Novel DFTs for Circuit Initialization to Reduce Functional Fmax Test Time

Ujjwal Guin[1], Tapan Chakraborty[2], and Mohammed Tehranipoor[1]

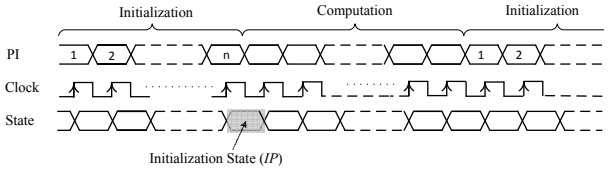
[1]ECE Dept., University of Connecticut [2]Qualcomm
{ujjwal, tehrani}@engr.uconn.edu tapanc@qualcomm.com

***Abstract*—Using functional test for F_{max} analysis is still an effective method used in practice in spite of the fact that the test cost associated with functional F_{max} test remains to be a major problem. In this paper, we develop novel design-for-testability (DFT) structures to considerably reduce the cost of initializing the circuit during functional test.**

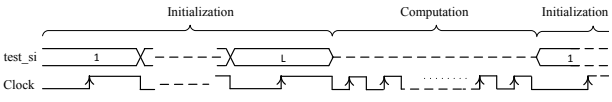
I. Introduction

Over the past decade, there has been a major shift toward using structural tests to reduce the overall test cost. Structural tests are now used for stuck-at faults, open faults, bridging faults, transition delay faults and path delay faults. In recent years, several research attempts have been made to estimate the functional F_{max} using structural tests [1] [2] [3] [4] [5]. However, due to the increased noise and crosstalk effects in scan-based testing, the inaccuracy of performance verification using path and transition delay faults have kept functional F_{max} test a viable approach for many semiconductor companies. The major issues associated with functional test include: (i) difficulty of generating effective test patterns targeting long paths in the circuit, (ii) increased test cost as it may contribute up to 30-40% of the total production test cost. Further analyzing the cost of functional F_{max} test shows that the initialization phase of the test contributes significantly to the total test time, and (iii) the requirement for a high-speed tester in order to apply functional test patterns to the circuit under test.

Today's complex designs are highly modular and consist of multiple modules. In functional mode, these modules work inter-connectedly to produce the functional response. Let us consider for an example of a microprocessor. It consists of a core, floating point unit, memory, dram controller, cache controller etc. To test the floating point unit, we must initialize the core first. Thus, a set of functional test vectors are required to test a module while other modules help to produce to the correct inputs to the module being tested and pass the response to the primary outputs. To do so, each module has to be initialized properly or it can be stated that the finite state machine corresponding to that module has to be set to a desired functionally valid state. We represent this valid state as *initialization state* (IS) or *initialization pattern* (IP). Setting the initialization state of a complex industrial circuit through primary inputs becomes a major challenge and could potentially take millions of clock cycles depending on the size and type of the circuit [6]. Moreover, it leads to the waste of resources to write the test patterns by functional test engineers. On the other hand, design for testability (DFT) based on scan and automatic test pattern generation (ATPG) has been widely accepted in industry to obtain high test coverage for structural test. Shifting the initialization state through DFT structure into the scan chains is a possible approach. Figure 1(a) shows the functional testing process to initialize the circuit through primary inputs (*PIs*). The circuit is set to the initialization state after n functional clock cycles (initialization cycles) and the corresponding initialization time is referred as T_I. Figure 1(b) shows the initialization of the circuit through DFT structure. The *IP* is shifted into the scan chains through the test inputs ($test\_si$). Then the circuit is set to functional mode and the computation phase starts. This initialization time is referred as DFT-based initialization time (T_{DFTI}).



(a) Timing waveforms for functional testing without using DFT.



(b) Timing waveforms for functional testing using DFT.

Figure 1. Timing waveforms for functional testing.

Today, almost every industrial circuit comes with compression as a part of DFT to compress the test data and thus reduce test time and cost. Compression is basically either in the form of linear feedback shift register (LFSR) based [7] [8] or adaptive scan based [9]. The standard compression technique relies entirely on the high percentage of don't-cares (Xs) in the test cubes. The compression scheme fails to find a solution when the test cube is dominated by care bits [8]. The problem with using compression in the initialization phase is that the *IP* is completely specified and there are no don't-care bits in it.

Currently, there is not enough research carried out to address the initialization during functional testing. In [6], the authors

mention the initialization for functional testing using full-hold scan system. They used DFT to initialize the design. In their approach, the design is loaded with all 0s during reset pin assertion and then a scan store operation is executed to initialize the state of 200K nonarray sequential elements in the design to 0.

In this paper, we will describe two different approaches using the existing DFT structure to expedite the initialization phase. We have achieved the *Best Case* and an intermediate value between the *Best Case* and *Worst Case* initialization time, where,

- *Best Case*: T_{DFTI} is L shift clock cycles. This is the minimum number of clock cycles required to shift in an IP into the scan chains.
- *Worst Case*: The regular DFT compression is bypassed and T_{DFTI} is kL shift clock cycles. In this case N bits are shifted into N scan chains directly from the tester. The process of shifting an IP into the scan chains through N pins will be repeated $k = M/N$ times.

where, L, N, M, and k denote scan length, # tester pins, # internal scan chains, and compression ratio respectively.

This paper will be organized as follows: in Section II, we will present the underlying principles of the pattern dependent approach and its architecture. In Section III, we will describe the theory and architecture of pattern independent approach. In Section IV, we will present the simulation results. We will conclude with final remarks in Section V.

II. Pattern Dependent Approach

In pattern dependent (PD) approach, the fixed initialization pattern (IP) is applied during the initialization of the design.

A. PD Architecture

Figure 2 describes the proposed flow for the PD architecture. It starts with the random selection of few bits from the IP, generally less than 5% of the total bits present in it and assumes the rest of the bits are don't-cares. Each selection bit represents a distinct equation which results in a set of linear equations. The objective is to find a solution set rather than a single solution from that set of equations. If the number of independent variables is greater than the number of equations, we will get a solution set. The detailed process of forming the set of linear equations and finding a solution is described in [7] [8]. Then, we apply these solutions from the set one by one to the decompressor to generate the pattern. The generated pattern and the IP are compared bit-wise to find the conflicts. Here, a conflict represents a mismatch of the value of a particular bit position of the generated pattern with the IP. Finally, we select the solution from the set that gives the least conflicts. The complexity of this approach is $O(p)$ where p is the number of solutions present in the solution set.

By flipping these conflicting bits, the generated pattern and IP will match exactly and our objective will be fulfilled to generate the fully specified IP. The XOR function can satisfy our requirement of flipping the bits. Thus, we have introduced an additional XOR network in front of the internal scan chains

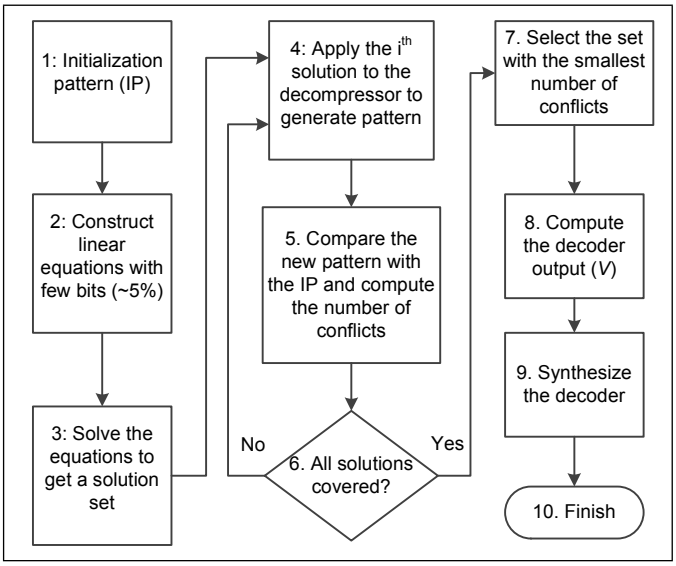


Figure 2. PD approach.

to select the output or the inverted output of the decompressor. If there is a conflict, the other input of the XOR gate will be "1" and the decompressor output will be inverted. To generate the other input of XOR network, we have introduced a custom decoder with output V where, $V = [V_1\ V_2\ \ldots\ V_L]$. V_i $(0 \le i \le L)$ represents the decoder output at i^{th} clock cycle. V_i can be represented as $V_i = [v_1\ v_2\ \ldots\ v_M]^T$ where,

$$v_j = \begin{cases} 1, & \text{if there is a conflict} \\ 0, & \text{otherwise} \end{cases}$$

Figure 3 presents the proposed decoder-based architecture for the initialization during functional testing. The compressed test data, i.e., the solution corresponding to the least conflicts obtained before, are applied to test pins (*test_si*). This design needs one extra pin *IN_EN* from the tester to enable the initialization phase. The number of inputs of the decoder is $log_2(L)$ and is generated by a modulo L counter which is driven by the same shift clock. The counter will be initialized to zero when $IN\_EN = 0$. The decoder output will always be zero when $IN\_EN = 0$, i.e., except for the initialization phases.

III. Pattern Independent Approach

The idea of developing a structure utilizing compression to shift any initialization pattern during functional testing is more promising. However, after studying the constraints from the decompressor, we have concluded that it is not possible to construct a fully specified (all care bits) test cube, i.e., the initialization pattern using decompressor. If we bypass the decompressor and shift the initialization pattern through existing DFT structure we can initialize the design to any functionally valid state. However, the initialization time will increase to *Worst Case* initialization time.

The core idea of the pattern independent approach is based on the slow shift-in process for the structural testing. Generally, the functional frequency (f_f) is much faster than the shift-in frequency (f_s). Thus, the initialization through a slow shift-

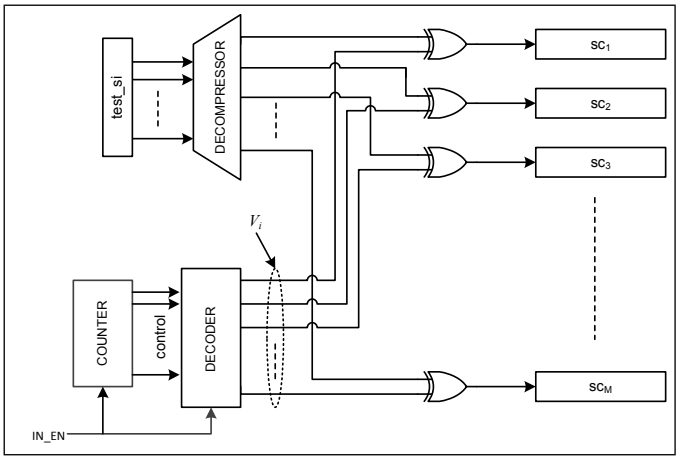


Figure 3. PD architecture.

in frequency results in increased initialization time. However, we simply cannot increase the shift-in frequency due to the switching power dissipation in the scan chain. Our objective will be fulfilled if we accept the high speed data from the tester and load the data into the scan chains with the slower shift-in clock. This conversion can be done by a serial-in parallel-out shift register. The register will accept the high speed data and load at a much slower rate.

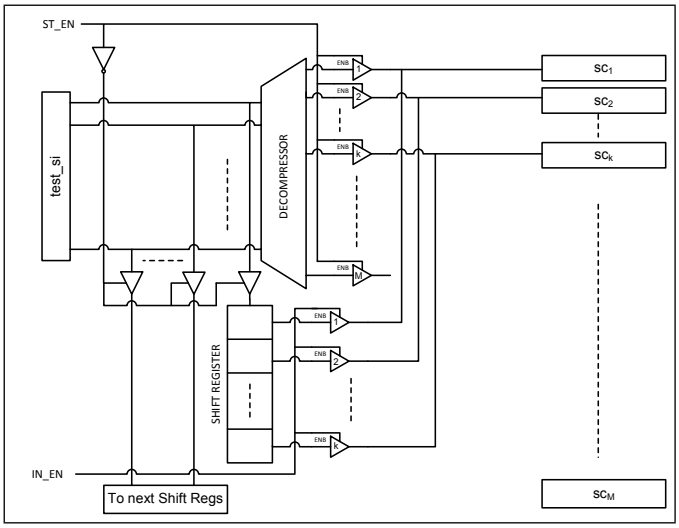


Figure 4. PI architecture.

Figure 4 presents the PI architecture. It bypasses the decompressor completely. The decompressor output is connected to a tri-state buffer network (*network* 1) that is controlled by *ST_EN* pin. This pin will always be at logic zero during the functional testing. Each *test_si* input is connected to a k bit serial-in parallel-out shift register. This shift register runs at functional clock frequency and loads the stored value to k scan chains at a rate of f_f/k. The output of this shift register is connected to another tri-state buffer network (*network* 2) controlled by *IN_EN*. The output of *network* 1 and *network* 2 are connected together. During the initialization phase the output of *network* 2 is sent to the internal scan chains while the output of *network* 1 is sent to the scan chains during computation phase. To eliminate switching power dissipation during structural testing, another buffer network controlled by the inverted *ST_EN* has been added in front of the shift registers.

This design works as follows: an internal storage unit (shown as shift register in Figure 4) accepts the high speed data from the tester and sends them to the scan chain at a slower rate. If $k > f_f/f_s$, then the shift-in clock must be slowed down to match the capture and load data rate. As a result, the new shift-in frequency will be, $f_s^{new} = f_f/k$. Again, if $k < f_f/f_s$, then the functional clock must be slowed down to match the capture and load data rate which needs the functional frequency during initialization to be, $f_f^{new} = kf_s$.

The initialization time will be,

$$= \begin{cases} L \text{ shift clock cycles}, & \text{when } k \leq f_f/f_s \\ kL(f_s/f_f) \text{ shift clock cycles}, & \text{when } k > f_f/f_s \end{cases}$$

IV. Simulation Results

The proposed initialization schemes were implemented on the *ITC'99* *b19* benchmark which has 6042 flip-flops and 53373 gates. All simulations were carried out in the Synopsys environment [10]. We used 90nm technology to implement our proposed architectures.

A. Pattern Dependent Approach

The PD approach tries to find the minimum size decoder by selecting the solution that leads to the minimum number of conflicts. We have selected LFSR-based compression architecture and simulated it using C/C++. The XOR equations corresponding to the selection bits are solved using prolog solver [11]. Each solution from the solution set obtained from the solver was applied to the compression architecture and conflicts were computed using C/C++. We have selected the minimum conflict solution to minimize the decoder area. The area overhead versus compression ratio is shown in Figure 5. It shows that the area overhead decreases linearly with the increase of compression ratio (except for $k < 5$). With the increase of compression ratio, the length of the scan chain becomes shorter for a specific design and the number of decoder inputs ($log_2(L)$) become lesser. With a compression ratio of 30, the area overhead due to the decoder is around 0.63% where as it is 0.91% with a compression ratio of 5.

B. Pattern Independent Approach

The design described in Figure 4 has a k-stage serial-in parallel-out shift register for each tester pin. It is intuitive that the area overhead will not change with the compression. The area overhead is one shift register stage and three buffers per scan chain. However, the area overhead will vary with the length of the scan chain as shown in Figure 6. The area overhead decreases with the increase of scan length. We have simulated the design with a varying scan length (L) of 100 to 250 as in the case for modern industrial designs. The area overhead is around 0.22% for a scan length of 250.

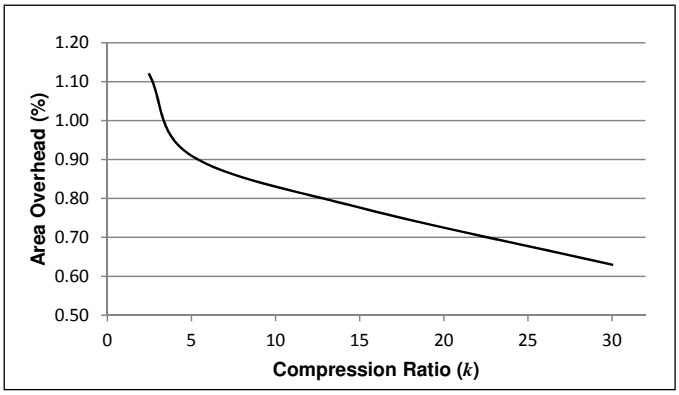


Figure 5. Compression ratio vs area overhead analysis for PD architecture.

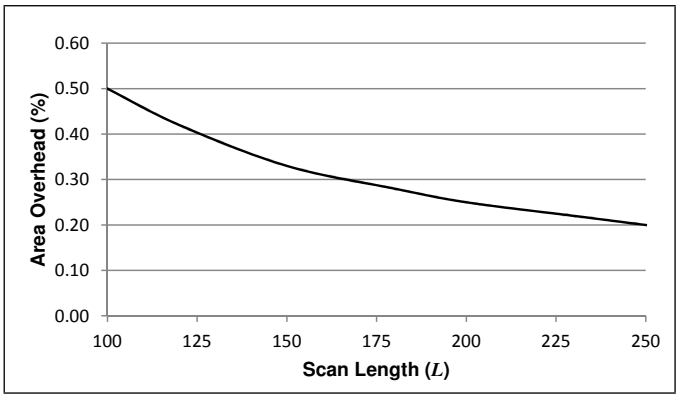


Figure 6. Scan length vs area overhead analysis for PI architecture.

C. Comparison Between PD and PI Architectures

The comparisons between the two designs are shown in Table I. We have calculated DFT-based initialization time (T_{DFTI}), area/power overhead, and test data (bits for one *IP*) for comparison. The initialization time for PD architecture is L shift clock cycles whereas PI architecture provides an initialization time of $kL(f_s/f_f)$ cycles (worst case). The area overhead is extremely small for the PI architecture (0.22%) whereas it is close to 1% for PD architecture. However, we must keep in mind that PD architecture can generate only one *IP*. The power overhead is insignificant (<0.3%) for PD architecture. However, there is an increase in power consumption for PI architecture and is approximately 5%. The test data required for the PD and PI architectures are LM/k and LM bits respectively.

Table I
COMPARISONS WITH DIFFERENT ARCHITECTURES

Architecture	T_{DFTI} (# Scan Clock Cycles)	Area Overhead (%)	Power Overhead (%)	Test Data (Bits)
PD	L	$\approx$1	<0.3	LM/k
PI	$kL(f_s/f_f)$	0.22	5.2	LM

Table II presents the improvement for functional F_{max} test with different initialization cycles. We have assumed that the time required for the initialization and computation phases are equal during functional testing. The time for the computation phase remains constant whether or not we use DFT for initialization. Column 2 and 3 represent T_{DFTI} with a single initialization phase for PD and PI architectures. The test parameters, i.e., f_f, f_s, L, and k, are 200 MHz, 40 MHz, 200 and 30, respectively. Columns 4 and 5 show the percentage improvement for the functional F_{max} test for PD and PI architectures, respectively. We have calculated the improvement using the formula, $\left(\frac{T_{Fmax}-T_{Fmax}^{DFT}}{T_{Fmax}}\right)$ where T_{Fmax} and T_{Fmax}^{DFT} are the F_{max} test time without and with DFT, respectively. The improvement becomes greater with higher initialization cycles and eventually reaches to the ratio of initialization to the total test time i.e., 50% for our simulation scenario.

Table II
FUNCTIONAL F_{max} TEST TIME IMPROVEMENT

# *IN* cycles (n)	T_{DFTI} (sec)		Test time improvement (%)	
	PD architecture	PI architecture	PD architecture	PI architecture
1K	$5x10^{-6}$	$3x10^{-5}$	0.00	-250.00
10K	$5x10^{-6}$	$3x10^{-5}$	45.00	20.00
30K	$5x10^{-6}$	$3x10^{-5}$	48.33	40.00
1M	$5x10^{-6}$	$3x10^{-5}$	49.95	49.70
10M	$5x10^{-6}$	$3x10^{-5}$	50.00	49.97

V. CONCLUSION

In this paper, we have presented two novel initialization schemes for the functional testing to reduce overall test time. The proposed procedures take into account area and power overhead during initialization. Both pattern dependent and independent schemes are equally significant in different design scenarios where power or area is concerned.

REFERENCES

[1] B. Cory, R. Kapur, and B. Underwood, "Speed binning with path delay test in 150-nm technology," *Design Test of Computers, IEEE*, vol. 20, no. 5, pp. 41 – 45, Sept.-Oct. 2003.

[2] B. Kruseman, A. Majhi, and G. Gronthoud, "On performance testing with path delay patterns," in *Proc. IEEE VLSI Test Symposium*, May 2007, pp. 29 –34.

[3] P. Bastani, K. Killpack, L.-C. Wang, and E. Chiprout, "Speedpath prediction based on learning from a small set of examples," in *Proc. IEEE-ACM Design Automation Conference*, June 2008, pp. 217 –222.

[4] J. Chen, J. Zeng, L.-C. Wang, and M. Mateja, "Correlating system test fmax with structural test fmax and process monitoring measurements," in *Proc. on Asia and South Pacific Design Automation Conference (ASP-DAC)*, Jan. 2010, pp. 419 –424.

[5] J. Chen, J. Zeng, L.-C. Wang, J. Rearick, and M. Mateja, "Selecting the most relevant structural fmax for system fmax correlation," in *Proc. IEEE VLSI Test Symposium*, Apr. 2010, pp. 99 –104.

[6] R. Kuppuswamy, P. DesRosier, D. Feltham, R. Sheikh, and P. Thadikaran, "Full Hold-Scan Systems in Microprocessors: Cost/Benefit Analysis," *Intel Technical Journal*, vol. 8, no. 1, pp. 63–72, Feb. 2004.

[7] B. Koenemann, "LFSR-coded test pattern for scan designs," in *Proc. European Design and Test Conference*, 1991, pp. 237 –242.

[8] J. Rajski, J. Tyszer, M. Kassab, and N. Mukherjee, "Embedded Deterministic Test," *IEEE Transactions on CAD*, vol. 23, no. 5, pp. 776–792, May 2004.

[9] A. Chandra, R. Kapur, and Y. Kanzawa, "Scalable Adaptive Scan (SAS)," in *Proc. of the conference on Design, automation and test in Europe*, Apr. 2009, pp. 1476–1481.

[10] "Synopsys Inc. SOLD Y-2007, Volumes 1, 2 and 3," 2007, http://www.synopsys.com/Systems/BlockDesign/Pages/default.aspx.

[11] "B-Prolog, Version 7.7," http://www.probp.com/.