Timing-Accurate Estimation of IR-Drop Impact on Logic- and Clock-Paths During At-Speed Scan Test

Holst, Stefan; Schneider, Eric; Wen, Xiaoqing; Kajihara, Seiji; Yamato, Yuta; Wunderlich, Hans-Joachim; Kochte, Michael A.

Proceedings of the 25th IEEE Asian Test Symposium (ATS'16) Hiroshima, Japan, 21-24 November 2016

doi: http://dx.doi.org/10.1109/ATS.2016.49

Abstract: IR-drop induced false capture failures and test clock stretch are severe problems in at-speed scan testing. We propose a new method to efficiently and accurately identify these problems. For the first time, our approach considers the additional dynamic power caused by glitches, the spatial and temporal distribution of all toggles, and their impact on both logic paths and the clock tree without time-consuming electrical simulations.

Preprint

General Copyright Notice

This article may be used for research, teaching and private study purposes. Any substantial or systematic reproduction, re-distribution, re-selling, loan or sub-licensing, systematic supply or distribution in any form to anyone is expressly forbidden.

This is the author's "personal copy" of the final, accepted version of the paper published by IEEE.[1]

[1] **IEEE COPYRIGHT NOTICE**
©2016 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

Timing-Accurate Estimation of IR-Drop Impact on Logic- and Clock-Paths During At-Speed Scan Test

Stefan Holst[1], Eric Schneider[2], Xiaoqing Wen[1], Seiji Kajihara[1],
Yuta Yamato[3], Hans-Joachim Wunderlich[2], Michael A. Kochte[2]

[1]Kyushu Institute of Technology, Iizuka, Japan, Email: {holst,wen,kajihara}@cse.kyutech.ac.jp
[2]University of Stuttgart, Stuttgart, Germany, Email: {schneiec,kochte,wu}@iti.uni-stuttgart.de
[3]Nara Institute of Science and Technology, Nara, Japan, Email: yamato@is.naist.jp

***Abstract*—IR-drop induced false capture failures and test clock stretch are severe problems in at-speed scan testing. We propose a new method to efficiently and accurately identify these problems. For the first time, our approach considers the additional dynamic power caused by glitches, the spatial and temporal distribution of all toggles, and their impact on both logic paths and the clock tree without time-consuming electrical simulations.**

I. Introduction

The instantaneous current drawn by gates when launching delay tests during at-speed scan testing can lead to excessive regional IR-drop. IR-drop affects both gates on logic paths and buffers on clock trees, which may change the circuit timing significantly [1], [2], [3] as shown in Figure 1.

If excessive IR-drop affects a long sensitized logic path during scan test, it may fail although the path would meet the timing requirements during nominal functional operation. This results in over-testing, which leads to unnecessary and costly yield-loss [4]. If excessive IR-drop affects clock buffers, the capture clock may reach some flip-flops much later. This is known as test clock stretch, which leads to delay test quality degradation as paths violating their timing requirements may wrongly pass the test [5], [1].

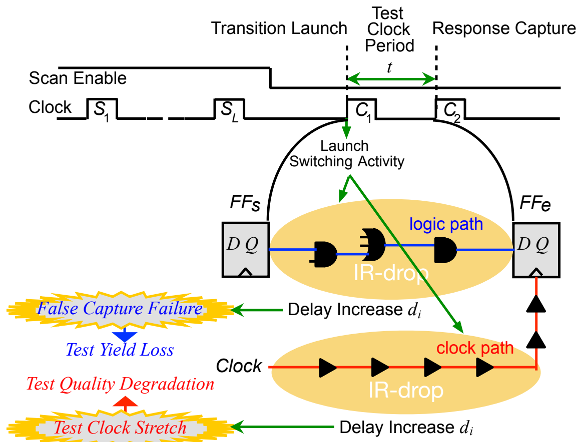

Fig. 1. IR-drop impact on logic paths and clock paths during at-speed scan testing.

A viable estimate of the IR-drop impact on the circuit timing requires the consideration of both, the *spatial proximity* of power-consuming switching activity to the path under consideration as well as the *temporal relation* between this activity and the signal stabilization times along the path under consideration.

Switching at a cell causes IR-drop not only at the cell itself but also at neighboring cells. Especially the cells sharing a power/ground rail at the lowest metal layer and nearest vias with the switching cell are highly affected since the switching current flows through the same rail, which usually has much higher resistance than power stripes at higher layers [6]. Thus, areas in the layout with many switching cells will show a higher IR-drop as the compound effect of the power needed for each switching puts a lot of strain on only a few power/ground rails.

The delay of a cell increases with larger IR-drop, i.e., reduced effective supply voltage at the cell. In the context of the logic circuit, this delay increase is only visible and relevant at the point in time where the cell executes its final, stabilizing transition within the current clock cycle. Active power is usually consumed over some time in the first half of the clock cycle as transitions travel through the combinational circuit. Cells at the very beginning tend to be less affected by IR-drop, because by the time of their stabilizing transition there has not been enough switching activity in the circuit yet. Parasitic elements on the power rails can delay the drop in supply voltage even further as reported in [7].

Yet another reason why the consideration of logic transitions over time is important are multiple switchings of a signal (glitches) in the circuit. It has been reported that up to 70% of dynamic power can be attributed to these glitches [8], so they need to be taken into consideration in order to get meaningful information on the switching activity of any circuit.

The detailed analysis of the complex interactions among switching activity, power supply noise and circuit delay currently requires electrical level simulations with tools like SPICE. However, it is still impractical to simulate large designs using SPICE. Even dedicated tools for dynamic power grid analysis at electrical level [9] allow only the analysis of very few patterns in large designs.

To obtain reasonable estimates on dynamic power consumption without simulations at electrical level, many state-of-the-art power simulation tools are usually based on timing simulation at logic level combined with weighted switching activity (WSA) to estimate strain on the power distribution network [4]. While logic level timing simulation is essential to model the impact of glitches, it is still the bottleneck in all power estimation tools as a huge amount of toggle data needs to be produced and aggregated to obtain the results of interest.

The work in [10] combines statistical static IR-drop estimation method with simulation-based dynamic estimation. The statistical IR-drop estimation assumes a fixed toggle activity in the circuit and is able to identify blocks in a larger design that are likely to suffer from excessive IR-drop. The dynamic estimation performs gate-level timing simulation using a commercial simulator, estimates regional IR-drop based on the switching activity of each gate and then re-simulates the design with modified gate delays that take IR-drop into account. Although very accurate, the dynamic estimation was done only on very few patterns as it is computationally very expensive with conventional tools.

The temporal relation between the stabilization time on a path and the switching activity surrounding this path was first recognized in [11], which uses static timing analysis to determine the time relation between nodes. Static timing analysis, however, only yields upper bounds for the stabilization times of signals and does not consider glitches preceding the final stabilization that may impact the path delay.

The general relationship between switching activity, IR-drop and cell delay increase has been addressed in several previous works and is understood quite thoroughly. The work in [12] shows how a model that links switching activity to IR-drop can be constructed from an electrical level analysis of a few select tests. It has also been shown that there is a linear relationship between IR-drop and cell delay [13], [14].

The main bottleneck of all of the above-described methods is the generation of accurate switching activity data. The timing simulation methods used in previous methods are either computationally too expensive to be applied to more than a few patterns, or their accuracy is very limited as they ignore the impact of glitches. With the recent advances in timing simulation algorithms using *General-Purpose Computing on Graphics Processing Units* (GP-GPU), it has become feasible to calculate all signal transitions and their timing relations in an extremely efficient manner [15].

In this work, we show, for the first time, how to employ efficient GPU-based timing simulation for the calculation of IR-drop impact on both logic paths and clock paths during at-speed scan test. The major challenge in GPU-based computing is the bandwidth bottleneck between GPU and CPU. Therefore, our method keeps all the toggle data in the GPU memory while selectively querying the switching activities in specific regions and during specific time intervals for estimating the impact on certain paths of interest. We apply our method to an at-speed scan test scenario to check the long sensitized paths in delay tests for IR-drop related delay changes.

Section II provides some fundamental background on the timing of delay tests and the target of our estimation method. Section III describes the data-parallel simulator that runs on the GPU and calculates all the necessary toggle data. Section IV shows how the toggle data is used to accurately estimate IR-drop for logic and clock paths. Section V evaluates the performance of our simulator and discusses how the IR-drop impact results using accurate toggle data compare to previous methods.

II. Delay Test Timing

Consider the timing at an output signal o between launch and capture of a delay test given as test pair v shown in Fig. 2. The signal delay $d(o,v) = d_n(o,v) + d_i(o,v)$ of the longest sensitized path leading to o is the sum of its nominal delay $d_n(o,v)$ and an IR-drop-induced delay change $d_i(o,v)$.

The time between the launch of the second vector of the delay test and the capture, $t(o,v) = t_n + t_i(o,v)$, is the sum of the nominal test cycle time t_n and an IR-drop-induced test time change $t_i(o,v)$. This test time change is caused by IR-drop at buffers of the clock path. The slack at output o under test vector pair v is the difference between the test time and the signal delay $s(o,v) = t(o,v) - d(o,v)$.

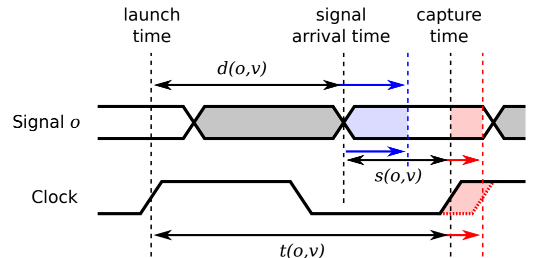

Fig. 2. Relation between test time, delay and slack at an output o under test vector pair v during at-speed scan-test.

Any robustly activated and propagated delay fault on the longest sensitized path to the output o with a delay larger than $s(o,v)$ will be detected at o. If excessive IR-drop affects the logic path such that $d_i(o,v)$ is so large that the slack $s(o,v)$ becomes negative even in good circuits, a false capture failure occurs, which leads to over-testing if not handled correctly [16]. If excessive IR-drop affects the clock tree and leads to test clock stretch ($t_i(o,v) > 0$), delay fault coverage decreases as a delay fault only slightly larger than the nominal slack $s_n(o,v) = t_n - d_n(o,v)$ is not detected but may cause problems during functional operation.

The goal of this work is to obtain estimates for $d_i(o,v)$ and $t_i(o,v)$ from given test and circuit data. The IR-drop impact of the logic paths $d_i(o,v)$ is the sum of the delay changes of all the cells and interconnects on that path. Let logic path $LP(o,v)$ be the set of cells on the longest sensitized path for test v to the endpoint o. We can then write $d_i(o,v) =$

$\sum_{g \in LP(o,v)} \delta_i(g, v, d(g, v))$ with $\delta_i(g, v, t)$ being the IR-drop induced delay change at a single gate g under test v at time t. Note that the above formula is recursive as each term depends on the overall delay $d(g, v)$ of its path prefix. Still, the result can be calculated in linear time as long as the gates on the path are evaluated in-order.

To ease the following discussion, we assume that the clock tree is perfectly balanced under nominal conditions. Typically, the clock tree is the most carefully designed part in layout synthesis so that the launch clock cycle reaches all flip-flop at the same time. Also, we assume that the power network recover from shift switching activity before the delay test begins. Our method can be easily extended to incorporate unbalanced clock trees as well. The amount of test clock stretch $t_i(o, v)$ is the sum of the delay changes of all the clock buffers and lines from the clock source to the flip-flop at o. With clock path $CP(o)$ being the set of all cells and interconnects on the clock tree leading to o, the clock stretch is calculated in much the same way as the delay change on logic paths: $t_i(o, v) = \sum_{b \in CP(o)} \delta_i(b, v, d(b, v))$.

What remains now is the calculation of $\delta_i(g, v, t)$ for any cell g (logic gate or clock buffer), delay test v, and time t. In this calculation, both the spatial relation between g and power-consuming aggressors as well as the temporal relation between t and the toggles on the aggressors have to be considered.

As discussed earlier, several works already provide good models to calculate delay change of gates from switching activity of surrounding gates [12], [13], [14]. The main goal of our work is to provide the means for efficiently obtaining the switching activity data in the first place. Our method is not limited to any specific IR-drop model, so instead of calculating the delay directly, we obtain a measure that corresponds to the IR-drop impact on victim gates and paths. The IR-drop impact measure $w_i(g, v, t)$ for a single cell g is calculated using a simple weighted switching activity (WSA) [4] model. Let R_g be the set of aggressor cells near g, and let f_c be the number of cells driven by a cell c (its fanout). Then,

$$w_i(g, v, t) = \sum_{c \in R_g} a(c, t) \cdot (f_c + 1)$$

with $a(c, t)$ being the number of transitions of a cell c until time t. The IR-drop impact value is accumulated along the logic paths and clock paths under consideration. Again, we use this simple model for demonstration purposes only. In applications to actual designs, the model will change based on the technology node used and the extracted parasitics. These technology dependent models, however, are beyond the scope of this work.

III. Data-Parallel Timing Simulation on GPUs

To generate necessary toggle data from delay test pattern pairs, we use a modified version of a GPU-accelerated time simulator proposed in [15]. The simulator operates on combinational gate-level netlists and exploits multiple dimensions of data-parallelism to propagate the patterns from the inputs to the outputs.

Before the circuit is simulated, some easy pre-processing steps are necessary. First, the combinational logic circuit is extracted from the post-layout netlist. This is done by replacing the flip-flops with pseudo-primary inputs and outputs. The timing information for each cell is loaded from the SDF (Standard Delay Format) file generated by a commercial physical synthesis or parasitics extraction tool. The simulator supports an industry-standard pin-to-pin delay model including short pulse filtering. The combinational gate-level netlist is then topologically ordered and uploaded to GPU memory.

Unlike common timing simulators whose fundamental unit of computation consists of a single signal change (an event), the GPU-based simulator bundles all toggles of a signal into complete waveforms, which are then propagated through the circuit. A waveform contains the complete history of one signal from its initial value and its transitions to its final value. Once the waveforms at all the inputs of a cell are available, its output waveform is calculated in one atomic operation using the input waveforms, the logic function of the cell and its delay parameters. Fig. 3 illustrates how the waveforms are propagated through the circuit level by level in topological order and how data-parallelism is exploited. Further details can be found in [15].

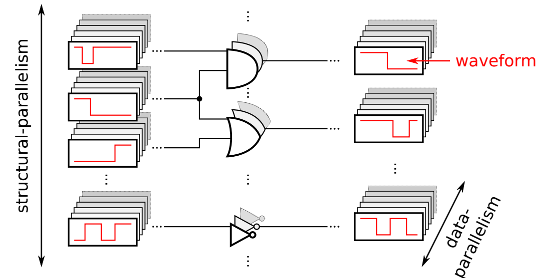

Fig. 3. Waveform propagation through the combinational circuit with two dimensions of parallelism: structural- (gates) and data-parallelism (stimuli).

One of the advantages of waveform-based simulation is a very space-efficient storage of signal toggles—each toggle is represented by only one floating-point value (the time of the transition). This allows for an enormous amount of switching information to be kept in the GPU memory. During the propagation of waveforms, the original simulator overwrote the toggle data of previous signals as much as possible to improve memory efficiency even further. For the present application, however, the toggle data of all the internal signals is necessary to calculate the IR-drop impact later on, so the simulator has been modified to not overwrite previously calculated signal values and keep all data in memory. This reduces memory efficiency, but our experiments show that even with this modification, modern GPU cards are able to hold the toggle data for hundreds of delay tests at a time. Once a set of delay tests are simulated in this fashion, all toggle data

is available in GPU memory and the CPU host process will request only the data necessary for IR-drop impact estimation as described in the following section.

IV. Estimation of IR-Drop Impact for Paths

Any logic path or clock path in the circuit can be described as a set of cells C. While the GPU memory holds all toggle data for a test, the stabilization time t_c for each cell $c \in C$ is obtained by simply reading the relevant data from GPU memory. The IR-Drop impact value of the path C and test v is computed as $w_i(v, C) = \sum_{c \in C} w_i(c, v, t_c)$. The value for each cell $w_i(c, v, t_c)$ is obtained by requesting the weighted switching activity for test pattern pair v, region of c, and time interval until t_c from the GPU. Upon request, the necessary toggle data is read and summed up with the appropriate weights. The layout regions, i.e., set of aggressor cells, for each cell c as well as the granularity of time intervals can be freely defined by the user.

Fig. 4 shows a simple example of a standard cell based layout with power stripes interspersed with cell rows. A simple way to define these region of aggressor cells is to group all cells in the layout that influence each other's supply voltage because they share power rails. In this simple example, the regions form a partition over all the cells. Overlapping regions are also supported, for instance to model influence between neighboring rows.

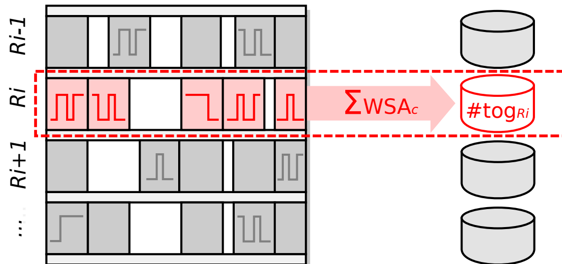

Fig. 4. Regions for counting WSA in a standard cell based design.

Again, our method is not limited to the particular example shown above as regions can be chosen freely by considering the particular aspects and structure of the power grid design and desired spatial resolution.

The time domain in the IR-drop impact estimation is quantized. The cycle time t is divided into a number of intervals or time slices. For each interval only those toggles are counted that fall into that time slice. The number of time intervals as well as their distribution within the complete cycle time is configurable depending on the desired resolution and performance requirements.

Each request of the host to the GPU is cached, so if multiple cells of a path are located in the same region and stabilize in the same time slice, summation is performed only once.

V. Experimental Results

The proposed simulation approach has been evaluated for the largest ITC'99 benchmark circuits. The designs have been synthesized and layouted with the clock distribution network using a 90nm standard cell library. For each design 1,000 random input stimuli pairs have been applied for evaluation. All experiments were executed on a host machine containing eight Intel® Xeon® processors clocked at 3.0 GHz and 128GB of RAM. Furthermore NVIDIA® Tesla® K80 dual-GPU accelerators with 2×2096 cores and 2×12GB of memory have been utilized for performing the timing simulation.

We first compare the switching activity of untimed and timing-accurate simulation over the whole clock period. For each applied pattern pair the WSA ratio between timing-accurate and untimed simulation was calculated. The histogram in Fig. 5 shows the amount of tests over all WSA ratios. The majority of the patterns exhibit 30% (b15 and b17) to 80% (others) higher WSA due to glitches. This confirms earlier reports that disregarding glitches may lead to severe underestimation of the power dissipation.

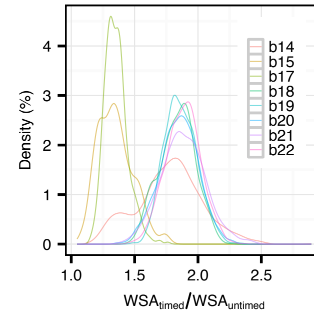

Fig. 5. Pattern distribution of WSA ratio between timing-accurate (WSA_{timed}) and untimed simulation ($WSA_{untimed}$)

The detailed results are given in Table I. Columns 1–2 contain the name and the number of gates of the respective circuits. The runtime required to compute the transition counts and WSA using the GPU-accelerated time simulator is shown in Col. 3. Columns 4–7 compare the toggle counts obtained from untimed logic simulation and timing-accurate simulation in more detail.

First, we report the total number of gate evaluations, where no glitches and hence no toggle loss occurred, due to either static signals without value change (Col. 4) or single signal transitions (Col. 5). Then, we report the number of evaluations of signals showing glitches, distinguishing signals with static (Col. 6) or dynamic hazards (Col. 7). These glitches cause an underestimation of the toggle count in untimed simulation compared to timing-accurate simulation since they are not considered in untimed simulation.

The total WSA computed over all patterns is shown in Col. 8–9, which was calculated for both untimed and timing-accurate simulation respectively. The last two columns contain the average ($\Delta avg.$) and the maximum ($\Delta max.$) deviation of the WSA of untimed and timing-accurate simulation per pattern.

TABLE I: EVALUATION OF 1000 RANDOM STIMULI PAIRS

Circuit(1)	Gates(2)	Runtime(3)	Simulation Cases (Gate Evaluations)				Total WSA		Per Pattern WSA Difference	
			Glitch-free		with Glitches					
			Static(4)	Transition(5)	Static(6)	Dynamic(7)	untimed(8)	timed(9)	$\Delta avg.$(10)	Δmax(11)
b14	4043	0.876s	2341804	1189118	398186	113892	3788498	6823208	+77.4%	+178.9%
b15	7348	1.289s	5429645	1582777	280035	55543	4479156	6160792	+35.9%	+76.3%
b17	22874	3.089s	16831005	5018383	853079	171533	13858622	19020954	+35.3%	+66.8%
b18	55515	6.603s	38680991	12116197	3478877	1238935	36355704	67743180	+82.5%	+113.2%
b19	81108	9.394s	56286256	18069462	5021902	1730380	53333678	100639262	+84.4%	+124.2%
b20	9073	1.427s	5060433	2766955	957958	287654	8681754	16167042	+83.7%	+120.5%
b21	8831	1.527s	4930737	2654482	950534	295247	8389259	15910545	+86.9%	+141.8%
b22	13859	2.013s	7802333	4157742	1453306	445619	13098815	24738657	+86.4%	+125.5%
			(67.78%)	(23.47%)	(6.61%)	(2.14%)				

For each investigated circuit, the timing-accurate simulation of the complete test set was finished within seconds. In almost 10 percent of all the gate evaluations performed, differences in the toggle counts of untimed and timed simulation due to hazards occurred (Col. 6–7). Among them, 76 percent were static hazards identified by timing-accurate simulation. These static hazards are the major contributor to the total WSA difference shown in Col. 8–9. In general, timing-accurate simulation showed 70% more signal toggles compared to untimed simulation, with a maximum difference of 24 transitions for a pattern at a single node in circuit b19.

In the next experiment, the spatial and temporal distribution of the WSA is investigated. Each standard cell row in the layout forms a *region*. For each circuit, the time of the latest arriving transition at any output for all patterns is determined and used as a clock period. The clock period is then split into 32 equidistant *time slices* for a more fine-grained evaluation.

Fig. 6 a) shows the switching activity in the circuits over time. High switching activity is observed in the beginning with 50 percent of the total activity already taking place within the first five time slices for most of the circuits. Fig. 6 b) shows the distribution of average switching activity during test application over both space and time for the benchmark circuit b14. As shown, the switching activity in many layout regions tends to start relatively low directly after launch before rising during the first third of the clock cycle and then tapering off towards the capture time. This observation underlines the need for considering both temporal and spatial relations for estimating IR-drop impact on paths.

In a last experiment, the IR-drop impact on the longest sensitized logic paths and their associated clock paths were investigated (Table II). For each circuit, the three longest sensitized paths between a pseudo-primary input and a pseudo-primary output in the complete test were selected by obtaining and sorting the latest stabilization times LS at pseudo-primary outputs. The IR-drop impact (Total WSA) on these paths is shown for three different simulation models. The first model (Col. 6) is untimed simulation in which all glitches are ignored. The second model (Col. 7) is timing-accurate simulation with support for glitches, but without considering the temporal relation between switching activities and stabilization times. The third model (Col. 8) is timing-accurate simulation with complete support for glitches and temporal relations. Columns 9–11 shows the IR-drop impact on associated the clock path with the same models. The WSA calculated by timing-accurate simulation is generally higher and significant differences in the increase can be observed. While trace 3 of b15 only shows a modest increase from 1226 to 1832, other values show three times higher IR-drop impact caused by glitches (e.g. b14, trace 1). IR-drop impact estimations based on untimed simulations without support for glitches are clearly less accurate. We can observe the same pattern in the comparison of the third model with the second model. In some cases identical impact results are observed while in other cases, only a fraction of toggle activity has an impact on the path delay. This shows that both the support of glitches and temporal relations are essential for meaningful IR-drop impact estimation.

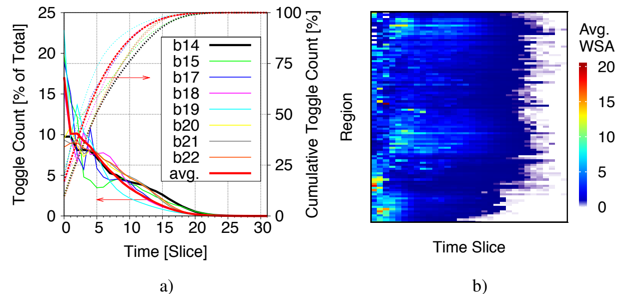

Fig. 6. Switching activity over the clock interval for all circuits a) and average WSA over all patterns per regions and time slice in b14 b).

VI. CONCLUSIONS

IR-drop induced false capture failures and test clock stretch are severe problems in at-speed scan testing. We proposed a new method to efficiently and accurately identify these problems. Our GPU-based logic timing simulator is able to produce the necessary toggle data in a matter of seconds and allows for detailed investigation of switching activity in given regions and at given times. We confirmed that the consideration of glitches

TABLE II: LOGIC PATH AND CLOCK PATH WSA OF THE THREE LONGEST TRACES

Circuit(1)	Trace(2)	$LS^{(3)}$	Pattern(4)	Output(5)	Total WSA Logic Path			Total WSA Clock Path		
					untimed(6)	timed		untimed(9)	timed	
						full(7)	sliced(8)		full(10)	sliced(11)
b14	1	2.805ns	861	reg3_reg[12]	1635	5661	4554	134	240	240
	2	2.762ns	213	reg2_reg[24]	1704	3762	2876	135	321	321
	3	2.701ns	976	reg3_reg[15]	1525	2453	1886	162	210	210
b15	1	2.348ns	344	InstQueue_reg[7][5]	882	1288	1112	133	211	211
	2	2.339ns	344	InstQueue_reg[7][6]	847	1317	1139	133	211	211
	3	2.333ns	208	InstQueue_reg[15][5]	1226	1832	1480	200	212	212
b17	1	2.791ns	791	P2_InstQueue_reg[8][6]	1945	2673	2353	311	395	395
	2	2.780ns	791	P2_InstQueue_reg[8][1]	1930	2622	2304	334	394	394
	3	2.778ns	791	P2_InstQueue_reg[8][5]	1868	2620	2302	284	368	368
b18	1	4.261ns	905	P2_P1_InstQueue_reg[1][7]	7603	18039	16461	457	831	831
	2	4.250ns	824	P2_P1_InstQueue_reg[13][7]	7758	21582	19625	691	1141	1141
	3	4.074ns	40	P1_P1_InstQueue_reg[1][0]	6612	14080	12736	682	1632	1632
b19	1	3.791ns	33	P1_P1_P1_InstQueue_reg[3][1]	2021	4503	3997	876	1436	1436
	2	3.789ns	743	P1_P1_P1_InstQueue_reg[6][1]	7712	17324	15891	1105	1885	1885
	3	3.756ns	33	P1_P1_P1_InstQueue_reg[3][2]	1781	4849	4354	849	1521	1521
b20	1	3.110ns	590	P1_reg0_reg[13]	2782	6500	5406	243	623	623
	2	3.107ns	317	P1_reg0_reg[16]	1002	3052	2962	240	530	530
	3	3.098ns	317	P1_reg0_reg[13]	3304	7722	6562	261	673	673
b21	1	3.012ns	931	P1_reg2_reg[14]	2366	5568	5042	218	440	440
	2	2.859ns	742	P2_reg2_reg[19]	3111	7515	6371	175	579	579
	3	2.813ns	888	P1_reg3_reg[8]	1824	3900	3342	222	400	400
b22	1	3.067ns	189	P1_reg3_reg[17]	2796	7150	5984	341	527	527
	2	3.063ns	621	P1_reg3_reg[21]	2782	5526	4749	251	489	489
	3	2.995ns	189	P1_reg3_reg[18]	2951	7177	5960	341	527	527

in aggressors as well as spatial proximity and temporal relation between aggressors and victims is essential for accurate IR-drop impact estimation. The presented approach provides an excellent platform to identify and analyze IR-drop related timing problems in large circuits and for a high number of test patterns.

ACKNOWLEDGMENT

This work was jointly supported by JSPS Grant-in-Aid for Scientific Research (B) #25280016, JSPS Grant-in-Aid for Scientific Research on Innovative Areas #15K12003, DFG under grant WU 245/16-1 (PARSIVAL), JSPS Grant-in-Aid for the Promotions of Bilateral Joint Research Projects (Japan-Germany), and the German Academic Exchange Service DAAD (supported by the BMBF).

REFERENCES

[1] R. Franch, P. Restle *et al.*, “On-chip Timing Uncertainty Measurements on IBM Microprocessors,” in *Proc. IEEE Int’l Test Conf. (ITC)*, Oct. 2008, pp. 1–7.

[2] J. Saxena, K. M. Butler *et al.*, “A Case Study of IR-Drop in Structured At-Speed Testing,” in *Proc. Int’l Test Conf. (ITC)*, Sep. 2003, pp. 1098–1104, Paper 42.2.

[3] K. Asada, X. Wen *et al.*, “Logic/Clock-Path-Aware At-Speed Scan Test Generation for Avoiding False Capture Failures and Reducing Clock Stretch,” in *Proc. IEEE 24th Asian Test Symp. (ATS)*, Nov. 2015, pp. 103–108.

[4] P. Girard, N. Nicolici, and X. Wen, Eds., *Power-Aware Testing and Test Strategies for Low Power Devices*. Springer New York, 2009.

[5] J. Rearick and R. Rodgers, “Calibrating Clock Stretch During AC Scan Testing,” in *Proc. IEEE Int’l Test Conf. (ITC)*, Nov. 2005, pp. 1–8, Paper 11.3.

[6] J. Ma and M. Tehranipoor, “Layout-Aware Critical Path Delay Test Under Maximum Power Supply Noise Effects,” *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 30, no. 12, pp. 1923–1934, Dec. 2011.

[7] P. Pant and J. Zelman, “Understanding power supply droop during at-speed scan testing,” in *Proc. IEEE 27th VLSI Test Symp. (VTS)*, May 2009, pp. 227–232.

[8] A. Shen, A. Ghosh *et al.*, “On Average Power Dissipation and Random Pattern Testability of CMOS Combinational Logic Networks,” in *Proc. IEEE/ACM Int’l Conf. on Computer-Aided Design (ICCAD)*, Nov. 1992, pp. 402–407.

[9] Synopsys, “PrimeRail—In-Design Rail Analysis for Place-and-Route Engineers,” 2009, whitepaper.

[10] N. Ahmed, M. Tehranipoor, and V. Jayaram, “Transition Delay Fault Test Pattern Generation Considering Supply Voltage Noise in a SOC Design,” in *Proc. ACM/IEEE 44th Design Automation Conf. (DAC)*, Jun. 2007, pp. 533–538.

[11] K. Miyase, R. Sakai *et al.*, “A Capture-Safety Checking Metric Based on Transition-Time-Relation for At-Speed Scan Testing,” *IEICE Transactions*, vol. 96-D, no. 9, pp. 2003–2011, 2013.

[12] Y. Yamato, T. Yoneda *et al.*, “A Fast and Accurate Per-Cell Dynamic IR-drop Estimation Method for At-Speed Scan Test Pattern Validation,” in *Proc. IEEE Int’l Test Conf. (ITC)*, Nov. 2012, paper 6.2, pp. 1–8, Paper 6.2.

[13] K. Arabi, R. Saleh, and X. Meng, “Power Supply Noise in SoCs: Metrics, Management, and Measurement,” *IEEE Design Test of Computers*, vol. 24, no. 3, pp. 236–244, May 2007.

[14] J. Jiang, M. Aparicio *et al.*, “MIRID: Mixed-Mode IR-Drop Induced Delay Simulator,” in *Proc. 22nd Asian Test Symp. (ATS)*, Nov. 2013, pp. 177–182.

[15] S. Holst, M. E. Imhof, and H.-J. Wunderlich, “High-Throughput Logic Timing Simulation on GPGPUs,” *ACM Trans. on Design Automation of Electronic Systems (TODAES)*, vol. 20, no. 3, pp. 1–22, Jun. 2015.

[16] X. Wen, K. Enokimoto *et al.*, “Power-Aware Test Generation with Guaranteed Launch Safety for At-Speed Scan Testing,” in *Proc. IEEE 29th VLSI Test Symp. (VTS)*, May 2011, pp. 166–171.