# Transition-Time-Relation Based Capture-Safety Checking for At-Speed Scan Test Generation

K. Miyase[1,2], X. Wen[1,2], M. Aso[3], H. Furukawa[3], Y. Yamato[4], S. Kajihara[1,2]

1 : Kyushu Institue of Technology, 2 : JST CREST,

3 : Renesas Micro Systems Co. Ltd., 4 : Fukuoka Industry Science & Technology Foundation

{k_miyase,wen, kajihara}@cse.kyutech.ac.jp, {masao.asou.wm, hiroshi.furukawa.wh}@rms.renesas.com, yamato@lab-ist.jp

***Abstract*—Excessive capture power in at-speed scan testing may cause timing failures, resulting in test-induced yield loss. This has made capture-safety checking mandatory for test vectors. This paper presents a novel metric, called the TTR (Transition-Time-Relation-based) metric, which takes transition time relations into consideration in capture-safety checking. Capture-safety checking with the TTR metric greatly improves the accuracy of test vector sign-off and low-capture-power test generation.**

## I. Introduction

Power reduction, in addition to timing closure and area minimization, is now mandatory for LSI designs. Various techniques, such as clock gating, multi-threshold voltages, power domain portioning, dynamic voltage scaling, etc, have been proposed for reducing ***function power***. With these techniques, designers can achieve a low functional power level. However, low functional power does not mean low ***test power***. In fact, test power can be several times higher than functional power [1] due to high fault / block parallelism and non-functional clocking used during testing for higher test efficiency. Excessive test power may cause severe problems, especially in ***at-speed scan testing***.

Scan testing has two modes, namely *shift* and *capture*. Shift is for loading test stimuli and unloading test responses through scan chains, while capture is for capturing test responses from the circuit-under-test. At-speed scan testing is usually realized by using the *LOC* (*Launch-On-Capture*) [2] clocking scheme which is widely used in industry due to its simple physical implementation.

There are only two capture clock cycles in the LOC scheme, which means that the accumulative impact of capture power is negligible. However, the instantaneous impact of capture power may cause ***capture malfunction*** [3], as described below:

Two capture cycles in the LOC scheme are a ***launch cycle*** and a ***capture cycle***. If excessive switching activity occurs in the launch cycle, excessive IR-drop may occur, leading to excessive path delay and ultimately timing failures in the capture cycle. That is, unexpected test responses may be captured in capture cycle, even though the circuit-under-test is defect-free and functionally operational. Particularly in the testing of high-speed devices, even small delay increase due to excessive IR-drop may cause capture malfunction, resulting in test-induced yield loss [3].

In order to tackle the problem of capture malfunction, it is critical to check whether a test vector may cause excessive switching activity in the launch cycle or not. That is, ***capture-safety checking*** needs to be conducted, either in test vector sign-off or in test generation. Capture-unsafe test vectors need to be discarded or rescued by various low-capture-power techniques based on DFT, ATPG, and test vector modification [3-8,10]. It is clear that the effectiveness and efficiency of test vector sign-off and low-capture-power test generation are determined by the accuracy of capture-safety checking.

Previous capture-safety checking metrics can be classified from spatial and temporal perspectives, as illustrated in Figure 1. From the spatial perspective, capture-safety checking metrics can be classified as (S1) ***global*** (*the switching activity of the whole circuit is checked*) [7,8], (S2) ***regional*** (*the switching activity in specific regions is checked*) [6], (S3) ***structural-long-path-based*** (*the switching activity around structurally long paths is checked*) [9], and (S4) ***sensitized-long-path-based*** (*the switching activity around sensitized long paths is checked*) [10]. From the temporal perspective, capture-safety checking metrics can be classified as (T1) ***total*** (*the switching activity in the whole launch cycle is checked*) [7,8], (T2) ***instantaneous*** (*peak or instantaneous switching activity is checked*) [6], and (T3) ***transition-window-based*** (*switching activity in the transition window is checked*) [11].

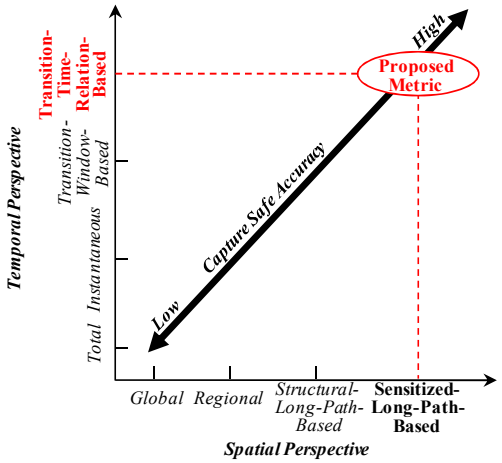


Figure 1. Classification of Capture-Safety Checking Metrics.

Generally, a metric focused on switching activity around sensitized long paths is more accurate than others. The reason is that a sensitized long path is very susceptible to IR-drop-induced delay increase. That is, the delay of an on-path gate *G* increases when its supply voltage drops due to the transitions occurring at its neighboring gates. However, from the temporal perspective, if the transition at a neighboring gate occurs after the transition of *G*, it will not impact the supply voltage of *G*. That is, a late transition will not impact an early transition. Clearly, this transition time relation is important for the accuracy of capture-safety checking. However, none of previous capture-safety

978-3-9810801-7-9/DATE11/©2011 EDAA

checking metrics takes such transition time relations into consideration.

In this paper, we propose a novel capture-safety checking metric based on ***transition time relation*** (***TTR***). The TTR metric focuses on long sensitized paths (i.e., paths whose length exceeding a threshold) and evaluates the delay increase along each long sensitized path based on the transition time relation. As shown in Figure 1, the proposed metric has higher accuracy compared to previous metrics.

In order to take transition time relations into consideration, the proposed metric first identifies sensitized long paths and their neighboring node (logic gate) set where nodes in the set are closely located and share a power supply net. Then, it assesses the delay increase of each sensitized long path using the ***TTR metric***, which is based on those transitions that occur earlier than any transition at each on-path node. Detailed evaluation experiments based the largest ITC'99 circuit demonstrated the accuracy and scalability of the TTR metric for capture safety checking.

## II. TTR-Based Capture-Safety Checking

### A. Proposed Capture-Safety Checking Flow

Figure 2 shows the proposed capture-safety checking flow. The details are described in the subsections *B*, *C*, *D*, and *E*.

① **Power-Network-Based Region Partitioning:** This is a pre-processing based on power supply network design and layout information (DEF). Each region consists of nodes sharing a power supply net.

② **Sensitized Long Path Identification:** All paths that are sensitized and longer than a threshold are identified. Capture-safety checking will be conducted by assessing the delay increase along sensitized long paths.

③ **Impact Node Set Identification:** The set of impact nodes that have a significant impact on the IR-drop on an on-path node of a sensitized long path is indentified for each on-path node.

④ **TTR Metric Calculation:** A ***TTR value*** is calculated for a test vector, which is based on the impact node set of each on-path node of each sensitized long path. This *TTR* value is used to check the capture-safety of the test vector.

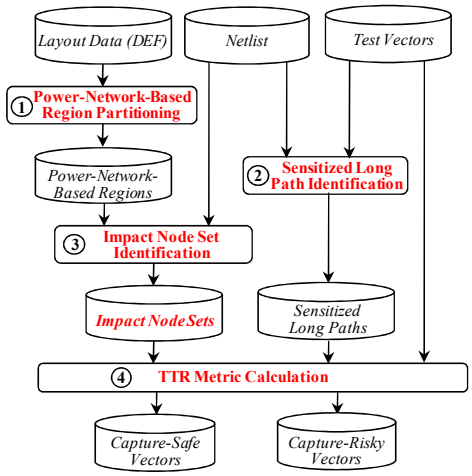


Figure 2. TTR-Based Capture-Safety Checking Flow.

### B. Power-Network-Based Region Partitioning

In order to assess the IR-drop at a node for capture-safety checking, it is necessary to identify all nodes that have significant impact on the node. Generally, if two nodes are located closely to each other and share a power supply net, the transition at one node (*aggressor*) will significantly impact the other node (*victim*) in terms of IR-drop.

Therefore, we partition a circuit into small regions, called ***power-network-based regions***, each consisting of nodes that are closely located and share a power supply net. The optimal region size depends on the power supply network design and node layout. In our experiments, we partition a circuit so that each region consists of approximately 10 nodes since each node shares the same close-by power rail. Note that this partitioning only needs to be conducted once as pre-processing.

Figure 3 shows an example of a power-network-based region *R*, which consists of six nodes. It is clear that if, for example, the IR-drop impact at $n_3$ needs to be assessed, it is only necessary to take the transitions occurring at nodes $n_1$, $n_2$, $n_4$, $n_5$ and $n_6$ into consideration.

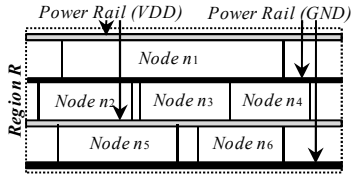


Figure 3. Power-Network-Based Design.

### C. Sensitized Long Path Identification

In order to evaluate the TTR metric, we identify sensitized long paths of a test vector as target paths in capture-safety checking. That is, we check the switching activity around such target paths to determine the capture-safety of the test vector. This is because an unsensitized path or a short path is unlikely to cause capture malfunction even if there is excessive switching activity around the path. Whether a path is long or not is determined by a designer-specified threshold. In our experiments, we used the efficient path extraction technique in [12], which allows us to obtain sensitized paths in descending order of path lengths in relatively short time. In addition, we used 70% of the structurally longest path as the threshold in our experiments.

### D. Impact Node Set Identification

Not the transitions at all nodes physically close to a node have a significant impact on the delay increase of the node. That is, spatial accuracy only is not enough to guarantee accurate capture-safety checking, which also requires temporal accuracy. For this reason, we obtain the ***TTR impact node set*** which takes transition time relations into consideration.

First, ***primary regions*** are identified from power-network-based regions. A primary region is a region that passes through at least one sensitized long path. In addition, an ***off-path-primary node*** is an off-path (*with respect to a sensitized long path*) node that exists in a primary region. An example is shown in Figure 4, where $P_1 \sim P_3$ are

sensitized long paths and $PR_1 \sim PR_4$ are primary regions among power-network-based regions $R_1 \sim R_6$.

Next, in order to take transition time relations into consideration, the ***impact node set*** for an on-path node $v$, denoted by $INS(v)$, is defined as the set of all nodes whose transitions occur earlier than the transition on the on-path node. There are two approaches to determining transition time relations, i.e., a static approach that compares distances from flip-flops (sources) to nodes (destinations) and a dynamic approach that uses timing-accurate logic simulation. Generally, the static approach yields accurate enough results within reasonable computation time.

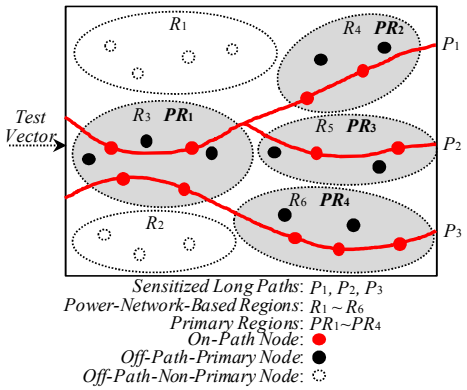


Figure 4. Primary Regions and Various Types of Nodes.

An example is shown in Figure 5, where the target on-path node is $n_2$. If the lengths of nodes $n_1$, $p_1$, $p_2$, $p_3$, and $p_4$ from flip-flops are shorter than that of $n_2$, $INS(n_2)$ is $\{n_1, p_1, p_2, p_3, p_4\}$. By only considering those nodes whose transitions have impacts on $n_2$ while excluding those nodes whose transitions have no impact on $n_2$, higher accuracy can be expected in assessing the delay increase at $n_2$.

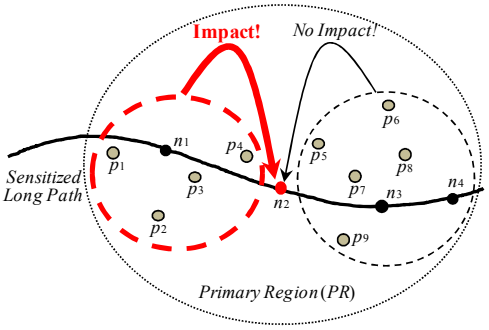


Figure 5. Impact Node Set.

### *E. TTR Metric Calculation*

The proposed metric calculates the TTR value for each test vector. For this purposes, three types of impact factors, namely, *node impact factor* ($IF_{node}$), *primary-region impact factor* ($IF_{primary\_region}$), and *path impact factor* ($IF_{path}$), are calculated first as follows:

• **Node Impact Factor**: In order to assess the impact of the nodes in a primary region $PR_j$ on an on-path node $n_i$, the node impact factor for node $n_i$ in primary region $PR_j$, denoted by $IF_{\text{node}}(n_i, PR_j)$, is defined as follows:

$$IF_{node}(n_i, PR_j) = \frac{\sum_1^{k_i}(\#Fout + 1)\ for\ switching\ nodes\ in\ INS(n_i)}{\sum_1^{m}(\#Fout + 1)\ for\ nodes\ in\ PR_j}$$

where $\#Fout$ is the fanout count of a node, $k_i$ is the number of nodes in $INS(n_i)$, and $m$ is the number of nodes in $PR_j$. Clearly, $IF_{\text{node}}(n_i,PR_j)$ is the weighted switching activity of the nodes in $INS(n_i)$ divided by the total weighted switching activity of all nodes in $PR_j$. As a result, transition time relations are taken into consideration.

• **Primary-Region Impact Factor**: Node impact factor values for all nodes in a primary region can be summed up for each sensitized long path in order to assess the primary region's impact on the sensitized long path. The primary-region impact factor for primary region $PR_j$ and the sensitized long path $P_k$, denoted by $IF_{primary\_region}(PR_j, P_k)$, is defined as follows:

$$IF_{primary\_region}(PR_j, P_k) = \sum_{i=1}^{m} IF_{node}(n_i, PR_j)$$

where $m$ is the number of on-path nodes for each sensitized long path in primary region $PR_j$.

• **Path Impact Factor**: The sum of the primary-region impact factor values of all primary regions for a sensitized long path is divided by the length of the path for the purpose of normalization. The path impact factor for sensitized long path $P_k$, denoted by $IF_{path}(P_k)$, is defined as follows:

$$IF_{path}(P_k) = \frac{\sum_{j=1}^{q} IF_{primary\_region}(PR_j, P_k)}{Length(P_k)}$$

where $q$ is the number of primary regions that the target sensitized long path $P_k$ passes through, and $Length(P_k)$ is the length of $P_k$, which is the number of its on-path nodes.

An example for calculating these impact factors is shown in Figure 5. Suppose that the fanout count of every node is 2, impact node sets for $n_1$, $n_2$, $n_3$, $n_4$ are $INS(n_1) = \{p_1\}$, $INS(n_2) = \{n_1, p_1, p_2, p_3, p_4\}$, $INS(n_3) = \{n_1, n_2, p_1, p_2, p_3, p_4, p_5, p_7, p_9\}$, and $INS(n_4) = \{n_1, n_2, n_3, p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8, p_9\}$. Also suppose that transitions occur at $n_1$, $n_2$, $n_3$, $n_4$,$p_1$, $p_3$, $p_5$, $p_7$, $p_9$, and that there is only one primary region. Based on the above definitions, $IF_{\text{node}}(n_1,PR) = 3/39$, $IF_{\text{node}}(n_2, PR) = 9/39$, $IF_{\text{node}}(n_3, PR) = 21/39$, and $IF_{\text{node}}(n_4, PR) = 24/39$. Furthermore, $IF_{primary\_region}(PR, P) = 57/39$, and $IF_{path}(P) = 57/156$.

• **TTR Value**: Although a test vector may sensitize more than one long path, the sensitized long path with the largest $IF_{path}$ determines the capture-safety of the test vector. Therefore, in order to check capture-safety for a test vector $v$, the $TTR$ value, denoted by $TTR(v)$ , is defined as follows:

$$TTR(v) = \text{Max}\{IF_{path}(P_1), \ldots, IF_{path}(P_r)\}$$

where $P_1$, $P_2$, …, and $P_r$ are sensitized long paths under the test vector $v$.

In capture-safety checking, the TTR value of a test vector is calculated and compared with a threshold to determine whether the test vector is capture-safe or not. The threshold can be set based on the worst TTR value. If transitions simultaneously occur at all nodes of a sensitized long path,

its TTR value is 1. For example, the threshold can be set as 10% of the worst value. This is similar to the general threshold of power budget and delay slacks in design phase. The time complexity of calculating the TTR value of a test vector is $O(m)$, where $m$ is the number of sensitized long paths by the test vector.

## III. Experimental Results

We implemented the proposed TTR metric using the C programming language, and conducted detailed analysis experiments with *b19* which is the largest ITC'99 benchmark circuit (129,130 gates / 6,130 FFs). This circuit was synthesized using Design Compiler® and placed & routed using IC-Compiler® with the SAED_EDK90nm library. 2,763 transition fault test vectors with 71.7% fault coverage were generated using TetraMAX®. All sensitized paths whose lengths are greater than 70% of the longest structural path were identified [12] as sensitized long paths. The total number of such paths is 2,016.

In order to obtain the golden result against which the proposed TTR metric was to be evaluated, we first conducted IR-drop analysis with PrimeRail® and obtained exact delay for each sensitized long path by using PrimeTime®. Test vectors were then sorted based on path delay increases and path lengths, and the worst $N$ test vectors were treated as capture-risky test vectors ($N$ = 5, 10, 20), shown as "Worst 5", "Worst 10", and "Worst 20" in Table 1 and 2. This capture-safety checking result was used as the golden result to assess the accuracy of the proposed TTR metric and the widely-adopted WSA metric. The WSA metric checks capture safety by calculating the value of weighted transitions for the whole circuit in the launch cycle and compare with a threshold, which is a percentage of the maximum WSA. In our experiments, we used 80%, 85%, and 90% as threshold percentages.

Table I shows the number of correct capture-risky judgments and the ratios of correct judgments by the TTR and WSA metrics. Note that the threshold of the TTR metric is set to 10% of the worst TTR value. From Table 1, it can be seen that the proposed TTR metric achieved much higher accuracy than the WSA metric with various thresholds. Particularly, the 5 capture-risky test vectors in the golden results of the case "Worst 5" are more likely to be real capture-risky test vectors than those of the cases "Worst 10" and "Worst 20". It can be seen that 4 of them were also correctly identified by the TTR metric.

TABLE I. Results of Capture-Safety Checking

| | # Correct Risky Judgments | | | | Correct Judgment Ratio | | | |
|---|---|---|---|---|---|---|---|---|
| | **TTR** | WSA | | | **TTR** | WSA | | |
| | | 80% | 85% | 90% | | 80% | 85% | 90% |
| *Worst 5* | **4** | 1 | 1 | 0 | **80%** | 20% | 20% | 0% |
| *Worst10* | **5** | 1 | 1 | 0 | **50%** | 10% | 10% | 0% |
| *Worst20* | **8** | 2 | 2 | 0 | **40%** | 10% | 10% | 0% |

Table II shows pessimistic and optimistic ratios. The pessimistic ratio is the percentage of test vectors incorrectly identified as capture-risky, while the optimistic ratio is the percentage of test vectors incorrectly identified as capture-safe. Clearly, pessimistic judgment causes over-test-induced yield loss, while optimistic judgment causes under-test-induced test quality degradation. From Table II, the TTR metric is more accurate since it has lower pessimistic and optimistic ratios than the other WSA-based metrics.

We used a workstation (Dual-Core AMD Opteron™: 2.8GHz / 16GB) for experiments. On average, the CPU time for checking the capture-safety of one test vector with the TTR metric and the WSA-based metrics were 4362.2 seconds and 0.1 seconds, respecitvely; while IR-drop-analysis checking using EDA tools took 7203.5 seconds.

TABLE II. Pessimistic and Optimistic Ratio

| | Pessimistic Ratio | | | | Optimistic Ratio | | | |
|---|---|---|---|---|---|---|---|---|
| | **TTR** | WSA | | | **TTR** | WSA | | |
| | | 80% | 85% | 90% | | 80% | 85% | 90% |
| *Worst 5* | **69%** | 99% | 98% | 100% | **20%** | 80% | 80% | 100% |
| *Worst10* | **62%** | 99% | 98% | 100% | **40%** | 90% | 90% | 100% |
| *Worst20* | **38%** | 98% | 95% | 100% | **70%** | 90% | 90% | 100% |

## IV. Conclusions

In this paper, we proposed the TTR metric, a novel metric that can more accurately identify capture-risky test vectors for at-speed scan testing. The TTR metric takes transition time relations into consideration in assessing the switching activity in the neighboring area around each sensitized long path. The advantage of the TTR metric for capture-safety checking was demonstrated by detailed analysis on the largest ITC'99 benchmark circuit. Future work includes implementing the TTR metric in a complete capture-safe test generation flow.

## Acknowledgment

This work was partly supported by JSPS KAKENHI Grant-in-Aid for Scientific Research (B) 22300017.

## References

[1] Y. Zorian, "A Distributed BIST Control Scheme for Complex VLSI Devices," *Proc. VLSI Test Symp.*, pp. 4-9, 1993.

[2] X. Lin, et al., "High-Frequency, At-Speed Scan Testing," *IEEE Design & Test of Computers*, pp. 17-25, September-October, 2003.

[3] P. Girard, N. Nicolici, X. Wen (Editors), *Power-Aware Testing and Test Strategies for Low Power Devices*. Springer, New York, Oct. 2009

[4] P. Girard, et al., *Low-Power Testing (Chapter 7) in Advanced SOC Test Architectures - Towards Nanometer Designs*, Morgan Kaufmann, 2007.

[5] K. Noda, et al., "Power and Noise Aware Test Using Preliminary Estimation," *Proc. VLSI-Int'l Symp. on VLSI Design, Automation and Test*, pp. 323-326, 2009.

[6] V. R. Devanathan, et al., "A Stochastic Pattern Generation and Optimization Framework for Variation-Tolerant, Power-Safe Scan Test," *Proc. Intl. Test Conf.*, Paper 13.1, 2007.

[7] X. Wen, et al., "On Low-Capture-Power Test Generation for Scan Testing," *Proc. VLSI Test Symp.*, pp. 265-270, 2005.

[8] S. Remersaro, et al., "Low Shift and Capture Power Scan Tests," *Proc. VLSI Design*, pp. 793-798, 2007.

[9] X. Wang, et al., "A Novel Architecture for On-Chip Path Delay Measurement," *Proc. Int'l Test Conf.*, Paper 12.1, 2009.

[10] X. Wen, K. Miyase, et al., "Critical-Path-Aware X-Filling for Effective IR-Drop Reduction in At-Speed Scan Testing," *Proc. Design Automation Conference*, pp. 527-532, 2007.

[11] N. Ahmed, et al., "Transition Delay Fault Test Pattern Generation Considering Supply Voltage Noise in a SOC Design," *Proc. Design Automation Conf.*, pp. 533-538, 2007.

[12] Y. -C. Ju, et al., "Incremental Techniques for the Identification of Statically Sensitizable Critical Paths," *Proc. Design Automation Conf.*, pp. 541-546, 1991.