Delays in the Generation of Test Patterns and in the Selection of Critical Paths

[#1] Divi Saikrishna, *M.Tech Student,*
[#,2] N. Umapathi , [#3]Sathish Mothe, *[#2, 3] Associate Professor,*
Department of ECE,
[#1, 2, 3] jyothishmathi Institute of Technology and Science, Karimngar, Telangana.

Abstract: Because it constantly increases the sensitivity to the statistical delay of a path, a test pattern has typically been considered a random variable with a Gaussian distribution. In various circuit configurations, the pattern-created delay is non-Gaussian, and its sensitivity varies between test patterns. Probability mass functions are used to express it (PMFs). By employing many uncorrelated test patterns per path, this article explains how autonomous test pattern generation (ATPG) can increase defect coverage (DC). The ATPG method's impact is evaluated by comparing it to previous methods. It has been shown that critical pathways can be identified for further exploration using the ATPG supplied.

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

For deep submicron designs, it is difficult to regulate for variations in the manufacturing process, resulting in varying component delays between chips that have been made. Component delays are expressed as statistical numbers instead of discrete values as a result of this. Statistical models rather than discrete timing models are utilized to better capture the timing effects. Contrary to discrete variables, the frequency of delay faults is a statistical measure. Deep submicron technologies have focused on path delay problems due to the fact that they involve distributed statistical delays along the way[1].

Statistics already in place use Gaussian or skewed-Gaussian random variables to represent the delays in data. These statistical methods treat path delays as Gaussian or skewed-Gaussian random variables, which presume that a path is sensitive at all times. When all of the paths have been selected, statistical delays of the circuit's latency are taken into account. Counting the number of

times a delayed sensitized critical path occurs in relation to a reference test clock can be used to determine a circuit's defect coverage (DC) (Tclk).

The sensitized likelihood of the path, on the other hand, is determined by the test pattern (a pair of input vectors). It may be necessary to conduct a large number of tests to determine the static probability of each critical path. Therefore, a large sample size should be avoided. A DC may be optimistic even if no test set can achieve the estimated static probability for a critical path.

When a test pattern is used to determine the likelihood of sensitizing a path, the distribution is not Gaussian or skewed Gaussian. Thus, the path's sensitization probabilities are represented as probability mass functions (PMFs), a digital signature encoded with the path's sensitization probabilities at each delay interval. There are P time intervals between I delay instances in a PMF[2].

For the sake of clarity, let's focus on the first issue in this article: p. Finding a test set entails locating the path p's largest possible DC. When determining the PMFs for various test patterns, one might turn to the Dp,t approach. There may be redundant Dp,t PMFs for the same path if the Dp,t PMFs are closely correlated. Low correlation Dpt PMFs with a well-defined critical route p and a high DC are created using an ATPG[3] approach.

Afterwards, the inquiry expands to encompass a number of important paths P. All of P's major pathways must be considered in order to find a test set with the best defect probability. The method stores a huge number of Dp,t PMFs for each significant path. This approach, when applied to all Dp,t PMFs connected with a critical path p, makes them redundant, making them valuable for critical path selection purposes. The joint DC can be determined using a test pattern that appears in the multiple path test set.

As a side note, this is the first time that test patterns have been designed taking into account the probability of failures. These methods use statistical methods that are only applicable to Gaussian or Gaussian-like distribution. According to the results of the trials, this methodology may lead to DC that is both erroneous and too optimistic. This draft of the story was previously published[4].

2. RELATED WORK

AUTOMATIC TEST PATTERN GENERATION

Test engineers have greatly benefited from ATPG tools. Structural fault models utilized by the tool generate patterns that can be used in a variety of systems with diverse architectures. Commercial and academic institutions alike make these instruments readily available to the industry.

As many defects as feasible can be identified with the least amount of patterns using these methods, which generate patterns that stimulate a problem site and send the projected consequence of a defect

to an observation point. Patterns for some flaws that cannot be checked due to the circuit's topology will be provided by ATPG. Fault coverage is calculated as follows: DT is the total number of discovered flaws, and TF is the total number of in-design defects that we have detected. One typical statistic employed by these tools is test coverage. Due to redundant circuits or tied-off signals, some fault types are inaccessible for testing, hence the equation takes this into consideration when computing test coverage. Classifying UT mistakes and test coverage percentages are two different things under ATPG terminology[5].

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

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

.

While the fault coverage statistic shows how well the ATPG tool covered the design's details, the test coverage meter is a more accurate indicator of how well the ATPG tool performed. There should never be a large discrepancy between these two variables while designing something.

In spite of the fact that each commercial vendor has its own proprietary pattern and coverage optimization technique, they all follow the same basic approach indicated in the Figure. The first step is to develop a list of design defects. The tools then go through and fix each of the problems one at a time. During the construction of a pattern, the patterns are constantly compressed. A specific fault site can be stimulated and observed with only a small portion of the pattern. The remaining don't care bits can be filled in at random or using another fill scheme once the pattern formation and compaction process has filled in a portion of the care-bits. A fault simulation is run on the final, fully described pattern in order to discover any extra issues that may have occurred by chance.

Creating a set of test patterns that may be used to identify a specific sort of problem is done through automated test pattern creation (ATPG). Algorithm ATPG can produce a collection of test patterns from a test protocol, test limitations, and design data (such as netlists) (which specifies which problems are addressed). The produced test patterns can then be used to find design problems[6]. The input test patterns can be used to identify a defect that can be found. You could argue this is simply a missed opportunity, but it's not the case here. Structural patterns are unable to detect certain forms of design errors, referred to as invisible problems.

Activating and propagating an error at the CUT level is how ATPG algorithms work. In order to detect a problem, the output signal must differ from its predicted value[7].

There are various ATPG algorithms in the literature. To give you an idea, here are some examples:

Roth's D-Algorithm (D-ALG)

Algorithm PODEM by Goel

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)

ATPG literature typically uses the following words to describe test generation:

Controllability:

A measure of how difficult it is to change the value of a node.

Observability:

It's tough to get the value of a node to make it to the main output or flip-flop for scanning.

Sensitization:

Fault sensitization is the process of allowing a fault to generate a real erroneous value at the problem location.

Propagation:

This approach is used to transmit error effects to the main output or scan flip-flop.

Justification:

An internal circuit node is driven to a specified value by identifying the input combination required.

3. DIFFERENT ATPG MODES AND ANALYSIS

Just a few of the various operational modes that an ATPG instrument has to offer are listed here.

As a combinational only tool, Basic-Scan ATPG can be used in Basic-Scan mode. Use scan elements for sequential items if you want to ensure that all of your tests are covered. The circuitry hidden underneath the ROMs can be exposed by merging them. ATPG Sequential High-Speed ATPG Partially scanned designs are not completely supported by Fast-Sequential ATPG[8]. A working latch, non-Scan Flop RAM and ROM can be used to transfer data between scan loading and scanning unloading. There must be direct programming access for non-scan sections of the clock and reset signals. All of the ATPG's Sequences More test coverage can be achieved with a Fast-Sequential ATPG since it allows for numerous capture cycles between scan loading and unloading. Capture cycles can be employed indefinitely because no modifications are necessary on non-Scan components during scan loading and unloading of the clock and reset signals.

AT-SPEED TEST

Launch-off-capture (LOC), also known as broadside testing, and launch-off-shift testing are two techniques to apply input test patterns (LOS). During the first at-speed test cycle, both techniques use scan-based testing, but the data is launched in a different manner. LOS uses the shift path, whereas LOC uses the functional path to start transitions. The scan-enable signal is set to 1 during the LOS[9] launch cycle and 0 during the LOC launch cycle as a result of this disparity. Attempts have been made to combine the two approaches in an effort to gain the advantages of both while also reducing their disadvantages, but they have failed.

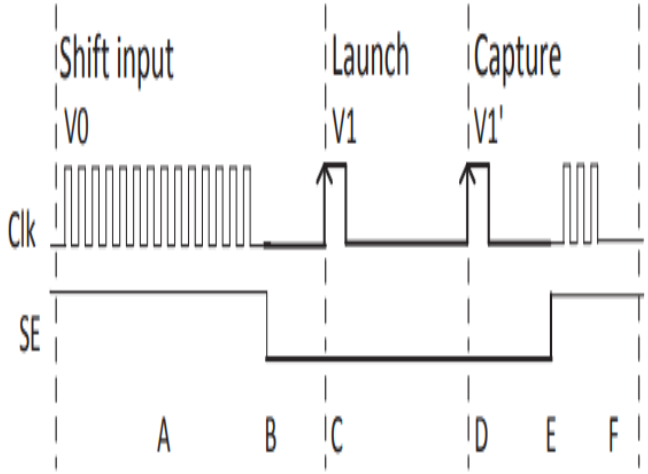
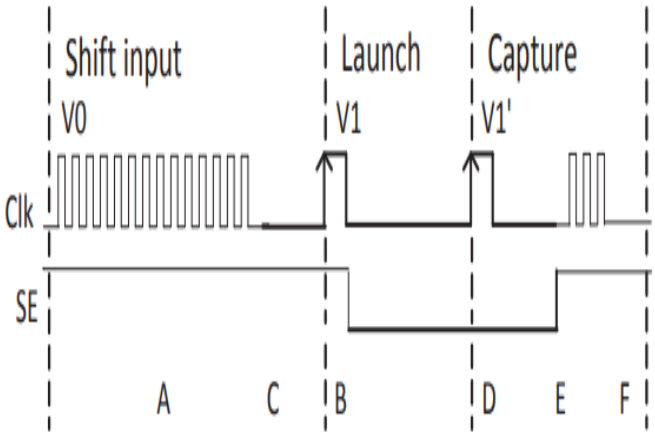


Figure: At-speed pattern generation using launch-off capture

This is the last scan shift required to enter pattern V0 into the scan chain, as seen by the line A in each photo. On line B, the SE signal goes from 1 to 0. Transition from V0 to V1 is triggered by an at-speed pulse on Line C. Using pattern V1 as an example, we can see the functional response shown in Line D. Line E resets the SE signal to 1 for both LOC and LOS. Our ability to see and apply a different pattern resumes at F[10].



There are benefits and drawbacks to everything, as is the case with everything. The scan-enable signal must be timing closed since it must be asserted during the launch cycle and abandoned during the capture cycle. In order to perform scanning, a timing-closed signal is required since the scan-enable signal must be asserted and abandoned sequentially during both launch and capture cycles. Scan enable must be timing closed, even if the LOS shift path allows more control over transition, because it must be asserted during the launch cycle and abandoned during the capture cycle to provide higher fault coverage in this technique. Scan-enable has less design restrictions than LOC because it is dependent on the device's operation. This makes it easier to incorporate in physical design. There are, however, some differences[11].

AT-SPEED DELAY TEST CHALLENGES

Power and timing integrity become increasingly critical in circuit design and testing as circuit complexity and functional frequency grow. Hot spots, supply voltage noise, and signal coupling effects can all have a substantial influence in terms of yield and reliability for an electronic component. Crosstalk noise and IR drop on power and ground lines (power supply noise) grow in percentage as the technology node shrinks. Circuit timing integrity can be degraded by noise sources such as crosstalk and power supply noise. As supply rails have become more constrained, so has the immunity of today's ICs to signal integrity difficulties, which are tied to power integrity concerns as well[12]. For high-end integrated circuit devices, the voltage fluctuation margin has decreased to less than 1 volt. Signal integrity and timing concerns are difficult to detect because of ground voltage oscillations generated by switching noise. At 90 nanometers (nm) and down, power, timing, and signal integrity are all linked.

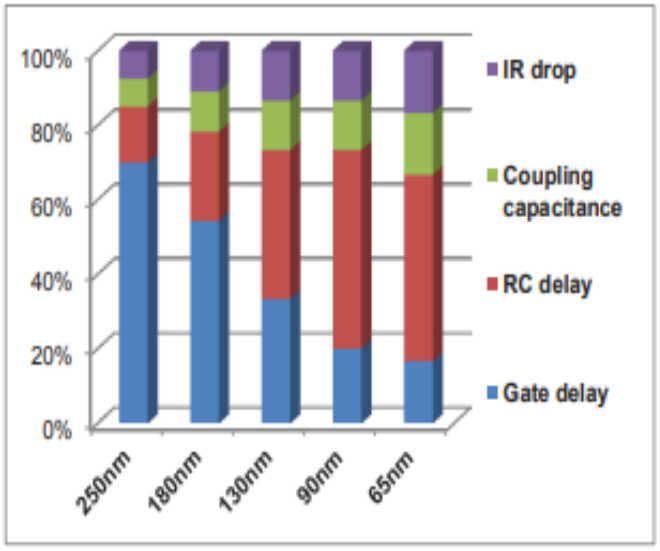


Figure: At nanometer process nodes, parasitic effects increase

Device noise immunity is weakened and signal integrity is compromised when design defects and silicon oddities combine to generate timing problems. Poor power planning or the absence of power vias for specified test vectors can lead to on-chip power droop. Timing errors can be caused by power droop affecting one or more gates on a crucial path. Only when specified test vectors are utilized as inputs may this failure occur. The failure is regarded an escape if the required test vector is not included in the pattern set and cannot be reproduced during diagnostics using the existing pattern set. Switching distribution on a layout or generated patterns is not taken into account by approaches to automatic test pattern generation (ATPG). ATPG tools' layout-unaware test patterns have resulted in customers reporting escapes and "NPF" sections. Therefore, high-quality test patterns are needed to detect and diagnose noise-induced delay concerns during production testing. Testing vectors will include noise effects such supply and crosstalk noise, as well as process changes (patterns)[13].

PATH DELAY FAULT TESTING

As shown in Figure a Verilog combinational circuit module with two inputs (Ip1 and Ip2) and two outputs was created to test for path delay faults in Verilog (Op1 Op2). No interconnect parasitics, cross-coupling capacitances, or supply voltages at the gate inputs are employed to account for the impacts of Xtalk noise, PSN, and GB during pattern formation. We use an ATPG tool and a 90nm standard cell library for the path delay test flow. IP1 to Op1 input patterns were constructed for a route delay fault test. Detecting a route delay in a circuit can be done using the ATPG tool's 01 01 input pattern.

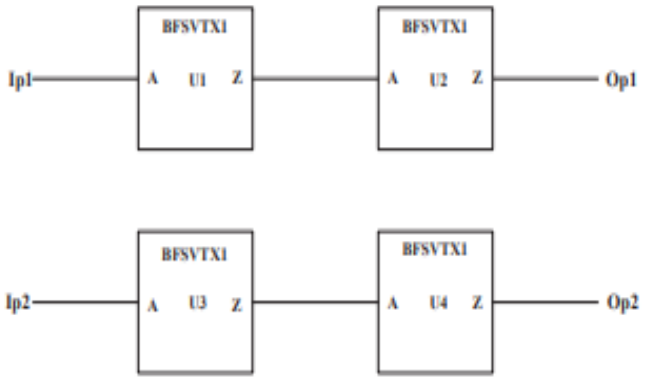


Figure: Verilog circuit for Path delay test in TetraMAX

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

Table: Input Pattern Comparison and Path Delay Variation

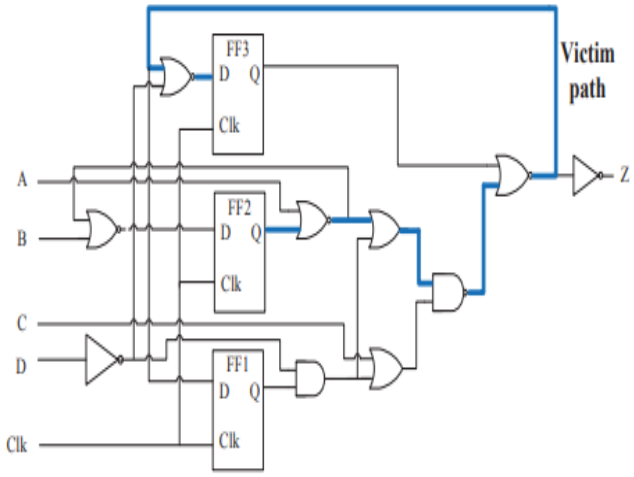
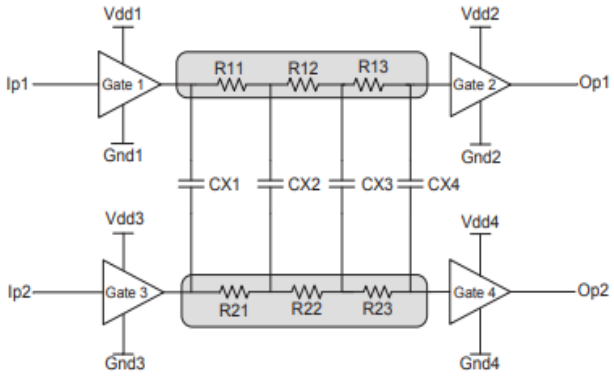


Figure: Path delay fault testing in TetraMAX

In order to test for path delay issues, we constructed a sequential Verilog circuit module like the one illustrated in Figure (with scan chains introduced). ATPG's TetraMAX® tool is used to perform delayed testing on this module at a clock frequency of one gigahertz (GHz). The 90nm standard cell library was utilized to generate the gate and flip-flop models. Because the tool does not allow the inclusion of interconnect models during pattern generation, crosstalk-induced delay concerns cannot be investigated. ATPG tool-generated vector pairs (V1, V2) for FF1 and FF2 flip-flop input signals, as well as the worst case route delay (2) recorded in SPICE by each pair of vectors, were detected along the victim path from FF2/Q to FF3[14].

PATH DELAY ANALYSIS

The picture displays the SPICE-designed buffer gate circuit used to examine the effects of crosstalk, PSN, and GB on path delay changes. This circuit is made up of four buffer gates, each with two inputs (Ip1 Ip2) and two outputs (Ip3 Ip4) (Op1 Op2). Network interconnects between the two gates are employed to exhibit crosstalk effects. Estimates of interconnect parasitics and CMOS models for buffer gates were made using Predictive Technology Model (PTM). SPICE simulations are utilized to establish the route delay from Ip1 to Op1 for all conceivable input pattern transitions at Ip1 Ip2. It is necessary to put coupling capacitances between the interconnects in order to produce Xtalk noise. R11, R12, R13, R21, R22, and R23 are also generated by the PTM intermediate interconnects models. Gnd1 Gnd2 Gnd3 Gnd4, are the supply and ground reference voltages for each gate in PSN and GB. Each of the three variables in this experiment is left at its default setting: nominal power supply voltage, ground reference voltage, and switching frequency.



Path delay modifications are seen at the target output Op1 for all potential input signal transition patterns, such as 10 10, 10 01, 01 10, and 01 01 at Ip1 Ip2. When analyzing the steady input patterns, we didn't bother to look at them because there is no voltage level transition between them in this circuit. There is a rising signal transition, a falling signal transition, a stable zero condition, and a stable one condition represented by vector pairs (01), (10) and (00). Tests were also carried out to see if there were any differences in the timing of input signals. - It took the main input Ip2 an additional 200 ps to arrive. According to estimated arrival times, Ip1 is either ahead or behind[15]. In order to accurately depict supply voltage restrictions in distributed power and ground network grids, all of these experiments used a 10% tolerance, which is an accurate portrayal. For PSN analysis, the gate supply voltages are 0.9V, 1V, and 1.1V, which are all 10% lower than the nominal 1V supply voltage. When it comes to ground voltages, they're the same as they were in Great Britain (GB). Assumed crosstalk capacitance of 2fF for Cx1 Cx2 Cx3 Cx4 due to the 10fF load capacitances of Gates 2 and 4 (derived from PTM interconnect model)[16].

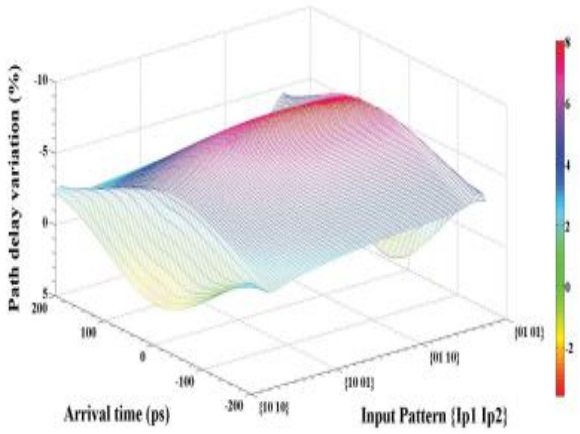


Figure: Path delay variations due to Xtalk noise

When Xtalk noise was present on the circuit path, path delay fluctuations were expressed in terms of the nominal delay. Negative z-axis numbers indicate that the victim is speeding up or slowing down. There are also longer route delays for some input patterns. It is easy to see here how the

worst-case path latency was calculated using the input pattern shown here. It becomes increasingly difficult to interpret the delay plot for larger circuits that include several corners.

Path delay variations are influenced by the arrival times of distinct input patterns. The victim's progress is described as being slowed or accelerated. There was a worst-case delay of -7.6 percent in this experiment for the input pattern 01,10. There are two possible outcomes from this experiment.

- The route delay of a victim can grow or decrease depending on the amount of crosstalk noise and the time of arrival.
- When the input signal is reversed, the worst-case route delay occurs.
- It is also possible that the cross coupling capacitances between the two interconnecting nets slow or speed up a same-direction signal transition.

{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: Path delay variations for different input patterns (Xtalk)

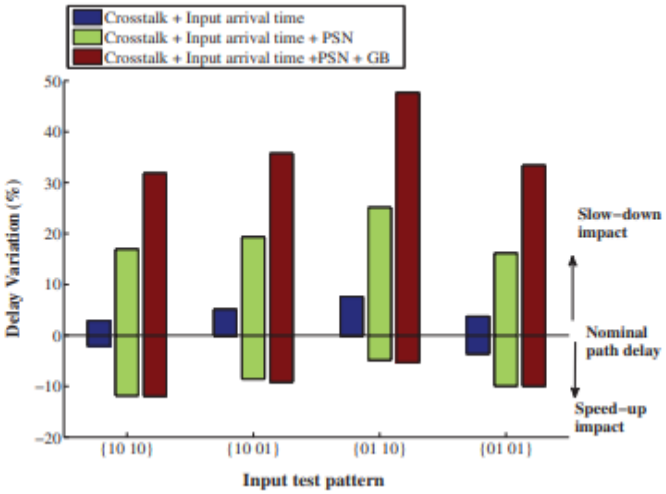


Figure: Path delay variation of buffer gate circuit

4. CONCLUSIONS

Static path delays are utilized in statistical critical path selection to identify critical paths. For a path or collection of paths with a high defect probability, a test approach is offered. It is a useful technique for avoiding overconfidence in DC accuracy by identifying the most crucial pathways. According to the results of the experiments, the proposed technique outperforms the most recent generation of DC technology.

Small delays can be detected using the ATPG's timing-aware SDF, which analyzes delay information to find transition faults across the longest pathways. When compared to standard transition fault test creation, the number of test patterns generated is substantially bigger. Due to the direct correlation between cost and number of patterns, we came up with an efficient way to reduce number of patterns without sacrificing the test's ability to detect minor delays. To identify timing-critical transition problems, ATPG is utilized in conjunction with an ATPG that is not timing aware.

REFERENCES

[1]. P. Alladi and S. Tragoudas, “Efficient selection of critical paths for delay defects in the presence of process variations,” in Proc. Int. Conf. Des. Technol. Integr. Syst. Nanosc. Era (DTIS), Apr. 2016, pp. 1–6.

[2]. S. Chakroborty and G. Saha, “Feature selection using singular value decomposition and QR factorization with column pivoting for textindependent speaker identification,” Speech Commun., vol. 52, no. 9, pp. 693–709, 2010.

[3]. K.-T. Chueng, S. Dey, M. Rodgers, and K. Roy, “Test challenges for deep sub-micron technologies,” in Proc. 37th Annu. Des. Automat. Conf., Jun. 2000, pp. 142–149.

[4]. J. Chung, J. Xiong, V. Zolotov, and J. A. Abraham, “Testabilitydriven statistical path selection,” IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., vol. 31, no. 8, pp. 1275–1287, Aug. 2012.

[5]. F. Firouzi, F. Ye, K. Chakrabarty, and M. B. Tahoori, “Aging- and variation-aware delay monitoring using representative critical path selection,” Trans. Des. Automat. Electron. Syst., vol. 20, no. 3, p. 39, 2015.

[6]. P. K. Javvaji and S. Tragoudas, “Efficient computation of the sensitization probability of a critical path considering process variations and path correlation,” in Proc. IEEE Int. Symp. Circuits Syst. (ISCAS), May 2017, pp. 1–4.

[7]. K. Baker, G. Gronthoud, M. Lousberg, I. Schanstra, and C. Hawkins, “Defect-based delay testing of resistive vias-contacts, a critical evaluation,” in Proc. IEEE Int. Test Conf., Sep. 1999, pp. 467–476.

[8]. M. A. Breuer, C. Gleason, and S. Gupta, “New validation and test problems for high performance deep sub-micro VLSI circuits,” in Proc. IEEE VLSI Test Symp., Apr. 1997, pp. 1–8.

[9]. Srinivas, L., & Umapathi, N. (2022, May). New realization of low area and high-performance Wallace tree multipliers using booth recoding unit. In *AIP Conference Proceedings* (Vol. 2393, No. 1, p. 020221). AIP Publishing LLC.

[10]. K.-T. Cheng, S. Dey, M. Rodgers, and K. Roy, “Test challenges for deep sub-micro technologies,” in Proc. ACM/IEEE Design Autom. Conf., Jun. 2000, pp. 142–149.

[11]. N. Umapathi, G. M. Krishna and L. Srinivas, "A Comprehensive Survey on Distinctive Implementations of Carry Select Adder," *2021 4th Biennial International Conference on Nascent Technologies in Engineering (ICNTE)*, 2021, pp. 1-5, doi: 10.1109/ICNTE51185.2021.9487718.

[12]. W. N. Li, S. M. Reddy, and S. K. Sahni, “On path selection in combinational logic circuits,” IEEE Trans. Comput.-Aided Des., vol. 8, no. 1, pp. 56–63, Jan. 1989.

[13]. X. Lin, K.-H. Tsai, C. Wang, M. Kassab, J. Rajski, T. Kobayashi, R. Klingenberg, Y. Sato, S. Hamada, and T. Aikyo, “Timing-Aware ATPG for High Quality At-speed Testing of Small Delay Defects,” ATS 2006, pp. 139-146.

[14]. S. Mitra, E. Volkerink, E.J. McCluskey, and S. Eichenberger, “Delay Defect Screening Using Process Monitor Structures,” Proc. VTS 2004, pp. 43-48.

[15]. K.S. Kim, S. Mitra, and P.G. Ryan, “Delay Defect Characteristics and Testing Strategies,” IEEE D&T of Computers, Sep.-Oct., 2003, pp. 16.

[16]. Murali Krishna G., Karthick G., Umapathi N. (2021) Design of Dynamic Comparator for Low-Power and High-Speed Applications. In: Kumar A., Mozar S. (eds) ICCCE 2020. Lecture Notes in Electrical Engineering, vol 698. Springer, Singapore. https://doi.org/10.1007/978-981-15-7961-5_110.

[17]. N. .Umapathi, G.L. 2020. Design and Implementation of Low Power 16x16 Multiplier using Dadda Algorithm and Optimized Full Adder. *International Journal of Advanced Science and Technology*. 29, 3 (Feb. 2020), 918 - 926.