

Hold Time Validation on Silicon and the Relevance of Hazards in Timing Analysis

Amitava Majumdar[†]

majumdara@stratosol.com

Wei-Yu Chen[‡]

weiyu.chen@sfbay.sun.com

Jun Guo[‡]

jun.guo@sfbay.sun.com

[†] Stratosphere Solutions, Inc., 830 Stewart Dr., Sunnyvale, CA 94085

[‡] Sun Microsystems, Inc., 410 N. Mary Ave., MS: USUN02-203, Sunnyvale, CA 94085

ABSTRACT

In this paper we motivate the explicit validation of hold-time violations in silicon and propose a method for doing so. New hold-time failure model and test pattern generation methodologies are defined. We outline conditions under which these tests can be applied reliably. We present results of applying these test patterns on a microprocessor and discuss the implications of intermittent failures on the relevance of hazards during timing analysis.

Categories and Subject Descriptors

B.8.1 [Integrated Circuits Performance and Reliability] *Reliability, Testing, and Fault-Tolerance.*

General Terms: Design, Performance, Reliability, Measurement.

Keywords

Hold time validation, Delay test, ATPG, Timing analysis

1. INTRODUCTION

Timing parameters such as max time (aka critical path setup time) and min time (aka hold time) are fine-tuned throughout the design process, often to the very final days. While violating setup time reduces the highest frequency at which a chip can run, violating hold-time (HT) can cause catastrophic failures. Hence, satisfying hold time requirements is extremely important [8][10], in some cases more so than satisfying setup time requirements. This is certainly true during first silicon bring-up and debug when validating functionality is the main goal (clock frequency to a lesser extent). And for designs that allow speed-binning, this is true even during production.

1.1. Sources of Hold Time Failures

The reason for a hold-time (HT) violation is a race condition between a data and a clock signal. When this race condition is aggravated by the presence of a larger than expected skew (delay) on the clock or a lower than expected delay on the data signal path, we have a HT violation. While some clock skew is inevitable, it's usually kept under tight control using timing analysis and timing correction methodologies.

However, hold-times are seldom explicitly checked in silicon. This is partly because process variations have, thus far, been well accounted for, during design. But that has changed significantly with 0.09 μ and later technology nodes [7][9][11][13][14][15][16][18], due mainly to the increased relevance of process variations. Furthermore, HT margins are non-monotonic functions of operating parameters and are difficult to model or check in design environments. Process variations can

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

DAC 2006, July 24–28, 2006, San Francisco, California, USA.

Copyright 2006 ACM 1-59593-381-6/06/0007...\$5.00.

1. Process variations can result in larger than tolerable clock skews in spite of an otherwise well designed clock distribution network. It can also cause devices to be faster than they were modeled to be.
2. Differential IR drop in power grids can cause a marginal race condition to manifest a real HT violation in silicon. IR drop (especially that caused by process variations) is usually not considered during timing analysis.
3. Inductive effects can cause clock skews [17]. Since inductance is not modeled in most design methodologies today, problems due to clock skews can go undetected during design.

1.2. Examples of Problems Caused by Clock Skew

Consider the circuit shown in Figure 1. All FFs are clocked by CLK. Assume that there is an extra relative delay δ on the branch *Clk-1* that feeds FF_2 and FF_3 . Clock skew can give rise to one or more of the following problems.

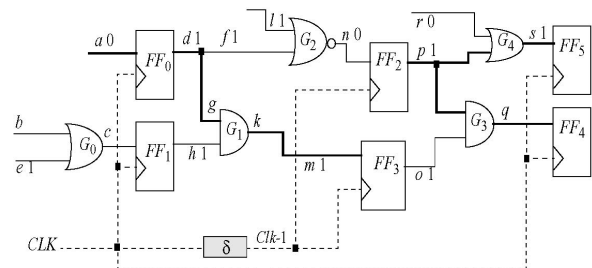


Figure 1: Example circuit showing the effects of skew on a clock branch. Initial values are shown against signal names.

Example 1: Skew causing a HT failure. Consider the path $\{d, g, k, m\}$ from FF_0 to FF_3 . Assume that initially $d, e, h = 1$ (as shown in the figure) and $a = 0$. With a positive edge on CLK, d undergoes a $1 \rightarrow 0$ transition that propagates to m . If δ is larger than the “delay” on path $\{d, g, k, m\}$, this transition reaches FF_3 before the clock pulse reaches it. Hence, instead of capturing a 1, FF_3 captures an erroneous state (a stable 0 or a metastable state that converges to 0).

Example 2: Setup-time (or path delay) failure. Consider the path $\{p, s\}$ from FF_2 to FF_5 . Initially $p = 1$. When a positive edge of CLK arrives at FF_2 , it captures a 0, propagating a $1 \rightarrow 0$ transition on $\{p, s\}$. In the presence of the extra skew on *Clk-1*, this transition starts δ units of time later than expected. If the delay on $\{p, s\}$ is greater than $T - \delta$ (T being the clock period) FF_5 captures a 1 instead of a 0, indicating a setup time failure caused by clock skew (instead of by the more traditional extra delay on a signal path).

Example 3: Skew invalidates tests for setup-time failures. In the absence of a HT failure, the conditions described in Example 2 is a “test” for a delay failure on $\{p, q\}$. Now consider a delay failure on $\{p, q\}$ caused by an extra physical delay on p . In the presence of a HT failure on path $\{d, g, k, m\}$ (from FF_0 to FF_3), signal o is ‘0’ after the first positive edge on CLK which, in turn, causes q to undergo a $1 \rightarrow 0$ transition, whether or not the same transition on $\{p, q\}$ reaches q in time or not. If the delay on path $\{o, q\}$ (from FF_3 to FF_4) is less

than $T-\delta$, FF_4 captures the correct value in the next clock cycle in spite of a delay failure existing on p . In other words, HT failure of $\{d, g, k, m\}$ masks the setup-time failure on $\{p, q\}$.

These examples, along with the fact that (a) HT failures are hard to detect using random or other AC test patterns, (b) HT closure in design is a human intensive and error-prone process today and (c) the presence of process variations can render deterministic HT analyses much less useful, all serve to highlight the need for explicitly testing for HT conditions in silicon.

1.3. Past Work

From the description above it is clear that hold time failures are a class of delay failures that are distinct in many ways from the traditional path delay faults studied extensively in the past [1][2][3][4][5][12][19]. HT failures were formally introduced in [12] where the authors present failure models and conditions for robust and non-robust tests.

1.4. Contribution

Our contribution is three-fold. First we extend prior results on HT failures by defining a fault model (Section 3) and presenting results to prune the failure set (Section 3.1.1). Second, we define test generation and application methodologies (Sections 4, 5, 6). Third, in Sections 7 and 8, we offer insights into the implications of our experiments into timing analysis. Finally we present some important conclusions in Section 9.

2. PRELIMINARIES

Definition 1 The source node of a path P is denoted as S_P . The destination node of P is denoted as D_P .

Definition 2 Let G_p denote the set of all gates in path P and E_p denote the set of all inputs to gates in G_p . The off-path input s_g of a gate $g \in G_p$ is an input signal to g such that $s_g \in E_p$. The set of off-path inputs to all gates in P is denoted by O_p .

Definition 3 A path P is enabled either (a) if all off-path inputs have enabling values or (b) if one or more off-path inputs to a gate g (in P) have controlling values then so does the corresponding on-path input to g .

Definition 4 A transition at the input of a gate g is called an enabling transition if the next value of the transition is an enabling value of g . Similarly a transition whose next value is a controlling value of g is called a controlling transition for g .

3. HOLD TIME (HT) FAILURE MODEL

The conditions necessary for a HT failure to manifest at a flip-flop F with hold time requirement h are as follows. There exists a path P with source $S_P = f$ and destination $D_P = F$ such that (a) the capture edge of a clock arrives at f at time T_f and triggers a transition at its output, (b) the same capture edge arrives at the clock port of F at time T_F and (c) the transition at the output of f propagates along P , and arrives at the input of F at time $t \leq T_F + h$.

Note that $t = T_f + \delta_p$, where δ_p is the delay of path P (assuming δ_p includes clock-to-q delay of flop f). Rewriting the above condition for a hold time violation we get $T_F - T_f \geq \delta_p - h$. Here $T_F - T_f$ is the skew on the clock. Note that as HT (requirement) h or skew $T_F - T_f$ increases, so does the likelihood of a HT violation at F . The likelihood decreases with an increase in the delay δ_p .

If we were to consider all of the above parameters in the failure model, the number of failures in the model would explode and the model would (for all practical purposes) be unusable. The one parameter that remains invariant between a representation of the design and its embodiment in silicon is the set of paths between FFs. Furthermore the set of paths is discrete and countable and hence is a

good candidate for describing and enumerating the set of HT failures in a design. The drawback of such a set lies in the fact that this description is incomplete from a HT standpoint. However, for test generation and failure simulation, the set of paths is sufficient in that all possible HT failures are considered by sampling paths from this set.

3.1. A General Hold-Time (HT) Fault Model

A storage graph of a circuit is a directed graph $G = (V, E)$ where each $v \in V$ represents a unique flip-flop and an edge $e \in E$ indicates a combinational signal path between FFs. The storage graph for the circuit of Figure 1 is shown in Figure 2.

The hold-time fault model for a circuit is a set of faults, each of whose elements corresponds to exactly one edge $e = (u, v) \in E$ in the storage graph of the circuit. Furthermore each edge in E has a corresponding element in the fault model.

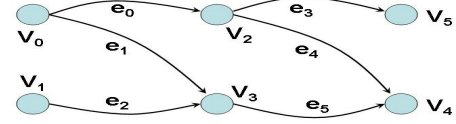


Figure 2: Storage graph for circuit of Figure 1.

3.1.1 Cyclic Paths

We can further prune the set of HT faults by removing a subset of cyclic paths, i.e. combinational paths that start and end at the same FF. In a storage-graph representation, such a path would correspond to an edge starting and ending at the same node.

Lemma 1: If a non-inverting cyclic path P has a transition at its input then there exists a gate in P , such that in the clock-cycle before the transition is launched, at least one of its off-path inputs has a controlling value and its on-path input has a non-controlling value.

Theorem 1: Non-inverting cyclic paths cannot manifest hold-time violations.

Theorem 1 implies that non-inverting cyclic paths can be eliminated from a HT failure model. The same cannot be said for inverting cyclic paths. The authors readily concede, however, that cyclic paths in general are highly unlikely to manifest HT failures.

3.2. A Target Hold-Time Failure Model

The population of faults described by the general model of Section 3.1 runs into the 10's of millions for most designs today, despite the elimination of cyclic paths offered by Theorem 1. Generating a test for each HT failure would be well nigh impossible. Hence, it is important to prune the set further to a small target subset. One such target model for HT faults is extracted from the results of static HT analysis (Figure 3) whose basis is (a) estimated delays of paths and (b) estimated clock skew between source and destination FFs.

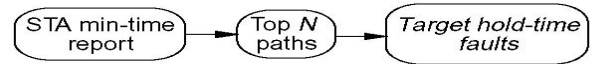


Figure 3: Simple flow to extract target HT failures from results of min-time Static Timing Analysis.

Target Model Size: N in the above procedure should be large enough to compensate for the inadequacies of models used in static timing analysis (STA) and parasitics extraction. Determining N is thus a statistical sampling problem, where the universe of paths is partially ordered through STA. In this paper we use a constant N and propose the sampling problem as part of future work.

False Paths: An important step not shown in Figure 3 is the removal of false paths before they are tested on silicon. Unfortunately, there are no efficient techniques (or commercial tools) to identify false paths. This is currently done manually by designers who understand the functionality of their design.

Our method provides an additional screen for HT paths. Paths need to be analyzed only after they show hold time marginality on silicon. This does not preclude analyzing and fixing paths in design. Instead, our method provides a way to complement STA (done during design) in ferreting out HT violations in silicon.

4. DETECTING A HOLD TIME FAILURE

In this section we expand on the conditions derived in [12] for propagating a HT transition through a gate. We also focus on some of the practical implications of failures/passes on testers.

Sensitizing a HT failure on path P involves launching a transition at its source S_P and setting up conditions along the path such that the transition propagates to the destination D_P . In this sense, the conditions for HT failure detection are not different from those required for setup-time failures. The off-path sequences for propagating a “hold-time” transition through a gate in P are illustrated using an AND gate as shown in Figure 4. Similar conditions can be derived for other gate types. We classify the sequences as (a) robust, (b) non-robust and (c) weak tests.

4.1. Propagating Enabling Transitions: Figure 4 (a)

A rising edge at the on-path input of the gate needs to be propagated to its output.

4.1.1 Case 2: Enabled Steady Value on Off-Path Inputs

This corresponds to Case 2 in the figure and is clearly a *robust sequence* at the gate’s off-path input. A failure [pass] of such a sequence on the tester unambiguously indicates the presence [absence] of a HT violation.

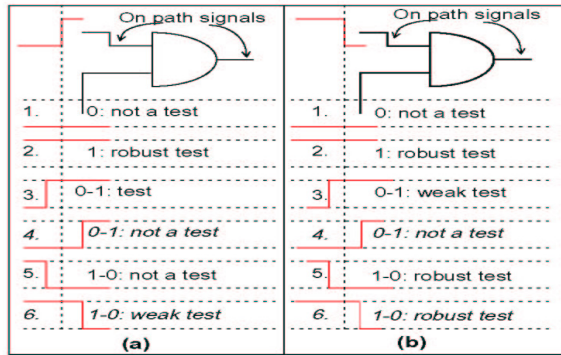


Figure 4: HT test conditions for (a) rising and (b) falling edges at on-path input of an AND gate.

4.1.2 Cases 3 & 4: Enabling Transition on Off-Path Inputs

When the off-path input undergoes a rising transition, the output of the AND gate undergoes a hazard-free rising transition at a time that is determined by the later of the input transitions.

1. *Case 3 is a test*: If the off-path input changes earlier than the on-path signal, then it results in a perfectly robust test for P .
2. *Case 4 does not test P* : If P is selected from a target fault set (Section 3.2) it is likely to be a short path. This means that the off-path input is likely to change after the on-path input (Case 4) and therefore it is not a test for P .
3. *Case 4 is a robust test for off-path input*: For the path Q on which the off-path input sits, Case 4 is a robust test in the same way as Case 3 is a robust test for path P .

Interpreting a tester failure: Due to the above reasons, while such a vector sequence is a non-robust test, if applying the test results in a mismatch, it is still considered a valid test. The reason is that whether it is Case 3 or Case 4, a failure indicates a HT violation on P as well as one on path Q containing the off-path input. Hence, despite the non-robustness of the test, it is still desirable to include it in the test suite.

Interpreting a tester pass: Unfortunately, if a test passes, we can neither confirm nor deny the existence of a HT violation. This is because, Case 4 invalidates the test and we do not know whether Case 3 or 4 has occurred. Hence a passing test is inconclusive.

4.1.3 Cases 5 & 6: Controlling Transition on Off-Path Inputs

It is clear that Case 5 cannot be a test. Case 6 results in a static-0 hazard at the gate’s output. However, it can still be considered a test for P since the leading edge of the static hazard is caused by the rising transition at the on-path input. If there is a HT violation with a magnitude less than the pulse-width of the static-0 hazard, it can be detected by such a test. Due to this non-determinism, a falling edge on an off-path input results in a *weak sequence*.

Interpreting a tester failure: If applying such a test sequence results in a failure then only Case 6 could have occurred and the resulting HT violation has a magnitude less than the pulse-width of the static-0 hazard at the gate’s output. Thus we can conclude that it is a valid test. Our experimental results (Section 8) show that such tests are indeed useful.

Interpreting a tester pass: Since we cannot predict whether Case 5 or 6 occurs and since Case 5 invalidates the test, a tester pass is inconclusive.

4.2. Propagating Controlling Transitions: Figure 4(b)

A falling edge at the on-path input of the AND gate needs to be propagated to its output.

4.2.1 Case 2: Enabled Steady Value on Off-Path Inputs

Again, Case 2 is a *robust sequence* and need not be elaborated further here. A failure [pass] on the tester unambiguously indicates the presence [absence] of a HT failure on P .

4.2.2 Cases 3 & 4: Enabling Transition on Off-Path Inputs

It is clear that Case 4 is not a test since the output of the AND gate is a steady 0. Case 3 results in a static-0 hazard and is thus a weak test. Detection of a HT violation depends on whether or not the magnitude of the violation is less or greater than the width of the hazard pulse. This introduces non-determinism in the detection process and hence its classification as a *weak sequence*.

Interpreting tester results: As in the case of Section 4.1.2, a mismatch implies that there is a HT violation and a pass is inconclusive.

4.2.3 Cases 5 & 6: Controlling Transition on Off-Path Inputs

These cases are classified as robust sequences. The reasoning behind this is as follows. Both these cases result in a hazard-free transition at the gate’s output. In the event of a mismatch on a tester, if indeed Case 5 had occurred in the design then it is a detection of a HT violation on a path containing the off-path input. And if Case 6 had occurred then it is a detection of a HT violation on P itself. Since in both the cases there is a deterministic detection of some HT violation, we classify these sequences as robust sequences.

Interpreting a tester failure: A failure of such a test clearly indicates the existence of a HT failure, either on P or on a path containing the off-path input.

Interpreting a tester pass: If such a test passes on the tester, then it clearly indicates that neither P nor the path containing the off-path input have an HT violation.

Note: The “clean”-ness of the interpretations above is similar to those derived for tests where the off-path input is a constant enabling value (Case 2 in Figure 4). This lends further credence to the assertion that these tests are indeed robust.

5. TEST PATTERN GENERATION

We now use the results of Section 4 to define a procedure for generating test patterns for a target path $P = (G, E)$.

Procedure 1 Hold-Time Test

1. Initialize V ($V \in \{0,1\}$) at the output of S_P ;
2. Justify \bar{V} at the data-input of S_P ;
3. For each off-path input in O_P
 - 3.1. justify **Robust Sequence**;
 - 3.2. else justify **Non Robust Sequence**
 - 3.3. else justify **Weak Sequence**
 - 3.4. else report **Untestable Path P**
4. Apply One Capture Clock

In the above procedure, a pattern is robust *if and only if* a robust sequence is found $\forall f \in O_P$. A non-robust pattern is generated *only if* a robust pattern cannot be found and a weak pattern is generated *only if* a robust or a non-robust pattern cannot be found.

5.1. Notes on Generated Patterns

1. Each test pattern spans 2 clock cycles but requires a single “functional” clock. The single clock functions as the “transition launch” clock at the source FF as well as the “capture” clock at the corresponding destination FF of the path under test¹.
2. A qualitative observation is that the probability of fortuitously detecting HT violations using stuck-fault DC tests is low. This is because HT tests (especially robust ones) require several inputs to be constrained simultaneously across 2 clock cycles.
3. Generating robust tests for HT failures in a target failure model (of Section 3.2) is “easier” than that for setup-time failures. This is because HT paths are likely to be shorter than setup-time paths and thus require fewer signals to be constrained. Hence, from an ATPG complexity standpoint, HT failures lie somewhere in between transition faults and setup-time faults.

5.2. Hold-Time Patterns Using Delay Test Patterns

The process of justifying off-path inputs across 2 cycles (Procedure 1) is no different than the one used to generate test patterns for setup-time failures (path delay faults in test parlance). Based on this observation we define a new test generation procedure that uses tests for setup-time failures (aka path-delay or PD) failures to generate test patterns for HT failures.

Procedure 2 PD to Hold-Time Transform

1. For each HT path P
 - 1.1. Generate PD pattern T with 2 clock cycles.
 - 1.2. Remove 1 clock from launch-capture sequence;
//Step 1.2 makes it a HT test pattern for P
2. If T is **Robust** or **Non-Robust**
 - 2.1. Flip expected value at D_P
3. If T is **Weak** leave expected value as is.

The expected value at D_P is flipped in order to reflect the fact that in a HT-clean circuit, the transition at S_P should not propagate to D_P . This is in contrast to PD tests where the transition in a setup-time clean circuit should propagate to D_P . Step 3, in contrast, reflects the fact that for a weak test, a hazard is introduced and therefore the PD transition does not propagate to D_P . Thus, the expected value at D_P need not be flipped.

Procedure 2 offers an economical alternative to implementing Procedure 1 from the ground up. It allows existing path-delay ATPG solutions to be reused for generating HT patterns. We implemented Procedure 2 using the PD capability of a commercial ATPG tool and present results from experiments on actual silicon.

¹ Since launch and capture happen with the same clock cycle, it is clear that hold-time paths cannot be tested using “Launch on Last Shift” (LOLS) method [6].

5.3. Classification of Tests by ATPG Tool

The commercial ATPG tool that we used, classified test patterns slightly differently than the norm: A PD pattern is considered **(a) robust** if it holds off-path inputs at constant enabled values throughout the 2 cycles., **(b) non-robust** if off-path inputs undergo enabling transitions and **(c) weak** if off-path inputs undergo controlling transitions, potentially causing hazards.

Based on these (somewhat stricter) definitions it is easily proven that a robust PD test is also a robust test for a HT failure on that path. But both non-robust and weak PD tests could generate hazards and thus translate to non-robust, weak or non-tests for the corresponding HT failure. Hence we will classify all non-robust/weak PD tests generated by Procedure 2 as weak HT tests.

6. TEST APPLICATION USING SCAN

Given that Procedures 1 and 2 generate 2-cycle functionally justified test patterns, they are naturally suited for application through scan. Scan-based application simply involves, shifting in the complete pattern and applying *exactly one capture clock*. While test patterns for HT failures require only one “functional” clock (and hence look like stuck-at tests), they are essentially AC tests and are subject to a host of electrical issues that are commonly ignored in most studies dealing with AC tests. The rest of this section addresses various issues that arise from scan-based application of HT AC tests.

6.1. Electrical & Thermal Considerations

6.1.1 $L(di/dt)$ Dampening of Power and Ground

In most ICs (certainly so in microprocessors) scan is done at much lower frequencies than the ICs’ functional frequency. Hence, the typical current profile on a power grid can be close to a sawtooth waveform, as shown in Figure 5. During scan, each positive edge on the clock (waveform at the bottom of Figure 5) generates a lot of switching activity, causing the current in the powergrid to (slowly) rise to a peak. The power-grid’s L acts as a damper.

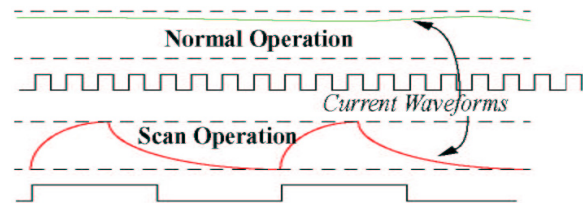


Figure 5: Current waveform in power-grid during scan operation vs that during normal operation.

Once the devices have switched, the current finally subsides (usually to leakage-only) before the next clock arrives, possibly causing the path to switch slower than it normally would. This invalidates the test and gives false negative errors. On the flip side, the same phenomenon could cause the clock-grid to slow down more than the signal path. This can result in false positive errors causing an otherwise good chip to fail a test. We consider this latter case to be *far less likely*, given the care taken in designing clock networks. Variations in current profile on the power-grid are functions of scan frequency, number and placement of decoupling capacitances and the cumulative leakage current.

6.1.2 Sensitivity to Supply Voltage (IR-Drop) and Temperature

Delays on clock and signal paths depend on device and wire delays, both of which are highly sensitive to temperature and voltage. A HT violation is the result of a race between a signal path and a clock path, two asynchronous delays. By its very nature, HT is chaotic and often defies prediction. The commonly held notion about HT violations is that increasing supply voltage and decreasing temperature exacerbates HT marginalities. This may be true for

gross HT violations. However, as shown through our experiments, a truly marginal min-time path could behave unpredictably, as is expected of chaotic phenomena.

6.1.3 Sensitivity to Clock Frequency

HT marginality is widely considered to be frequency-independent. It is indeed so, when studied as purely a logic and a temporal phenomenon. However, in the presence of an electrical environment, clock frequency comes into play in the form of power and ground noise (Figure 5). As scan frequency increases, or the power grid is better designed for low-frequency operation (through placement of decoupling capacitances etc.) the waveform smoothens out and the possibility of test invalidation reduces.

6.1.4 Summary of Test Application Issues

The quality of scan-based HT test depends on how well the actual electrical and thermal environment can be mimicked within a test setup, especially when different parameters influence each other. In the absence of a good facsimile environment, the tests run the risk of being invalidated. Based on some of the above reasoning we can conclude the following.

- It is generally difficult to detect HT violations. This is so because of the high likelihood of false negative errors.
- When a min-time path fails a scan-based min-time test on silicon, it is very likely to be a true HT marginality.
- A passing min-time path may or may not be violating HT and the results are inconclusive.

In summary therefore, when applying scan-based HT tests, failures are usually true failures, whereas passing tests are inconclusive.

6.2. Mitigating Electrical/Thermal Sensitivities

Some of the steps we have taken to mitigate the effect of electrical and thermal effects are as follows:

1. **Increased scan frequency:** In order to reduce the damping effects of the power-grid, we run the HT tests with the highest scan frequency. Before applying the tests, we calibrated each silicon unit for the highest scan frequency at which they pass all scan-shift and stuck-at tests.
2. **Reduce dead cycles after changes in scan-enable signal:** For mux-scan designs, the scan-enable (SE) signal is usually a slow signal and it is customary to allow enough time or “tester dead cycles” for a change in SE to reach all scan FFs. However, too many dead cycles can also invalidate tests due to di/dt effects (Section 6.1.1). In order to reduce this effect, we calibrated all silicon units for the minimum number of “dead cycles” for which they pass all scan-shift and stuck-at failure tests.
3. **Shmoo across temperatures and supply voltages:** In order to determine a good “test box” we shmooed all the units across several tester-controlled parameters: temperatures, supply voltages, clock frequencies as well as “dead cycles”.

7. EXPERIMENTAL SETUP

7.1. Test Case

We conducted experiments with a dual-core CMP microprocessor with ~162K FFs, an aggressive memory hierarchy of 2MB shared, on-chip level-2 cache and a shared, external level-3 tag. The design was targeted for a sub-100nm technology node. We selected 5 “known-good” units that passed all scan-based and functional tests.

7.2. Test Generation

The min-time static timing analysis report contained more than 1 million paths, all of which were deemed to be free of HT violation. From the report we selected the top $N = 20K$ paths. As part of Procedure 2, the 20K paths were translated and read by a commercial ATPG tool. The tool further pruned the set down to

8,206 paths between FFs, rejecting paths that were either to or from memory arrays and hence not accessible through scan. Thus 16,412 distinct path-delay faults, one for each transition on each path, were defined. The results of test pattern generation using Procedure 2 are shown in Table 1.

Path Delay Faults	Hold-Time Faults	Number
Robust	Robust	13,400
Non-robust / weak	Weak	1623
Untestable	Untestable	1,389
Total		16,412
Test coverage		91.54%
Test efficiency		97.84%
Number of patterns		1558

Table 1 Results of test pattern generation using Procedure 2.

Note that we did not have visibility into the ATPG tool to determine whether a weak test introduced a hazard or not. Hence the expected value of D_P was flipped for all tests (whether robust, non-robust or weak). *This meant that weak tests were expected to fail on silicon and this was verified in our experiments.*

Further, in keeping with our observations about the ease of test generation for HT failures (see Section 5.1), we find that a test coverage of >91% well exceeds the kind of coverage we normally see for setup-time (long-paths) failures.

7.3. Test Application: Shmoo Test

HT depends on the relative delays of two different paths and is thus a non-monotonic function of scan-frequency (F), supply voltage (V) and temperature (T). Due to this reason, the traditional method of testing an IC at the corners (or extremes) of a “test box” is not enough to detect all HT marginalities. Validating HT on silicon therefore, requires a systematic search of the 3-D FVT space.

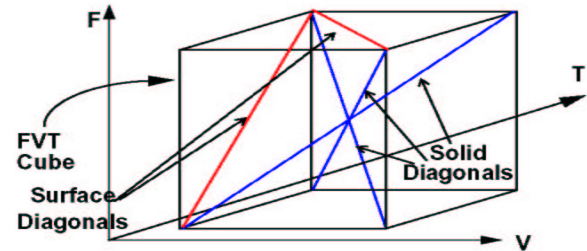


Figure 6: Sample FVT cube and the points of interest for HT validation.

We recommend that testing be done at different points along the 3 solid diagonals (Figure 6) and along as many of the 12 surface diagonals as possible. *We call this methodology shmoo test, based on the well known shmoo methodology.* Based on the above reasoning, the experiments performed with the 5 units are as follows:

1. FVT was sampled across a range at 60 points in the cube (a) *Scan frequency:* 25MHz to 125 MHz in steps of 25MHz, (b) *Temperature:* 90°C, 105°C, 120°C (c) *Supply voltage:* 0.9V, 1.0V, 1.1V, 1.2V.
2. We also ran the test patterns repetitively (10 times) on each unit at the same FVT coordinate of (50MHz, 1.1V, 105C) in order to check for timing marginalities that are independent of FVT.

8. EXPERIMENTAL RESULTS AND ANALYSIS

The goal of our HT silicon validation experiments was to supplement min-time static timing analysis. Towards this end we succeeded, in that all the “trustworthy” robust tests passed (see Table 2 below) and a majority of the weak tests failed as expected under all test conditions. Thus the design was deemed to be free of HT marginalities per min-time static timing analysis.

	# faults	% pass	% fail	% intermittent
Robust	13,400	100	0	0
Non-robust / weak	1,623	60.3	39.6	0.1

Table 2 Results of tests aggregated over 5 silicon units and shmoo space covered in the FVT cube.

While a majority of the non-robust tests passed, some of them failed, again under all test conditions. Due to this reason, the non-robust tests did not offer any conclusive evidence for or against the presence of HT violations on the paths they tested.

However, we did note an interesting phenomenon. We found that some (~0.1%) paths were intermittently passing and failing the same test patterns, even for the repetitive tests (where all parameters were kept constant). As with most HT test failures, these were indeed true failures (see Section 4.1.3).

8.1. Hazards Cause Intermittent Failures

We express the conclusion from the intermittent failures in the form of a theorem below.

Theorem 2: *A path P fails a hold-time test pattern T intermittently only if T introduces hazards in the path.*

Corollary (to Theorem 2): *A path P fails a test pattern T intermittently only if T is a weak test.*

The main implication of Theorem 2 is that the paths that fail weak tests intermittently do have a HT marginality that was sensitized by a hazard. Note that the ATPG tool generated a hazard-based test only because it could not find a hazard-free test. *Therefore, a hazard was (in some sense) required to detect the HT marginality on those paths.* Unfortunately, due to the complexity of considering hazards, static timing analysis (STA) tools today do not include hazards as part of its “timing simulation”. Hence such HT marginalities could not have been pointed out using conventional timing methodologies.

9. CONCLUSION

We present a methodology for validating HT in ICs and draw several important conclusions from the above results.

1. We can re-use traditional path-delay test generation techniques, with minor post-processing of patterns (see Procedure 2), to generate test patterns for detecting HT failures on silicon.
2. It is feasible to validate HT characteristics of an IC using scan-based application of such test patterns.
3. Due to the non-monotonic nature of HT, it is important to implement a *shmoo-test methodology* (Section 7.3) to detect HT violations.
4. *Hazards should be considered as part of HT analysis.*
5. In the absence of hazards in STA, the only way to check for all HT marginalities is to validate them in silicon.
6. Weak tests (often discarded as unreliable) play an important role in validating HT in silicon, especially in their ability to introduce hazards. Robust tests cannot introduce hazards.

We identify the following areas of future work: (a) implement a HT fault simulation capability, (b) simulate stuck-at production tests for HT coverage, before generating new HT patterns and (c) define a statistical sampling method to determine the size N of the target HT failure model.

Acknowledgement: The authors would like to thank Liang Chi Chen and Manuel d’Abreu for their suggestions and feedback on delay test generation.

10. References

- [1] Y.K. Malaiya, R. Narayanaswamy, “Testing for Timing Failures in Synchronous Sequential Integrated Circuits,” Proc. ITC, 1983, pp. 560-571.
- [2] G.L. Smith, “Model for Delay Failures Based upon Paths,” Proc. IEEE ITC, 1985, pp. 342-349.
- [3] S. Patil, S.M. Reddy, “A Test Generation System for Path Delay Failures,” Proc. ICCD, 1989 pp. 40-43.
- [4] C.J. Lin, S.M. Reddy, “On Delay Testing in Logic Circuits,” IEEE Trans. on Comp. Aided Design of Int. Circuits and Systems, Sept. 1987, pp. 694-703.
- [5] S.M. Reddy, C.J. Lin, S. Patil, “An Automatic Test Pattern Generator for the Detection of Path Delay Failures,” Proc. ICCAD, 1987, pp. 284-287.
- [6] S. Patil, J. Savir, “Skewed-Load Transition Test: Part 2, Coverage,” Proc. ITC, 1992, pp. 714-722.
- [7] J.L. Neves, E.G. Friedman, “Optimal Clock Skew Scheduling Tolerant to Process Variations,” Proc. of DAC, 1996, pp. 623-628.
- [8] J. G. Xi, D. Staepelaere, “Using Clock Skew as a Tool to Achieve Optimal Timing,” Integrated System Design, April 1999.
- [9] P. Zarkesh-Ha, T. Mule, J.D. Meindl, “Characterization and Modeling of Clock Skew with Process Variations,” Proc. of IEEE CICC, 1999, pp. 441-444.
- [10] D. Harris, M. Horowitz and D. Liu, “Timing Analysis Including Clock Skew,” IEEE Trans. on Comp. Aided Design of Int. Circuits and Systems, Vol.: 18, Nov. 1999, pp. 1608-1618.
- [11] R. Saleh et. al, “Clock Skew Verification in the Presence of IR-Drop in the Power Distribution Network,” IEEE Trans. on Comp. Aided Design of Int. Circuits and Systems, Vol. 19, June 2000, pp. 635-644.
- [12] S.M. Reddy et al., “On Validating Data Hold Times for Flip-Flops in Sequential Circuits,” Proc. of IEEE Int. Test Conf., 2000, pp. 317-325.
- [13] S. Sauter et. al, “Effect of Parameter Variations at Chip and Wafer Level on Clock Skews,” IEEE Trans. on Semi. Manufacturing, Vol. 13, Nov. 2000, pp. 395 - 400.
- [14] V. Mehrotra, D. Boning, “Technology Scaling Impact of Variation on Clock Skew and Interconnect Delay,” Proc. of IEEE Interconnect Technology Conf., 2001, pp. 122-124.
- [15] Xiaohong Jiang, S. Horiguchi, “Statistical Skew Modeling for General Clock Distribution Networks in Presence of Process Variations,” IEEE Trans. on VLSI Systems, Vol. 9, Oct. 2001, pp. 704-717.
- [16] D. Harris, S.Naffziger, “Statistical Clock Skew Modeling with Data Delay Variations,” IEEE Trans. on VLSI Systems, Vol. 9, Dec. 2001, pp. 888 - 898.
- [17] B. Kleveland et. al, “High Frequency Characterization of On-Chip Digital Interconnects,” IEEE J. of Solid-State Circuits, Vol. 37, No. 6, June 2002, pp. 716-725.
- [18] P. Zuchowski, “The Titanic: What Went Wrong in Technology,” Proc. DAC 2005, Paper 21.1, June 2005.
- [19] C. Metra et. al, “The Other Side of the Timing Equation: A Result of Clock Faults,” Proc. of The Int. Symp. on Defect and Fault Tolerance in VLSI Sys., 2005, pp. 169-177.