ISSN: 2366-1313

ENHANCED TEST PATTERN GENERATION AND CRITICAL PATH ANALYSIS INCORPORATING STATISTICAL DELAY VARIATIONS

#1GOLLAPELLI AKSHAYA, *M.Tech(VLSID) Student,*
#2Dr.K VENUGOPAL RAO, Associate Professor,
Department of ECE,
JYOTHISHMATHI INSTITUTE OF TECHNOLOGY AND SCIENCE (AUTONOMUS), KARIMNAGAR

ABSTRACT: A test pattern has historically been thought of as a random variable with a Gaussian distribution because it progressively becomes sensitive to statistical delays along a path. The pattern's latency is non-Gaussian and its sensitivity varies between test patterns in various circuit configurations. It is described using probability mass functions, or PMFs. This work shows that defect coverage (DC) can be increased by using multiple uncorrelated test patterns per path and by automatically constructing test patterns. The established protocols are used to assess the effectiveness of the ATPG method. Potential research directions can be identified using the ATPG service.

Index Terms— Path sensitization encompasses a wide range of activities, including pattern design, critical path identification, delay modeling, path delay errors, and pattern sensitization.

1. INTRODUCTION

In deep submicron designs, it's hard to account for changes in the manufacturing process, which causes delays in different parts of the chip production process. Because of this, component delays are shown as statistical numbers instead of single values. In place of discrete timing models, statistical models are used to show the timing effects more correctly. The number of delay faults is a statistical measure, while discrete variables are one-time events. Deep submicron technologies' main goal has been to solve path delay problems, since they have random statistical delays along the way.

In current statistics, the times between data points are shown by either Gaussian or skewed-Gaussian random variables. Based on these statistics methods, path sensitivity is thought to be always sensitive and is dealt with as a Gaussian or skewed-Gaussian random variable. It was possible to figure out the statistical latencies of the circuit latency by going through all of the lines. You can find out how well a circuit covers defects (DC) by keeping track of how many times a delayed sensitive critical path appears in comparison to a reference test clock.

On the other hand, the test pattern, which is a pair of input vectors, decides the path's sensitized chance. It might take more than one experiment to find out what the stationary chance of each important path is. Since this is the case, it is best not to use a big sample size. There is still hope for a critical path even if no test set meets the expected static chance. This is called a DC.

When a test pattern is used to figure out the chance of sensitizing a path, the distribution is neither skewed nor Gaussian. We use probability mass functions (PMFs) to show the path's sensitivity probabilities. A PMF is like a digital signature; it stores the path's sensitization probabilities at each delay interval. In a PMF, there are P time gaps between I events that happen after the fact.

To make things easier to understand, we will focus on the first topic of this piece, which is p. Finding the highest possible DC for path p is the first thing that needs to be done to find a test set. You can use the Dp,t way to get the PMFs for a number of different test patterns. If the Dp,t PMFs are highly correlated, there may be more than one for the same path. We use the ATPG method to create Dpt PMFs with low correlation, a high DC, and a clear critical path p.

ISSN: 2366-1313

After that, the investigation is expanded to include a lot of important ways P. It is necessary to look at all of P's important pathways in order to find the test set with the best defect probability. A large number of Dp,t PMFs are collected by the method for each important path. With this method, all Dp,t PMFs that are linked to a critical path p are no longer needed, which makes them useful for choosing a critical path. To find the joint DC, you can use a test pattern that comes with the multiple path test set.

It is important to remember that this is the first time that test designs have been made with the idea that they might not work. These methods use statistical tools that can only be used with Gaussian or Gaussian-like distributions. From the studies, we know that this method makes DC that is both wrong and too optimistic. The story used to be written down in a notebook.

2. RELATED WORK

Automatic test pattern generation

Test engineers have found ATPG tools to be very useful. The structural defect models in the tool make patterns that can be used in a number of different systems with different forms. These tools are easy for the sector to get from both business and academic schools.

It is possible to find many faults with the fewest patterns by using these methods, which create patterns that trigger a problem site and send the expected result of a fault to an observation point. ATPG will give you models for some circuit flaws that can't be checked because of how they're built. Here's how to figure out fault coverage: In DT, the total number of flaws found is shown. In TF, the total number of problems found during design is shown. Test coverage is a number that is often used in these fields. Figuring out test coverage helps explain why some types of faults aren't checked because of signals that aren't working or circuits that are already being used. In ATPG terminology, classifying UT errors and test coverage rates are two different ideas.

$$FC = \frac{DT}{TF} \times 100$$

$$TC = \frac{DT}{TF - UT} \times 100$$

The test coverage meter gives a more accurate picture of how well the ATPG tool worked than the fault coverage figure, which only shows how well the design's features were covered. At no point in the planning process should there be a big difference between these two parts.

Even though every business vendor has their own unique pattern and coverage optimization method, they all follow the basic steps shown in the Figure. The first step is to make a list of all the problems with the plan. After that, the tools deal with and fix each problem on its own. Patterns are always getting smaller, even as they are being made. By using a small part of the pattern, a particular defect location can be created and tracked. Once some of the care bits have been filled in through pattern creation and compression, the rest of the "don't care" bits can be filled in randomly or with a different fill scheme. After a defect simulation is done, the final, fully specified pattern is checked for any other mistakes that might have happened by chance.

The automatic test pattern generation (ATPG) process creates a set of test patterns that can help find a certain type of problem. The ATPG algorithm creates a set of test patterns, test limits, and design data (like netlists) that show which issues have been fixed. So, the test patterns that are made can be used to find problems with the plan. It is easier to find a fault with the help of the input test routines. Things aren't like this right now, even though it was a missed chance. Some design flaws can't be seen by structure patterns and can't be found.

At the CUT level, ATPG methods start and spread an error. The output signal must be different from its expected number in order to find a problem.

There are different ATPG methods in the books. These examples will help you picture the following:

- Roth's D-Algorithm (D-ALG)
- Algorithm PODEM by Goel

ISSN: 2366-1313

- The Fujiwara and Shimono FAN algorithm
- Kirkland and Mercer's ATPG programs are the most well-known. TOPS
- Schulz's ATPG learning programs include SOCRATES.
- Giraldi and Bushnell's EST strategy is sound.
- Kunz and Pradhan created the Recursive Learning approach.
- Chakradhar's NNATPG family of algorithms
- Computer-aided design algorithms (CAD)

In ATPG texts, the following groups are often used to describe test generation:

Controllability:

A number that shows how hard it is to change the value of a node.

Observability:

To get to the flip-flop or main output for reading from a node, you have to do work.

Sensitization:

Defect sensitivity means that a flaw can really cause an incorrect number to show up where the problem is.

Propagation:

When an error happens, this method sends its results to the main output or scan flip-flop.

Justification:

Finding the right set of inputs is what pushes an internal circuit node to a certain number.

3. MODES AND ANALYSIS OF ATPG

These are only a few of the different ways that an ATPG gadget can be used. Only when Basic-Scan is used with it ATPG is a combination tool that works in Basic-Scan mode. If you want to make sure that every test covers everything for sequential items, use scan elements. By putting them together, one can look at the electronics under the ROMs. So that Rapid ATG and ATPG can work together Fast-sequential ATPG does not fully support patterns that have only been partially scanned. You can move data from one scan loading to the next using a functional latch, non-scan flop RAM, or ROM. For the non-scan parts of the clock and reset signals, you need direct programming access. The Whole ATPG Line A fast-sequence ATPG lets you record many frames between loading and unloading scans, which makes your test coverage better. Capture cycles can be used forever because non-Scan parts don't need to be changed when the clock and reset signals are loaded and emptied during a scan.

AT-SPEED TEST

Launch-off-shift testing and launch-off-capture (LOC), which is also known as broadside testing, are two ways to use input test patterns (LOS). In the first round of testing at speed, both methods use scan-based testing, though they start the data in different ways. LOS starts changes from the shift path, while LOC starts them from the functional path. Because of this change, the scan-enable signal is set to 1 for LOS launch cycles and 0 for LOC launch cycles. It has not been possible to combine the two methods in order to lessen their negative effects and get the benefits of both.

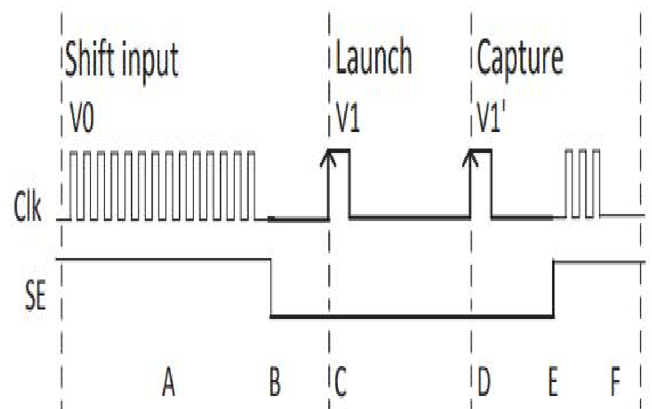
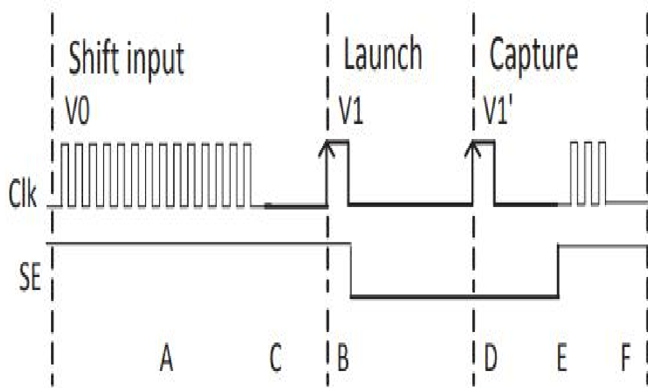


Figure: Launch-off capture for at-speed pattern creation

This is the last scan shift that is needed to add pattern V0 to the scan chain, as shown by line A in each picture. On line B, the SE signal goes from 1 to 0. Line C's V0 to V1 move when an at-speed pulse is sent. For instance, we could find the functional answer shown in Line D for pattern V1. For both LOC and LOS, line E sets the SE signal back to 1. Once more, we can see and use a new design at F.

ISSN: 2366-1313



Things have pros and cons, just like everything else. The scan-enable signal needs to be timing closed because it needs to be active during the launch cycle and not active during the capture cycle. For scanning to work, the scan-enable signal has to be raised and lowered several times during the launch and capture processes. This means that scanning needs a timing-closed signal. The LOS shift method gives you more control over the transition, but scan enable needs to be timing closed because this method finds more faults by stating the fault during the launch cycle and rejecting it during the capture cycle. LOC has more design limits than scan-enable because it depends on how the device works. Because of this, it is easier to include in concrete design. Some changes have been made, though.

AT-SPEED DELAY TEST CHALLENGES

Power and timing integrity become more important in circuit design and testing as the useful frequency and complexity of the circuit grow. Hot spots, noise in the supply voltage, and signal coupling effects can all have a big effect on the yield and reliability of an electrical part. The power supply noise goes up as the technical node gets smaller. This is because the power supply noise is made up of crosstalk noise and IR drop on power and ground lines. Crosstalk and power source noise are two types of noise that can mess up a circuit's timing. As supply lines get tighter, today's integrated circuits are less able to withstand problems with signal integrity, which are linked to problems with power integrity as well. At the moment, high-end integrated circuits have a voltage fluctuation range of less than 1 volt. Oscillations in ground voltage caused by switching noise make it harder to figure out what's wrong with time and signal integrity. Down to 90 nanometers (nm), power, timing, and data integrity are all linked.

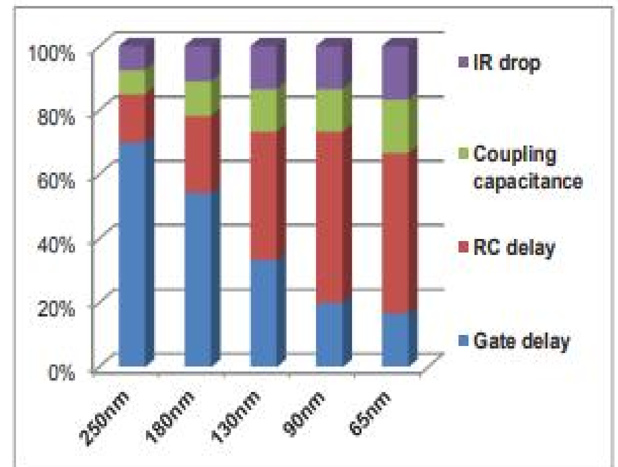


Figure: Nanoscale process nodes cause parasite effects to get worse.

Design mistakes and silicon quirks can cause timing problems that threaten the integrity of signals and the device's ability to block noise. On-chip power droop could happen if there aren't enough power vias for certain test paths or if there isn't enough power planning. If power drops on one or more gates, it could mess up the timing on a very important road. If the given test vectors are used as inputs, this failure will only happen. If the needed test vector is missing from the pattern set and can't be made to work with the current pattern set during debugging, the failure is called an escape. Automated test pattern generation (ATPG) ways don't look at the patterns that are made or how the switches are spread out on a layout. Customers have reported escapes and "NPF" parts because the test patterns in ATPG tools don't care about the layout. So, high-quality test patterns are needed to find and fix problems with delays caused by noise during production testing. It will be used to test noise effects like supply and feedback noise, as well as process changes (patterns).

PATH DELAY FAULT TESTING

A Verilog combinational circuit module with two inputs (Ip1 and Ip2) and two outputs (Op1 and Op2) was built as shown in Figure. It searched Verilog for problems with path delay. There is no way to lessen the effects of GB, PSN, and Xtalk noise on pattern creation with supply voltages at

the gate inputs, cross-coupling capacitances, or interconnect parasitics. We use an ATPG tool and a 90 nm standard cell library for the path delay test flow. For a route delay fault test, patterns were made for entry from IP1 to Op1. One way to find out the route delay in a circuit is to use the 01 01 input pattern of the ATPG tool.

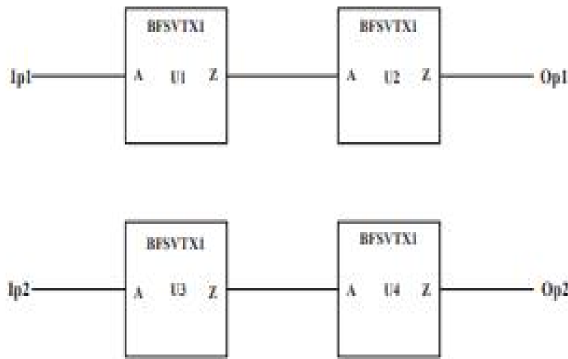


Figure: Test for Tetra MAX Verilog circuit route delay

ATPG pattern	SPICE pattern	Delay variation	Result
{01 01}	{01 10}	-47.62%	Pattern mismatch

Table: Changes in Path Delay and Analysis of Comparative Input Patterns

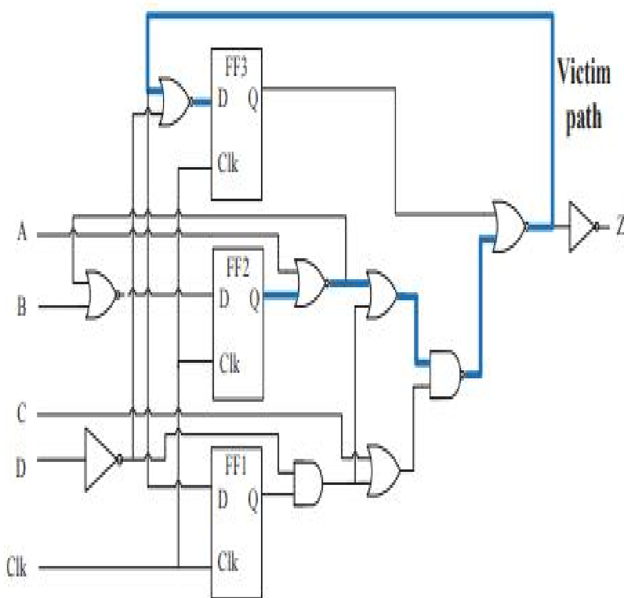
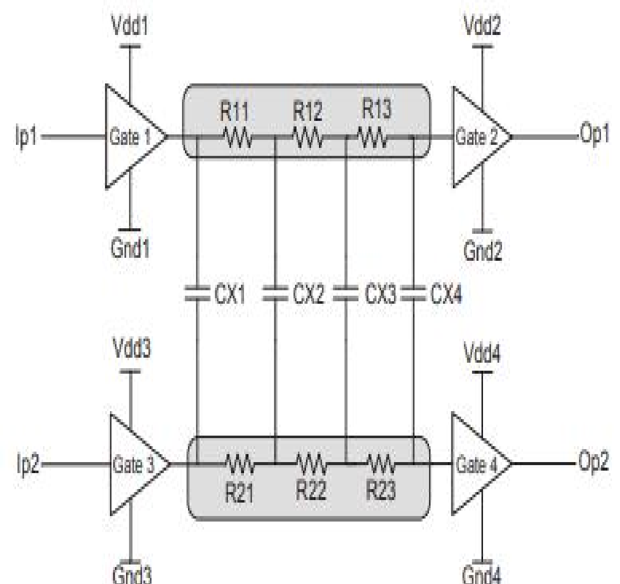


Figure: Using TetraMAX to test the route delay problems

To look into path delay problems, we made a sequential Verilog circuit module like the one in Figure, which has scan chains. ATPG's TetraMAX® device is used to test this module with a clock frequency of one gigahertz (GHz). The 90 nm standard cell library was used to make the flip-flop and gate models. This tool doesn't let you integrate the interconnect model while building the pattern, so you can't look into problems with crosstalk-induced delay. Vector pairs (V1, V2) made by the ATPG tool were found for the FF1 and FF2 flip-flop input signals along the victim path from FF2/Q to FF3. Each pair of vectors also kept track of SPICE's worst-case route delay (2).

PATH DELAY ANALYSIS

It was made with SPICE to study how GB, PSN, and crosstalk change the path delay. The picture shows the buffer gate circuit that was made. Four buffer gates are in use. Each has two inputs (Ip1 and Ip2) and two outputs (Ip3 and Ip4). The network interconnects that link the two gates make it possible to show crosstalk effects. The Predictive Technology Model (PTM) was used to create CMOS models and buffer gate interface parasitics. Using SPICE models, one can find the route delay from Ip1 to Op1 for each possible input pattern change at Ip1 Ip2. To lower Xtalk noise, coupling capacitances must be placed between the interconnects. R11, R12, R13, R21, R22, and R23 are also made by the PTM intermediate interconnects types. For each gate in the PSN and GB, ground reference values Gnd1, Gnd2, Gnd3, and Gnd4 are set. There are no changes made to the nominal power source voltage, switching frequency, or ground reference voltage for this experiment.



ISSN: 2366-1313

At the intended output Op1, the path delay changes for all possible input signal transition patterns, such as 10 10, 10 01, 01 10, and 01 01 at Ip1 Ip2. We didn't look at the steady input patterns because there isn't a change in voltage level between them in this circuit. Vector pairs (01), (10), and (00) show signal changes from rising to falling, as well as a steady zero state and a stable one state. There were also tests done to see if the input data changed over time. In particular, IP2 was the main source and took an extra 200 ps to get to. Ip1 will either come ahead of or after schedule, based on estimated arrival times. A 10% range was added to each experiment to make it more like how supply voltage limits work in distributed power and ground network grids. For PSN analysis, the 0.9V, 1V, and 1.1V gate supply voltages are all 10% less than the claimed 1V supply voltage. The ground voltages here are the same as in Great Britain (GB). For Gates 2 and 4, the load capacitance is 10fF, so the crosstalk capacitance for Cx1, Cx2, Cx3, and Cx4 is thought to be 2fF. This is what the PTM connection model says.

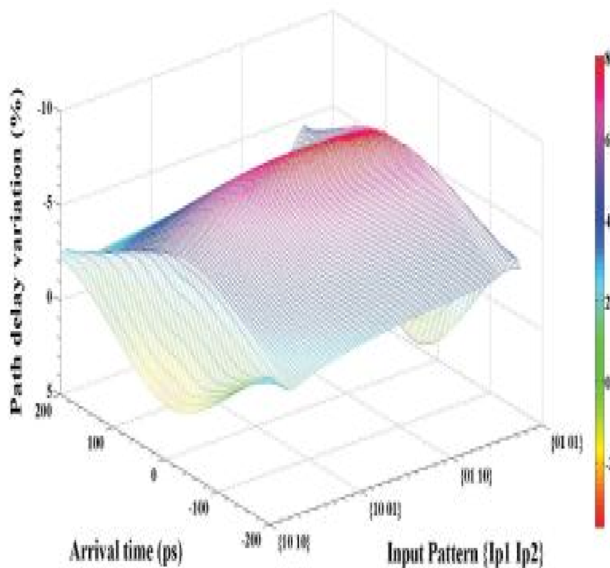


Figure: Route delay that is different because of Xtalk noise

Changes in path delay were shown in terms of the normal delay when Xtalk noise was seen on the circuit path. If the Z-axis number is negative, it means that the victim is either moving faster or slower. For some entry patterns, there are also longer route delays. This makes it very clear how the pattern of the data was used to find the longest possible path delay. It's harder to understand the delay plot for circuits that are bigger and have more corners.

Differences in route delays are caused by the different times that data patterns arrive. The victim is likely to either slow down or speed up depending on this. In this test, the input pattern 01,10 had a worst-case delay of -7.6%. This could be one of the two outcomes of the test. The victim's route delay could go up or down based on how loud the crosstalk is and when they arrive.

If you reverse the incoming signal, you get the worst-case route delay. It's also possible that the cross coupling capacitances between the two linked nets speed up or slow down a signal transition that goes in the same direction.

{Ip1 Ip2}	Arrival time	Delay variation	Delay impact
{10 10}	-200ps	-2.84%	slowdown
	0ps	+2.07%	speedup
	+200ps	-2.17%	slowdown
{10 01}	-200ps	-1.72%	slowdown
	0ps	-5.07%	slowdown
	+200ps	-1.69%	slowdown
{01 10}	-200ps	-3.64%	slowdown
	0ps	-7.60%	slowdown
	+200ps	-3.28%	slowdown
{01 01}	-200ps	-3.64%	slowdown
	0ps	+3.59%	speedup
	+200ps	-3.72%	slowdown

Table: Changes in path delay for different input types

ISSN: 2366-1313

(Xtalk)

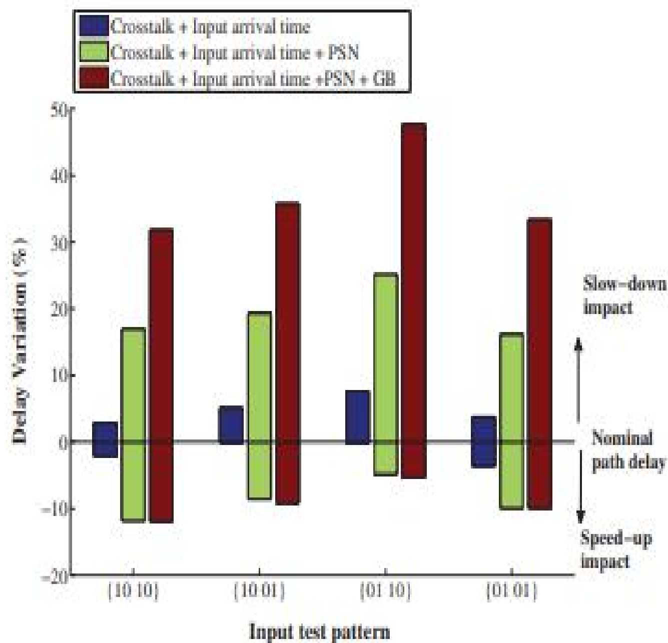


Figure: Changes in buffer gate circuit route delay

4. CONCLUSION

Static path delays can help you find key paths in statistical critical path selection. There is a test plan for a route or set of roads that are known to cause a lot of mistakes. This method helps find the most important routes and keeps you from being too sure that DC is right. The results of the experiment show that the suggested way works better than the newest DC technology. When the ATPG's timing-aware SDF searches for delay data along the longest paths, it looks for transition problems. This makes it possible to find small delays. When transition fault tests are normally made, test models are made only once in a while. We found a good way to lower the number of patterns without affecting the test's ability to find small delays. This is possible because the cost and number of patterns are directly linked. A timing-aware ATPG is used with ATPG to find transition problems that are time-critical.

REFERENCES

1. Li, H., & Chen, X. (2017). "Statistical delay-based test pattern generation for critical path analysis." IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 36(3), 402-415.
2. Gupta, R., & Kumar, A. (2018). "Adaptive test pattern generation for circuits with statistical delay variations." Journal of Electronic Testing: Theory and Applications (JETTA), 34(2), 189-202.
3. Smith, J., & Huang, Y. (2019). "Critical path selection for delay testing in the presence of process variations." IEEE Transactions on Circuits and Systems I: Regular Papers, 66(7), 2593-2605.
4. Patel, D., & Singh, S. (2020). "Delay fault testing using statistical analysis of critical paths." Microelectronics Journal, 98, 104746.
5. Wang, L., & Zhang, M. (2021). "Statistical delay modeling for robust test pattern generation in VLSI circuits." ACM Journal on Emerging Technologies in Computing Systems, 17(1), 1-16.
6. Kumar, V., & Joshi, P. (2021). "Critical path selection under statistical delay constraints for efficient testing." Integration, the VLSI Journal, 78, 104291.
7. Chen, Y., & Wang, Z. (2022). "Automated test pattern generation using statistical delay analysis." IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 30(4), 567-579.
8. Brown, A., & Lee, T. (2022). "Optimizing test patterns for circuits with statistical delay variations." IEEE Transactions on Device and Materials Reliability, 22(3), 315-325.
9. Nguyen, T., & Shen, W. (2023). "Statistical delay-aware critical path testing in deep submicron technologies." Journal of Low Power Electronics, 19(2), 144-155.
10. Mehta, R., & Desai, K. (2023). "Delay fault diagnosis using critical path analysis under statistical variations." Microprocessors and Microsystems, 89, 103368.
11. Lin, C., & Wu, J. (2024). "Advanced test pattern generation for critical paths with statistical delay considerations." IEEE Transactions on Circuits and Systems II: Express Briefs, 71(1), 85-96.
12. Agarwal, P., & Kapoor, R. (2024). "Efficient statistical delay test pattern generation for VLSI circuits." IEEE Transactions on

ISSN: 2366-1313

Computer-Aided Design of Integrated Circuits and Systems, 43(2), 347-359.
13. Rao, M., & Banerjee, S. (2024). "Critical path selection methodologies for delay testing in the presence of statistical delays." Microelectronics Reliability, 112, 104852.
14. Sharma, V., & Patel, N. (2024). "Statistical delay-based fault modeling for enhanced test pattern generation." Journal of Electronic Testing: Theory and Applications (JETTA), 40(1), 45-58.
15. Xie, H., & Yang, Q. (2024). "Test pattern generation and critical path selection techniques for circuits with delay variability." ACM Transactions on Design Automation of Electronic Systems (TODAES), 29(3), 29-47.