carnegie Mellon University, where he participated in the work described in this article.

**R.D. (Shawn) Blanton** is an associate professor in the Department of Electrical and Computer Engineering at Carnegie Mellon University, where he is a member of the Center for Silicon Systems Implementation. His research interests include CAD of VLSI circuits and systems, fault-tolerant computing and diagnosis; verification and testing, and computer architecture. Blanton has a BS in engineering from Calvin College, Grand Rapids, Michigan; an MS in electrical engineering from the University of Arizona; and a PhD in computer science and engineering from the University of Michigan, Ann Arbor. He is a member of the IEEE and the ACM.

**Anne Gattiker** is a research staff member at IBM’s Austin Research Laboratory. Her research responsibilities include IC test, reliability, diagnosis and design-for-manufacturability. Gattiker has a PhD from Carnegie Mellon University, where she was a National Science Foundation Fellow. She is a member of the IEEE.

**Wojciech Maly** is the Whitaker Professor of Electrical and Computer Engineering at Carnegie Mellon University. His research interests include the interfaces between VLSI design, testing, and manufacturing. Maly has an MS in electronic engineering from the Technical University of Warsaw, and a PhD from the Institute of Applied Cybernetics, Polish Academy of Sciences, Warsaw.

Direct questions and comments about this article to Shawn Blanton, 2109 Hamerschlag Hall, 5000 Forbes Ave., Electrical and Computer Engineering Dept., Carnegie Mellon Univ., Pittsburgh, PA 15213; blanton@ece.cmu.edu.

For further information on this or any other computing topic, visit our Digital Library at http://computer.org/publications/dlib.