A High Performance Scan Flip-Flop Design for Serial and Mixed Mode Scan Test

Satyadev Ahlawat*, Jaynarayan Tudu†, Anzhela Matrosova‡, and Virendra Singh*
*CADSL, Dept. of Electrical Engineering, Indian Institute of Technology Bombay, India
†Dept. of Computer Science and Automation, Indian Institute of Science Bangalore, India
‡Dept. of Applied Mathematics and Cybernatics, Tomsk State University, Russia
Email: satyadev@iitb.ac.in, jayttudu@csa.iisc.ernet.in, mau11@yandex.ru, viren@ee.iitb.ac.in

***Abstract*—Over the years, serial scan design has became the de-facto Design for Testability (*DFT*) technique. The ease of testing and high test coverage has made it to gain wide spread industrial acceptance. However, there are associated penalties with serial scan. These penalties include performance degradation, test data volume, test application time, and test power dissipation. The performance overhead of scan design is due to the scan multiplexers added to the inputs of every flip-flop. In today's very high speed designs with minimum possible combinational depth, the performance degradation caused by scan multiplexer has became magnified. Hence to maintain the circuit performance the timing overhead of scan design must be addressed.**

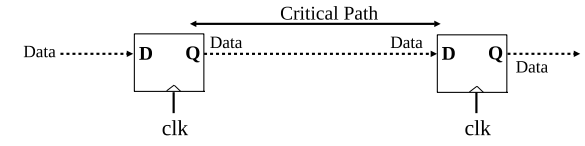
In this paper we propose a new scan flip-flop design that eliminates the performance overhead of serial scan. The proposed design removes the scan multiplexer off the functional path. The proposed design can help in improving the functional frequency of performance critical designs. Furthermore, the proposed design can be used as a common scan flip-flop in mixed mode scan test wherein it can be used as a serial scan cell as well as random access scan (*RAS*) cell.

I. Introduction

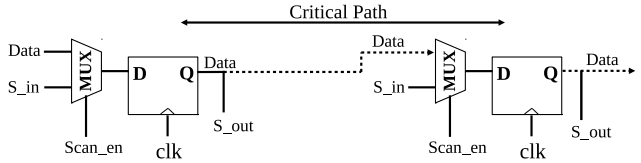
The testing of today's highly complex *SoC* designs is a big challenge faced by *VLSI* test community. Scan design is the only *DFT* approach that can effectively test a highly complex design with very high fault coverage. The objective of scan design is to achieve full controllability and observability of each and every flip-flop in the design. In a full scan design, each flip-flop is replaced by a scan flip-flop. A scan flip-flop is nothing but a muxed input master-slave based *D* type flip-flop. The scan multiplexer has two inputs: data input (*D*) and scan input (*SI*). The input selection is done using a control signal called *test_enable* (*TE*). In functional mode, data input is selected and the scan flip-flop function as a regular flip-flop. In test mode, scan input is selected and all the scan flip-flops are connected into one or more serial shift register(*s*). The serial shift register(*s*) are popularly known as scan chain(*s*). All flip-flops of the scan chain are loaded with desired data by consecutive application of the clock signal. The full scan design reduces the sequential test problem to combinational test problem.

Serial scan is obviously not free from drawbacks. There are some inherent penalties associated with serial scan. These penalties include: 1) performance overhead, 2) test data volume, 3) test power consumption, and 4) test application time. The performance overhead of serial scan is related to the scan multiplexer [1], [2]. The scan multiplexer falls into each clocked path and adds performance penalty of approximately two gate-delays. A circuit without scan design and with scan design are shown in Figure 1. As it is observable in Figure 1*a*, critical path of a sequential circuit without scan insertion is decided by the longest combinational path between two flip-flops. However, in a scan inserted sequential circuit (see Figure 1*b*) the same critical path is elongated by a scan multiplexer at the end of combinational path. Scan also adds an extra fanout at the output of flip-flop. Both of these factors increase the critical path delay and can reduce functional clock speed by 5 to 10% [1].

This makes it necessary to eliminate the performance overhead of scan multiplexer. Several solutions have been proposed to alleviate the performance penalty of scan design. One such solution that alleviates the performance overhead as well as other penalties associated with the serial scan design is the use of partial scan instead of full scan. In partial scan design, only a subset of all flip-flops in Circuit-Under-Test (*CUT*) are replaced by scan flip-flops to form a scan chain. This subset does not include flip-flops of the critical paths and hence reduces the performance penalty of scan. Additionally, the partial scan design techniques also reduce test data volume and test application time which are directly related to testing cost. However, the partial scan design techniques may reduce



(a) : Critical path of a sequential circuit



(b) : Critical path after scan insertion

Fig. 1. Scan design performance overhead [1]

978-1-5090-1507-8/16/$31.00 ©2016 IEEE

fault coverage of the *CUT*. The selection of flip-flops to be included into partial scan design can be testability measures based [3]–[6], structure based [7]–[10], or *ATPG* based [11]–[13]. The structure based techniques select flip-flops to cut-off the feedback path. These techniques make use of heuristics based on network topology for selecting a minimum set of flip-flops and do not explicitly analyze the circuit behavior. The *ATPG* based or test generation based techniques select flip-flops that are useful for test generation. These techniques first use sequential *ATPG* to generate test vectors for all possible faults. For the faults which remain undetectable, the culprit flip-flops are found and are included into partial scan design. These techniques are computationally intensive and are un-affordable. Most of the partial scan techniques require computationally demanding sequential *ATPG*, and can not be afforded with ever increasing circuit complexity. Furthermore, partial scan design techniques do not provide as high fault coverage as provided by full scan design, and are difficult to integrate into the existing industry design flow.

Random access scan (*RAS*) is an alternative *DFT* technique that can alleviate the problems associated with serial scan. Literature shows that *RAS* can greatly reduce test application time and test data volume along with test power reduction upto 99% [14]. However, the hardware overhead associated with *RAS* is prohibitively high. Furthermore, observability of storage cells and *RAS* architecture implementation are some other issues which need to be addressed properly [14]. Recently, mixed mode scan test has gained attention by the test community which try to exploit the best of both serial scan and *RAS* based *DFT* techniques [15]. In mixed mode scan test, part of the *CUT* is tested using serial scan and remaining part is tested using *RAS*. Mixed mode scan provides a trade-off between hardware overhead and inherent penalties of serial scan. However, mixed mode scan also faces some unresolved issues at implementation level. One of such issues is partitioning of the *CUT* between two *DFT* implementation, and un-availability of a single scan cell that can be used for both *DFT* structures.

This paper proposed a new scan flip-flop design that can be used as both a serial scan cell as well as a random access scan cell. The major advantages of the proposed scan flip-flop design are as follows:

1) The proposed design eliminates the performance penalty of serial scan by removing scan multiplexer off the functional path.
2) The proposed scan flip-flop can be used as a serial scan cell as well as a *RAS* cell, in mixed mode scan test. This can be very helpful in mixed mode scan design architecture implementation.
3) The proposed design does not use any extra control signal and has a minimal area overhead compared to conventional scan flip-flop.
4) The new scan flip-flop is capable of applying all sort of tests that can be applied with a conventional scan flip-flop. The proposed design fully comply with existing industry design and test flow.

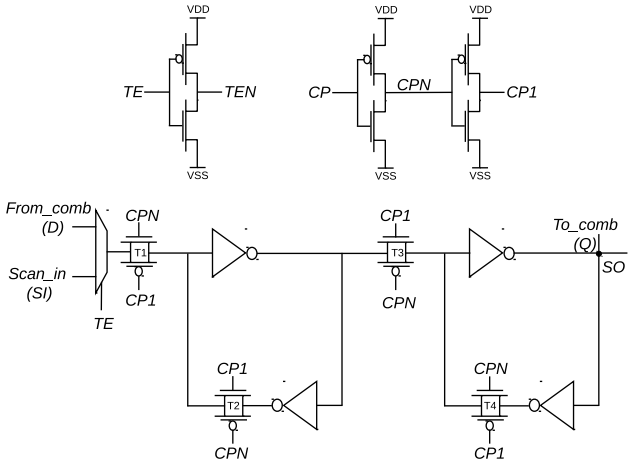


Fig. 2. A conventional scan flip-flop design

The remainder of the paper is organized as follows: Section II describes the conventional scan flip-flop and the proposed scan flip-flop design. Section II further elaborates on the details of test application process and retainment of test quality. Section III explains the possible use of proposed design in mixed mode scan design. Section IV explains the post layout timing results and verifies the efficacy of proposed design in eliminating the performance overhead of scan design. Section VI concludes the paper.

II. Preliminaries and Proposed Scan Flip-Flop

A large variety of scan flip-flop implementations are available in literature. A conventional scan flip-flop design is shown in Figure 2. This scan cell is a master slave latch based positive edge triggered muxed input *D* type flip-flop. The transmission gate $T1$, and the inverter pair connected back to back via transmission gate $T2$ forms the master latch. The slave latch comprises transmission gate $T3$ and the inverter pair connected back to back via transmission gate $T4$. The multiplexer at the input of master latch selects between functional input (*D*) or scan_input (*SI*) depending upon the value of test control signal *test_enable* (*TE*). In test mode, when *TE* is high (1), *SI* is selected and is connected to master latch's input. When the clock signal (*CP*) is low (0), the value of *SI* propagates to the master latch. In the mean time, slave latch retains the value from previous clock cycle. The value latched into the master propagates to slave latch when *CP* turns to high (1), and to the output *Q* of scan flip-flop. Similarly, when *test_enable* signal (*TE*) is set to 0, functional input *D* is selected and the circuit operates in functional mode.

A. Proposed Scan Flip-Flop Design

This section discusses the working of proposed scan flip-flop in different modes of operation. The proposed scan flip-flop schematic design is shown in Figure 3. Instead of a multiplexer at master latch's input, the proposed design uses a separate path for loading test vector values into the master

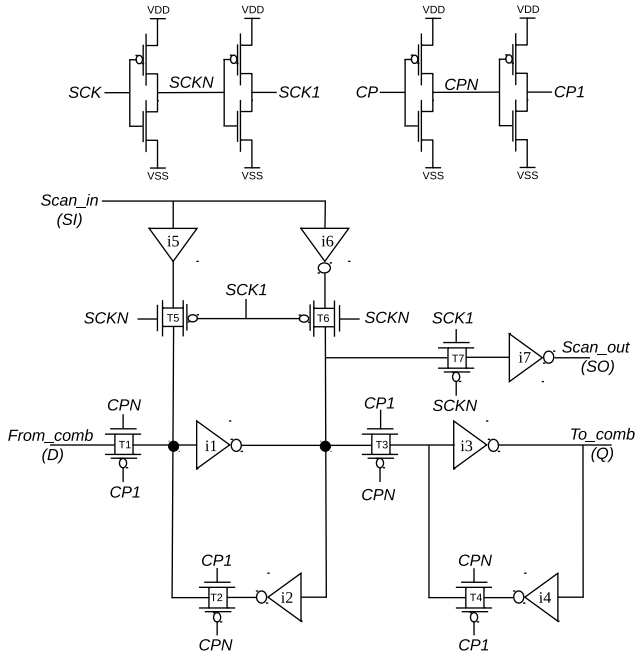


Fig. 3. Proposed scan flip-flop design

latch. Furthermore, the proposed scan flip-flop uses a low cost dynamic slave latch for shifting of test vectors in test mode. In functional mode, functional slave latch's output *Q* drives the combinational circuit inputs. The master latch of proposed scan flip-flop is formed by transmission gate $T1$, and inverter pair $(i1, i2)$ connected back to back via transmission gate $T2$. Similarly, the slave latch is formed by transmission gate $T3$, and inverter pair $(i3, i4)$ connected back to back via transmission gate $T4$. The dynamic slave latch comprises transmission gate $T7$ and inverter $i7$. The test mode path is formed by adding transmission gate $T5$, $T6$, buffer $i5$, and inverter $i6$ to the master latch structure. It should be noted that the extra gates added to master stage to form the test mode input path are not on the functional path. This extra circuitry remains disabled during functional mode and the proposed scan flip-flop acts as a regular flip-flop. The master latch and the slave latch are controlled by functional clock signal *CP*. The test mode input path is disabled by *test_enable* cum scan clock signal *SCK*. The details of working of the proposed design in different modes of operation is explained in the following subsections:

1) Functional mode: The proposed scan flip-flop works as a regular flip-flop in functional mode. In functional mode, scan clock signal *SCK* is kept at constant logic high (1) level. As long as *SCK* is at constant high (1) level the transmission gate $T5$, and $T6$ remain disabled. This disconnects the test mode input path from master structure and the proposed scan flip-flop functions as a regular flip-flop. The scan clock signal (*SCK*) held at constant high (1) level indicates functional mode operation. During the functional mode operation the transmission gate $T7$ always remains enabled. This keeps the dynamic slave latch always transparent during functional mode and makes the scan output (*SO*) toggle every time whenever there is a change in master latch's state. However, that is not of any concern as the scan output (*SO*) feeds the scan input path of successive scan flip-flop which remains disconnected from the master structure. In functional mode, the master latch gets it's input from functional data input *D*. When clock *CP* is low, the value of functional input *D* propagates into the functional master latch. When *CP* turns to high, the value latched into the master propagates to functional slave latch, and to output *Q* of the scan cell.

2) Test mode: While keeping the functional clock *CP* held at constant high (1) level, consecutive application of scan clock *SCK* makes the proposed scan flip-flop to function in test mode. As the functional clock *CP* is kept high (1), the transmission gate $T1$ always remains disabled in test mode. This disconnects the functional input *D* from the master latch. During test mode the master latch gets it's input from *scan_input SI*. The consecutive application of scan clock *SCK* loads the test values into the scan flip-flops. As can be observed in Figure 3, when *SCK* gets to logic low (0), $T5$ and $T6$ get enabled and the value of *SI* is written into the master latch in a similar way to memory write operation. It should be noted that in test mode, since *CP* is always high(1), the feedback path transmission gate $T2$ always remains enabled. This makes the master latch always trying to retain its previous value. However, it can be observed in Figure 3, the test mode input path circuit force writes the *SI* value simultaneously at both input and output nodes of inverter $i1$ via buffer $i5$ and inverter $i6$ respectively. This makes the write process very fast as far as logical fighting is concerned. When the scan clock *SCK* gets high (1), the dynamic slave latch transmission gate $T7$ gets enabled, and the master latch starts driving both dynamic slave latch inverter $i7$, and functional slave latch inverter $i3$. This propagates the test value latched into the master during negative clock cycle, to dynamic slave latch, and to *scan_output SO* of the scan cell.

When scan clock *SCK* gets to logic low (0), $T7$ gets disabled, and the input parasitic capacitance of inverter $i7$ drives the successive scan cell's *scan_input SI*. Due to very high impedance of the inverter, the parasitic capacitance do not discharge immediately and takes a long time. The parasitic capacitance discharge time decides the minimum scan clock frequency at which scan shifting can be done. However, a lower shift frequency is undesirable as it increases the test time, which in turn increases the test cost. It should be noted that in test mode the transmission gate $T3$ always remains enabled. This keeps the functional slave latch always transparent during test mode and makes the output (*Q*) toggle every time whenever there is a change in master latch's state. Every master latch in scan chain gets its scan input from preceding scan flip-flop's *SO* output, except the very first master latch in the scan chain which gets it's test input from a primary input pin. The scan output *SO* of the last flip-flop of the scan chain is connected to a primary output pin. The shifting of test vectors in scan chain is done using the

dynamic slave latch. Once the scan chain is loaded, test vector is launched via the functional slave latch. The test application process is elaborated in detail in the next section.

III. Application of Test Vectors

The proposed scan flip-flop allows to apply all kind of test vectors that can be applied using a conventional scan flip-flop. Before applying any kind of test vectors, scan chain integrity is verified by exercising scan flush test. Scan flush test is applied by propagating an all transition pattern, like 1100, through the scan chain without any response capture cycle in between. The scan clock *SCK* is always kept high during functional mode. When functional clock *CP* is high (1), falling edge on *SCK* switches the circuit from functional mode to test or scan mode. The functional clock *CP* is always kept high (1) during scan shift operation. On arrival of negative edge of *SCK*, the value of *SI* propagates into master latch via test input path. Next, the rising edge on *SCK* transfers the master latch value to dynamic slave latch and to the scan output node *SO*. By repetitive application of scan clock the flush pattern is propagated through the scan chain and is observed at primary output pin. The observation of correct input sequence at primary output pin verifies the integrity of scan chain. The scan integrity test covers all possible faults on the test mode input path and the input/output faults of dynamic slave latch as well as master latch. The scan integrity test does not cover input/output faults of functional slave latch. As we will see in the next subsection, these faults are covered during stuck-at fault test application.

A. Stuck-at-fault test

When clock *CP* is high (1), falling edge on scan clock *SCK* indicates the start of scan shifting. The *stuck-at-fault* test is applied by first loading the test vector via scan shifting path and then launching the test vector via functional slave latches. As explained earlier the functional slave latch always remains transparent during scan shifting process. So during the last *shift-cum-launch* cycle when negative edge at scan clock *SCK* comes, the test vector is applied via output *Q* of functional slave latches. It should be noted that the test vector is launched in last shift cycle at negative edge of the scan clock *SCK*. After launch of test vector, the scan clock *SCK* is kept high (1) to disable the scan input path. Now, to capture the test response the functional clock *CP* is clocked once. When the functional clock *CP* gets low, functional response is latched into the master latch via functional input *D*. On arrival of positive edge on the functional clock *CP*, response is propagated into the functional slave latch as well as into the dynamic slave latch. Once the test response is captured, functional clock *CK* is kept at logic high (1) level. This disconnects the functional input *D* from the master latch. Now, falling edge on scan clock *SCK* switches the circuit operation from capture to shift mode. At the same negative edge on *SCK*, functional response stored into slave latch gets transferred to functional master latch of next scan cell via the scan input path. The unloading of test response is done with simultaneous loading of next test vector.

The timing diagram for *stuck-at-fault* test application is shown in Figure 4. As stated earlier, input/output faults of the functional slave latch are not covered by the scan chain integrity test. These faults are covered by *stuck-at-fault* test, as the test vectors are launched via functional slave latch. So, these faults if there exist any will manifest during the test vector launch cycle. The input/output faults of functional master latch will cause wrong test vector launch, which in turn will result into wrong test vector response. This way these faults will be detected at the end of test response unloading process. Since, the same set of test vectors are shifted and applied during *stuck-at-fault* test, the same set of faults are covered as in conventional scan design based test process.

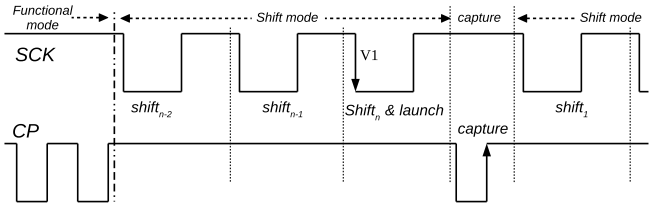


Fig. 4. Launch and capture in *stuck-at* test

B. Launch-on-capture test

In *launch-on-capture* (*LOC*) testing, a test vector pair ($V1$, $V2$) is applied to the *CUT*. The first vector $V1$ initializes the circuit and the second vector $V2$ launches the transition. The launch of initialization vector $V1$ and first response capture is performed in a way similar to *stuck-at-fault* test. As we know that vector $V2$ is the functional response of vector $V1$ so, response capture of $V1$ also acts as launch of transition vector $V2$. The response of vector $V2$ is captured by applying an *at-speed* functional clock cycle. The loading/unloading of test/response is done in a way similar to *stuck-at-fault* testing. The timing diagram for the *Launch-on-capture* test application is shown in Figure 5.

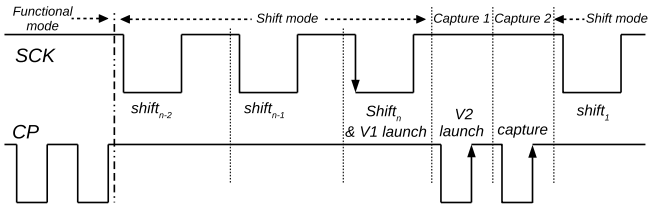


Fig. 5. Launch of $V1$, $V2$, and capture in *LOC* test

C. Launch-on-shift test

In *launch-on-shift* (*LOS*) testing, transition vector $V2$ is one bit shift of initialization vector $V1$. In *LOS* test, response of $V1$ is not captured. As explained earlier $V1$ is launched at negative edge of scan clock *SCK*. As $V2$ is one bit shift of $V1$, $V2$ is launched at next negative edge of *SCK*. Now to capture the response of $V2$ *at-speed*, the scan clock *SCK* needs to be

clocked at functional clock speed. To apply *LOS*, *SCK* must be timing closed. However it should be noted that *SCK* in the proposed design is an exact equivalent of *test_enable* signal *TE* in conventional scan flip-flop. In order to make *SCK* timing closed, that is also a global signal like the functional clock signal *CK*, it needs to be synthesized like a clock tree. That is a very costly task. Due to the high cost associated with *LOS* capable scan design, it is rarely exercised in industry. Application of *LOS* test even in conventional scan design is not possible without a timing closed *TE* signal. The *LOS* based transition delay fault test, without a fast *SCK* (or timing closed *TE* in case of conventional scan design) can be exercised by using the *AND-OR-INVERT* (*AOI*) circuitry proposed by Gefu et al. [16], [17]. Hence, neither conventional nor proposed design have the capability of exercising *LOS* based delay test. The proposed design does not use any extra control signal or testing constraint, and can be easily integrated into the existing *DFT* flow.

IV. Mixed mode scan design

In a mixed mode scan test, part of the *CUT* is tested using serial scan and rest of the circuit is tested using *RAS* based test [15]. The memory elements that are included into serial scan architecture are replaced by a serial scan flip-flop. The memory elements that are included into *RAS* test architecture are replaced by *RAS* cell. The proposed scan design eliminates the requirement of two separate scan cell libraries for serial scan cell and *RAS* cell. It provides a common scan cell that can be used for both serial scan and *RAS*. Schematic design of an area and performance efficient progressive random access scan cell proposed by Baik et al. [14] is shown in Figure 6. This *PRAS* cell is a modified design of a regular master slave based positive edge triggered *D* type flip-flop. The grey part

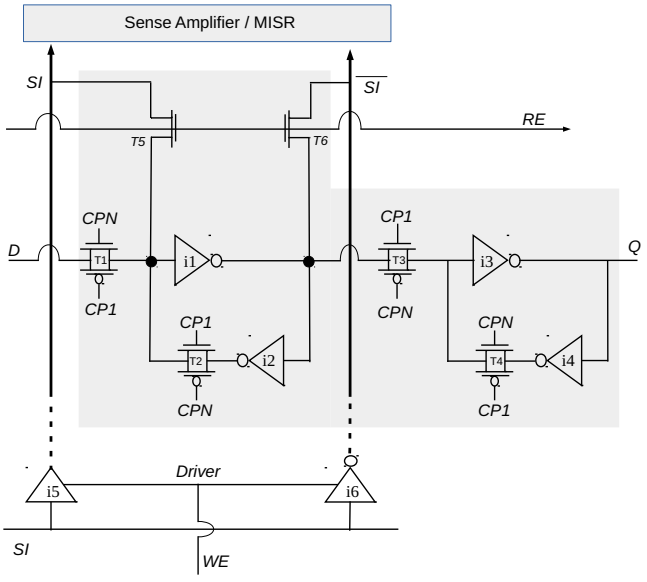


Fig. 6. Progressive Random Access Scan (*PRAS*) Cell [14]

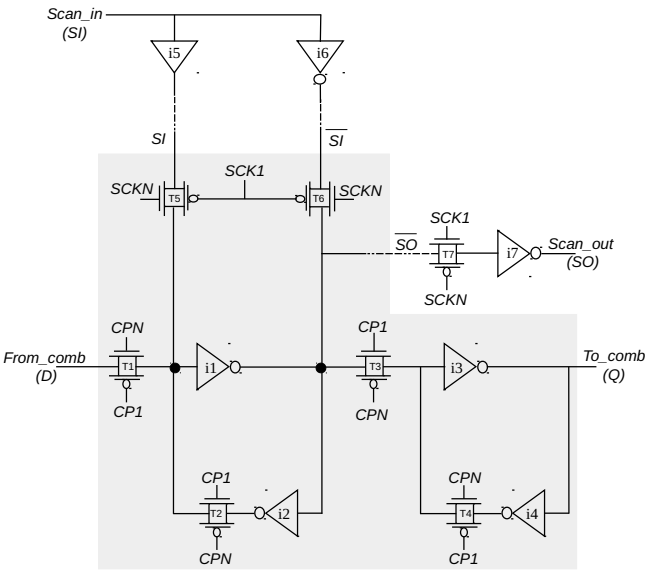


Fig. 7. Proposed design as a base scan cell for mixed mode scan test

in Figure 6 depicts the *PRAS* cell and the remaining circuit is part of *RAS* test architecture. Figure 7 represents the proposed flip-flop as a base scan cell. The grey box of the proposed flip-flop maps to the basic *PRAS* cell design except the access pass transistors are replaced by transmission gates ($T5$, $T6$). Both of these basic cells are functionally equivalent. Therefore, same basic cell of the proposed design can be used as a *PRAS* cell without modification. Note that the proposed scan cell has one extra output node *SO* which remains unconnected in *RAS* mode. The base scan cell can be synthesized as a *PRAS* cell by mapping the base scan cell in/out signals with corresponding *PRAS* in/out signals to use in *RAS* test architecture. Similarly, the base scan cell can be synthesized as serial scan cell with the full logic shown in Figure 7. It is worth to note that with a minor change in synthesis process the proposed scan flip-flop can be synthesized as both serial scan cell and *PRAS* cell. This can make the mixed mode scan architecture implementation efficient and easy.

V. Experimental Results

The post layout timing simulation of the proposed scan flip-flop has been carried out using *UMC's* $65nm$ technology at operating voltage of $1.2V$ for frequencies ranging from $500MHz$ to $1GHz$. The post layout timing diagram at a clock frequency of $500MHz$ has been put for reference in Figure 8. The timing diagram verifies the efficacy of the proposed design. The post layout timing simulation results are listed in Table I. In functional mode, clock to *Q* (t_{cq}) delay, and setup time (t_{setup}) of the proposed design are found to be $378ps$, and $290ps$ respectively. It can be observed from Table I, that t_{cq} of the proposed flip-flop is slightly higher than t_{cq} of the conventional flip-flop. This is due to the extra capacitive loading caused by dynamic slave latch. There is a

TABLE I
POST LAYOUT TIMING SIMULATION RESULTS AT $500MHz$

Functional Mode					
Scan cell ↓ Parameter →	*Clock to Q* (t_{cq})	*Setup time* (t_{setup})	*Hold time* (t_{hold})	t_{cq} + t_{setup} = t_{pd}	*Time gain w.r.t Conventional*
Conventional Scan Flip-Flop	$0.332ns$	$0.400ns$	$0.0ns$	$0.732ns$	$+64ps$
Proposed Scan Flip-Flop	$0.378ns$	$0.290ns$	$0.0ns$	$0.668ns$	
Test Mode					
Convential Scan Flip-Flop	$0.332ns$	$0.400ns$	$0.0ns$	$0.732ns$	$-103ps$
Proposed Scan Flip-Flop	$0.284ns$	$0.545ns$	$0.0ns$	$0.829ns$	

considerable decrease in t_{setup} of the proposed scan flip-flop due to elimination of input multiplexer. Overall the proposed design offers a time saving of $64ps$.

In test mode, propagation delay (t_{pd}) of the proposed scan flip-flop degrades by approximately $103ps$. However, test mode performance degradation is not of any concern as the test shifting is done at a frequency much lower than the functional frequency. Overall the time saving offered by proposed design is approximately equal to 3-4 inverters delay, where typical inverter delay in $65nm$ technology is approximately 15-18ps.

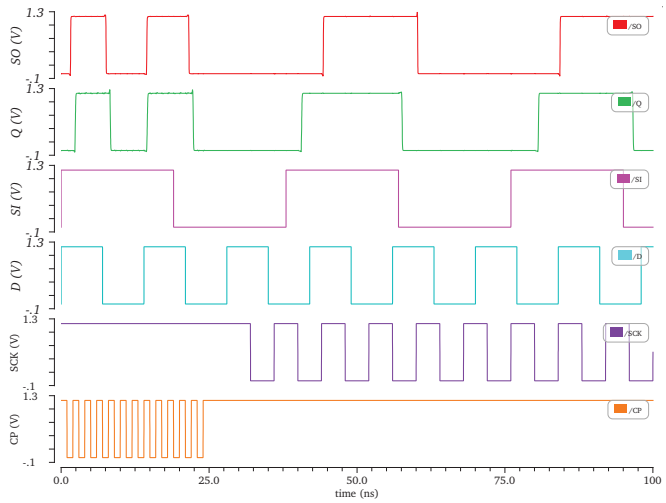


Fig. 8. Post layout timing waveform at 500MHz

VI. CONCLUSION

We have proposed a scan flip-flop design which eliminates the performance penalty of serial scan by removing scan multiplexer off the functional path. The proposed design uses a separate test mode input path and a low cost dynamic slave latch to scan shift the test vectors. The new scan flip-flop is capable of applying all sort of conventional tests and fully comply with the conventional industry design and test flow. Furthermore, the proposed scan flip-flop can be used as a serial scan cell as well as a *RAS* cell, in mixed mode scan test. This can be very helpful in mixed mode scan design architecture implementation. The proposed design does not use any extra control signal and has a minimal area overhead compared to conventional scan flip-flop.

REFERENCES

[1] M. L. Bushnell and V. D. Agrawal, *Essentials of Electronic Testing for Digital, Memory and Mixed-Signal VLSI Circuits*. New York, USA: Kluwer Academic Publishers, 2000.

[2] S. Ahlawat, J. T. Tudu, A. Matrosova, and V. Singh, "A new scan flip flop design to eliminate performance penalty of scan," in *24th IEEE Asian Test Symposium, ATS 2015, Mumbai, India, November 22-25*, 2015, pp. 25–30.

[3] P. Kalla and M. J. Ciesielski, "A comprehensive approach to the partial scan problem using implicit state enumeration," in *Proceedings of the IEEE International Test Conference*, ser. ITC '98, Washington, DC, USA, 1998, pp. 651–657.

[4] D. Xiang, S. Venkataraman, W. K. Fuchs, and J. H. Patel, "Partial scan design based on circuit state information," in *In Proc. of the Design Automation Conf., Las Vegas, NV*, 1996, pp. 807–812.

[5] V. Boppana and W. K. Fuchs, "Partial scan design based on state transition modeling," in *Proc. Int. Test Conf*, 1996, pp. 538–547.

[6] D. Xiang and J. H. Patel, "Partial scan design based on circuit state information and functional analysis," *IEEE Trans. Comput.*, vol. 53, pp. 276–287, March 2004.

[7] K. T. Cheng and V. D. Agrawal, "A partial scan method for sequential circuits with feedback," *IEEE Trans. Comput.*, vol. 39, pp. 544–548, April 1990.

[8] A. Kunzmann and J. H. Wunderlich, "An analytical approach to the partial scan problem," *Journal of Electronic Testing: Theory and Applications*, vol. 1, pp. 163–174, 1990.

[9] P. Ashar and S. Malik, "Implicit computation of minimum-cost feedback-vertex sets for partial scan and other applications," in *Proceedings of the 31st annual Design Automation Conference*, ser. DAC '94. New York, NY, USA: ACM, 1994, pp. 77–80.

[10] S. T. Chakradhar, A. Balakrishnan, and V. D. Agrawal, "An exact algorithm for selecting partial scan flip-flops," in *Proceedings of the 31st annual Design Automation Conference*, ser. DAC '94. New York, NY, USA: ACM, 1994, pp. 81–86.

[11] V. D. Agrawal, K. T. Cheng, D. D. Johnson, and T. Sheng Lin, "Designing circuits with partial scan," *IEEE Des. Test*, vol. 5, pp. 8–15, March 1988.

[12] X. Lin, I. Pomeranz, and S. M. Reddy, "Full scan fault coverage with partial scan," in *Proceedings of the conference on Design, automation and test in Europe*, ser. DATE '99. New York, NY, USA: ACM, 1999.

[13] M. S. Hsiao, G. S. Saund, E. M. Rudnick, and J. H. Patel, "Partial scan selection based on dynamic reachability and observability information," in *Proceedings of the Eleventh International Conference on VLSI Design*, ser. VLSID '98, Washington, DC, USA, 1998, pp. 174–.

[14] D. H. Baik and K. K. Saluja, "Progressive random access scan: a simultaneous solution to test power, test data volume and test time," in *Proceedings of the IEEE International Test Conference, ITC 2005*, Nov 2005, pp. 10 pp.–368.

[15] J. T. Tudu, "JSCAN: a joint-scan dft architecture to minimize test time, pattern volume, and power," in *20th International Symposium on VLSI Design and Test, VDAT 2016, Guwahati, India, May 24-27*, 2016 [In Press].

[16] G. Xu and A. D. Singh, "Achieving high transition delay fault coverage with partial DTSFF scan chains," in *IEEE International Test Conference, ITC 2007, Santa Clara, California, USA, October 21-26*, 2007, pp. 1–9.

[17] A. Mishra, N. Sinha, Satdev, V. Singh, S. Chakravarty, and A. D. Singh, "Modified scan flip-flop for low power testing," in *Proceedings of the 19th IEEE Asian Test Symposium, ATS 2010, 1-4 December 2010, Shanghai, China*, 2010, pp. 367–370.