2023 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION EUROPE 10 YEAR ANNIVERSARY

Initialization Techniques for D-Flip-Flop Based Scan Chain with Scan Flush Architecture in Gate-Level Simulation

Wei Jun Yeap, Programmable Solutions Group, Intel, Penang, Malaysia (*wei.jun.yeap@intel.com*)

Rahul Chauhan, Programmable Solutions Group, Intel, Bangalore, India (*rahul.chauhan@intel.com*)

Wonyoung Choi, Programmable Solutions Group, Intel, Munich, Germany (*wonyoung.choi@intel.com*)

***Abstract*— Flip-flop based scan chain design is one of the fundamental Design-for-Test (DFT) features that has been widely used in modern VLSI designs. It offers the circuit enhanced controllability and observability. Scan flush architecture involves a hardware that controls scan chain by flushing out the existing value in the scan flip-flops (SFF) during the initial stage of transition from non-Automatic Test Pattern Generation (ATPG) mode to ATPG mode. User will only have control on the SFFs after the flushing phase completes. This paper proposes the techniques to verify design that implements scan flush architecture to ensure values stored in SFFs during non-ATPG mode are cleared before the chip enters ATPG mode. This is to prevent any important secrets from leaking out to the users through ATPG scan. The proposed techniques are able to verify non-resettable SFFs and resettable SFFs with different inputs.**

Keywords—scan chain, scan flush, verification, gate-level simulation

I. INTRODUCTION

Scan design is a design where flip-flops are replaced by scan flip-flops (SFF), and they are connected to form one or more shift registers in the test mode. It is a Design-for-Test (DFT) features that is frequently used in VLSI design to provide easy read or write access to storage elements and allow the user to observe the states of storage elements. It helps to simplify test pattern generation and reduces chip debug time. [1] Figure 1 shows an example of a short scan chain stitched from 5 SFF in a sequential circuit.

Although scan chain design is beneficial for testing purposes, it also brings security threats, which may allow a malicious attacker to steal information such as the cipher key from the scan flops. One of the known vulnerabilities, Mode Switching Attack, is an attack in which the attacker constantly switches between non-scan mode and scan mode, while providing plain text to an encryption algorithm that takes multiple rounds such as Advanced Encryption Standard (AES). The intermediate encryption result generated by AES after each round will be stored in a state register. The immediate AES result in earlier rounds of encryption is not completely secure. If the state register is part of the scan chain, then the security can be compromised by attacker. [3][4]

The scan flush architecture is an architecture of scan chain which is designed to avoid Mode Switching Attacks. It involves an implementation of scan flush hardware that flushes out the existing value in flops of scan chains during the initial stage of transition from non-Automatic Test Pattern Generation (ATPG) mode to ATPG mode. The scan flush hardware will automatically scan in 0 into all the scan chains during this stage. Before the flushing completes, the user should not have any control on the scan chains. Therefore, it is required to have sufficient clock cycles for the flushing phase for ensuring that all the data stored in the SFFs are completely removed. This architecture achieves similar result as the reset-based solution mentioned in [3] and [4], however it has the benefit that reset is not required to be done during transition from non-ATPG mode to ATPG mode. This paper proposes techniques to verify that the scan flush hardware is working as expected by using Gate-Level Simulation (GLS).

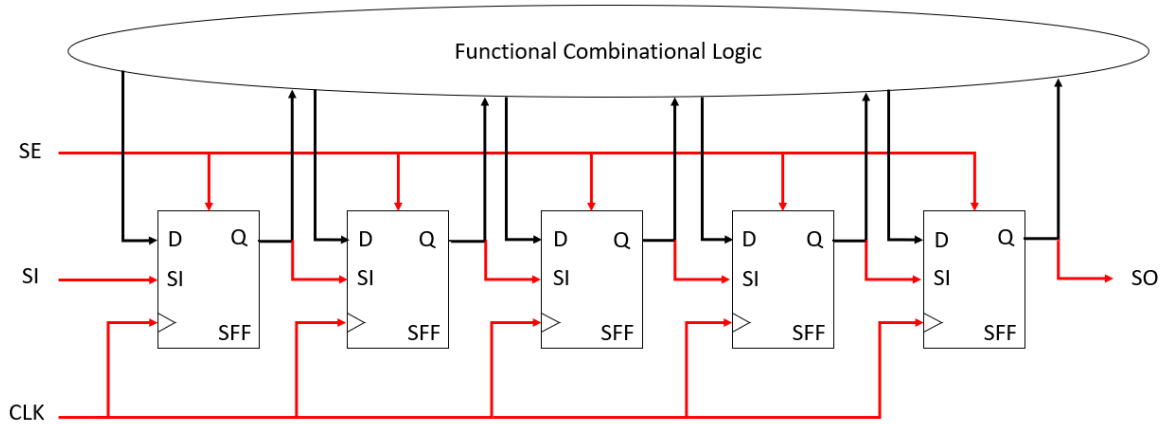

Figure 1 Scan chain [2]

II. Challenges in Verifying Scan Chain with Scan Flush Architecture in Gate-Level Simulation

In order to successfully verify scan chain design that implements the scan flush architecture in GLS, we have to ensure that all SFF should not keep their existing data after the flushing completes. The user must not have access to the data that are shifted out from the scan chain during the flushing process. They are also not allowed to scan in their own pattern during the process.

To prove that all the SFF are able to flush out their existing data, we had to initialize all the SFF with data that is opposite of the scan-in data during the flushing process. After the flushing is done, we use the simulation tool to extract the values of all the SFF at that simulation time, to check that they have the same value as the scan-in data from the scan flush hardware. Assertion checkers can be written to check that data cannot be shifted in and out by the user during the scan flush process.

Since the scan flush hardware always scan in 0 during the flushing phase, all SFF need to be initialized with 1, and the final output of all SFF needs to be 0. We cannot simply initialize the SFF to 1 by using force, as this will break the normal flow of data after the scan flushing starts, which defeats the purpose of this verification. Since there are resettable SFF and non-resettable SFF with different inputs (active clock or inactive clock), techniques to initialize them to 1 will be different for each category and combination of inputs.

III. Techniques to Initialize SFF to 1 in Scan Chain Design with Scan Flush Architecture

A. Flops with Preset Pins

Asynchronous SFF with preset pins can be easily initialized to 1, as the preset pins are able to override the Q-output of the SFF regardless of its D-input. The steps to initialize this type of SFF is as follows.

1) Wait for design power-on-reset (POR) to be done and clocks to be ready.
2) Forcing preset pin to 1 before scan flush starts.
3) Release force of preset pin immediately after scan flush starts by sampling the scan flush counter.

B. Flops without Preset Pins

For SFF without preset pins, we categorized them into 2 types which are non-resettable flops and resettable flops. The SFF can have different inputs such as active clock or inactive clock.

Before applying initialization techniques, we need to find the list of SFFs that need to be excluded from the initialization of flops. The SFFs that need to be excluded are the flops that would disrupt the design normal operation if it were set to 1. An example would be a POR related flop that is part of the scan chain. Initializing this type of flop to 1 before scan flush starts would cause unexpected behaviour and disrupt the hardware operation. The method to discover this category of flop is to force all the SFF to 1 after POR is done and design clock is ready. Although this would cause the scan flush operation to be unable to flush out the 1 value, the simulation waveform can still be used to determine whether the scan flush was able to complete or not. If it is not completed, then debugging has to be done manually to determine the SFF that causes the issue. All the SFF that is discovered through this method must be manually reviewed to ensure there is no risk in excluding them. If there is no risk, then it can be added into the exclusion list, otherwise, its initialization needs to be handled through some other design-specific methods.

Once all the prerequisite flows are done, the initialization steps will differ for each of the categories in the subsections.

1) Non-resettable SFF with Inactive Clock

The list of non-resettable flops in a design can be obtained using synthesis tool. First, we need to split the non-resettable SFFs into list of either gated or non-gated clocks**.** The testbench needs to start with forcing the D-input of all SFF to 1 after design main clock is ready. Next, use waveform extraction tools to extract the output for the flops at different times after the delay. If all the results of the sampling are 1 for specific flop, this means that clock is not gated for that flop, and vice-versa.

With the list of non-resettable SFFs with inactive clock ready. Initially, force the D-input of the flops to 1, followed by manually toggling a clock pulse using force and release feature of simulator. Once these steps are done, the flops should be initialized to 1. Once scan flush hardware starts to scan in 0 into the scan chains, it will clear the initialized flops. The force of D-input can be released any time after the scan flush starts. As the output of SFF will not be driven by D-input, but rather by SI-input during scan flush, this will not affect the scan flush operation.

2) Non-resettable SFF with Active Clock

The list of non-resettable SFFs with active clock will be available once the steps from III(B)(1) are run. Initially, force the D-input of the flops to 1. Once scan flush hardware starts to flush 0 into the scan chain, it will clear the initialized flops. Release the force of D-input sometime after the scan flush starts.

3) Synchronous Reset SFF with Inactive Clock

In order to extract the list of synchronous reset SFFs, the list of all SFFs should exclude the list of non-resettable flops from III(B)(1). A similar flow from III(B)(1) will be used to obtain the list of flops with gated or non-gated clock. The testbench needs to wait for design main clock to be ready and start with forcing the reset of all SFF to 0 or 1 depending on whether it is active low or active high reset. After that, it needs to force all the D-input to 1. Sampling is then done on each of the flops, at multiple times after the force of D-input. If all the results of the sampling are 1 for specific flop, this means that clock is not gated for that flop, and vice-versa.

The steps to initialize this category of flop starts by utilizing force to release the reset of the SFFs in this list. When the scan flush starts, the reset of all SFFs in the scan chains are always released, therefore releasing the reset earlier will not affect the scan flush behaviour. The next step is the same as the steps in III(B)(1), where the D-input of the flops is forced to 1, followed by manually toggling a clock pulse. An additional step that needs to be done is to release the force on reset before the scan flush is done. This is to ensure that it would not affect the reset behaviour of all SFFs in the scan chains when the design transition into ATPG mode.

4) Synchronous Reset SFF with Active Clock

Running the flow in III(B)(3) will generate the list of synchronous reset SFFs with active clock, by excluding the synchronous reset SFFs with inactive clock. First, release the reset of the SFFs using force, followed by forcing D-input to 1, which results in the flops being initialized to 1. After that, releasing of the force on D-input and reset has to be done within the scan flush duration to ensure it does not affect the normal behaviour of design after switching into ATPG mode.

IV. RESULT

Table 1 and Table 2 shows the total SFFs in the design being tested, and the number of flops in different categories that were successfully initialized to 1 after using the method proposed. 2421 flops were not initialized to 1 before scan flush started. This is due to the behaviour of some of the SFF in the design tested. The initialization steps are done in non-ATPG mode before the scan flush starts. In non-ATPG mode, some SFF are linked to each other, thus causing unexpected behaviour when initializing them to 1 simultaneously. To initialize the SFF, we are forcing the D-input. If it is connected to output of any other flops in non-ATPG mode, then it will backforce the output of the previous flop. If the previous flop that is backforced is part of the scan chain, then after releasing the initialization force, output of the previous flop will not be initialized to 1.

Table 1 Categories of SFFs in Design

Categories		**Number of SFFs**
a)	Flops Excluded	1
b)	Flops with Preset Pins	112
c)	Non-resettable SFF with Inactive Clock	8551
d)	Non-resettable SFF with Active Clock	5188
e)	Synchronous Reset SFF with Inactive Clock	187517
f)	Synchronous Reset SFF with Active Clock	208997

Table 2 Results of SFF Initialization on Design

Flops Initialized to 1 Before Scan Flush Starts	**407945**
Flops Not Initialized to 1 Before Scan Flush Starts	**2421**
Total Number of SFFs	**410366**
99.41% of the total SFFs are verified successfully using the proposed techniques.	

To prove our method, we ran a negative test, by modifying the gate-level netlist to connect the scan chain incorrectly. When running scan flush verification without going through the initialization process proposed, we did not observe any error in the flushing operation. However, when we ran the scan flush verification after initializing the SFFs to 1, we observed that the scan chain that has connection issues is unable to flush out the data properly.

V. ANALYSIS

The methodology proposed in this paper provides various benefits in scan flush design verification. The techniques only use common simulator features such as force and release, therefore allowing it to be used in any verification environment regardless of the simulator used. Successfully injecting 1 into the SFFs ensures that the scan flush hardware will prevent any secrets inside SFFs to be leaked out to the user during ATPG mode. This method can also verify that the scan chain in the design is stitched correctly.

Only 6 categories of SFFs are being used in the design tested. Additional methods need to be prepared for design that have SFFs that are not under these categories. Some flops also have to be excluded from the initialization, as they will disrupt the scan flush flow if they are initialized to 1 prior to scan flush, such as those that are related to POR. There is currently no automated method that can discover those flops, and manual reviews are needed for de-risking purposes. Furthermore, the 2421 flops from Table 2 that are unable to be initialized to 1 with the proposed methods, need to be handled manually with design specific solution for different scenarios.

Enhancement can be done in the future to make the method generalized for all types of flops. Current hardware simulators do provide features that allows easy initialization of registers in GLS. However, most of them are mainly targeted to perform initialization for non-resettable flops and will show unexpected behaviour when used differently. Hence, they are currently not suitable for our use case.

VI. CONCLUSION

This paper has introduced the challenges that will be faced during scan chain verification on scan flush architecture. The paper mainly focuses on the need to initialize all SFF to 1 for non-resettable and resettable flops. To solve this problem, we propose techniques to perform the necessary initialization for each type of flops with each possible combination of inputs in the design tested. For other designs with more types of flops, there may be room for improvement in the techniques proposed.

VII. REFERENCES

[1] P. Song, F. Stellari, T. Xia, A.J. Weger. “A Novel Scan Chain Diagnostics Technique Based on Light Emission from Leakage Current.” IEEE Xplore, 28 Oct. 2004, pp. 1, ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1386946&tag=1, https://doi.org/10.1109/TEST.2004.1386946. in press.

[2] S. Bhunia, M. Tehranipoor. Hardware Security: A Hands-on Learning Approach. 2019. Elsevier Inc., 2019, pp. 69, www.sciencedirect.com/science/article/pii/B9780128124772000083. in press.

[3] X. Li, W. Li, J. Ye, H. Li, Y. Hu. “Scan Chain Based Attacks and Countermeasures: A Survey.” IEEE Access, vol. 7, 26 June 2019, pp. 1-9, https://doi.org/10.1109/access.2019.2925237. in press.

[4] W.Z. Wang, et al. “Securing Cryptographic Chips against Scan-Based Attacks in Wireless Sensor Network Applications.” Sensors, vol. 19, no. 20, 22 Oct. 2019, pp. 1-14, https://doi.org/10.3390/s19204598. in press.

[5] W.Z. Wang, Y. Chen, S. Cai, Y. Peng. “Preventing Scan-Based Side-Channel Attacks by Scan Obfuscating with a Configurable Shift Register.” Security and Communication Networks, vol. 2021, no. 5222670, 5 Nov. 2021, www.hindawi.com/journals/scn/2021/5222670/, https://doi.org/10.1155/2021/5222670. in press.