

BOUNDARY SCAN DESIGN AND VERIFICATION FLOW USING BUFFER CHAIN METHOD

Eun-Seok Chae* and Hong-Shin Jun* and Hye-Kyong Song**

*System LSI Division, Design Service Team, Samsung Electronics, Kyungki 449-900, Korea

**Semiconductor R&D Division, CAE Team, Samsung Electronics, Kyungki 449-900, Korea

*E-mail:eschae@samsung.co.kr

Abstract

Boundary scan is well-known standard DFT technique, called IEEE Std 1149.1, that is designed with boundary scan registers placed between pads and internal logic. In this paper, we propose the boundary scan design and verification flow using a buffer chain method. The advantages of the proposed flow include: (1) The buffer chain method is easy to implement using Samsung in-house tool, (2) the flow is effective to solve the post-layout problems at the pre-layout step, and (3) the flow reduces the TAT (Turn Around Time) of the boundary scan design and verification.

I. INTRODUCTION

Boundary scan is a DFT technique that simplifies PCB (Printed Circuit Board) testing by providing a standard chip and board test interface.[5] The boundary scan uses a serial scan chain to access the chip I/Os on board. Since the scan chain passes through all the input and output pads of a chip, its inputs and outputs are accessible from the board for sampling data from the other chip and updating data to another chip simultaneously. Also, the sampled or updated data traverses through the long shifting path that spans the chip boundary. Because of these characteristics, the following problems may occur: invalid data shifting, slope waveform degradation, and ground bouncing.

The following Fig. 1 shows an example of boundary scan register and its port configurations that will be mentioned in this paper.

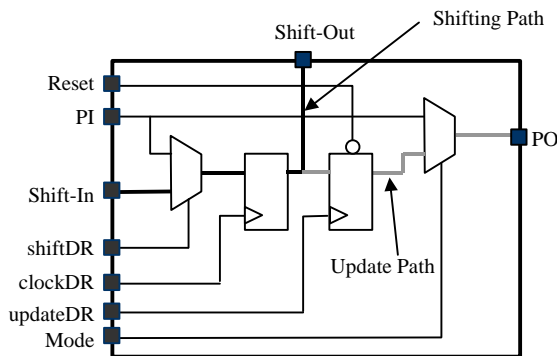


Fig 1. An Example of Boundary Scan Register Cell

The paper is organized as follows. The second section presents why those problems can occur and how they can be resolved. Also, we will describe the boundary scan design and verification flow that is to solve those problems. The third section shows the experimental results after the layout of a real device using the proposed flow.

The suggested design and verification flow involves Test-Compiler™, TestGen™ and in-house tool SADAS. The buffer chain method was implemented in SADAS.

II. METHODS

Problems and Solutions

Invalid data shifting can occur due to the hold time violation in a long shift register chain. Typically, a boundary scan chain has almost zero delay on its shifting path. The solution for this problem is that the flip-flop with the largest clock net delay, should be located closest to TDI pin, and that the flip-flop with the smallest clock net delay should be placed closest to TDO pin. The following explanations will provide basic concept on the reason why the *clockDR* network should be routed as mentioned above. Figure 2 shows the valid shift operation when there is no clock skew. Data A will be captured at the rising edge of CK1 (\hat{I}) by the flip-flop A. If the CK1 and CK2 have no clock skew, the flip-flop B will capture the data A at the rising edge of the CK2.

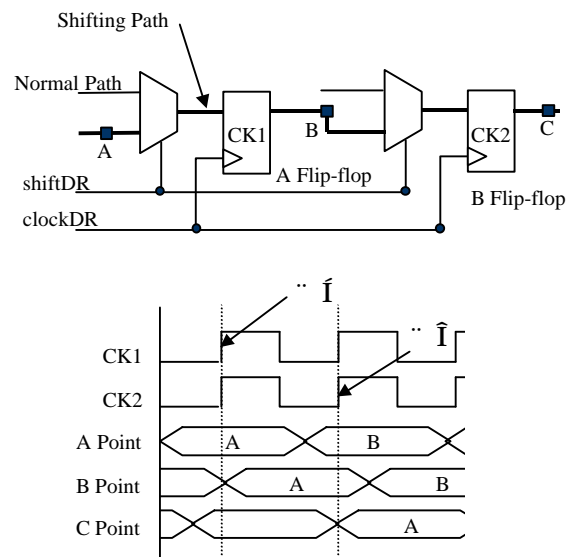


Fig 2. The Correct Operation

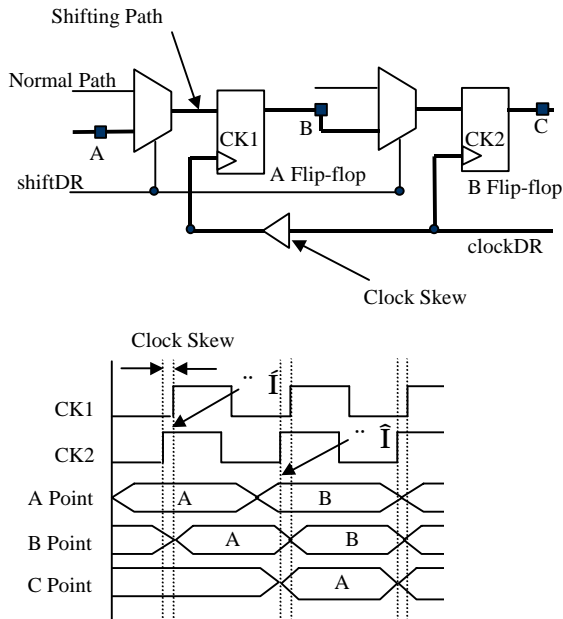


Fig 3. Operation with Clock Skew

Figure 3 depicts the shift operation when there is a clock skew. With the clock skew on the *clockDR* network, the waveform of CK2 can be delayed as shown in the figure. So, the data captured by the flip-flop B is the input value of the flip-flop A, instead of the output value of the flip-flop A. Hence, wrong data is shifted because of the skewed clock. For this reason, it violates the following scan design rule; Data should be shifted to one flip-flop per one clock period. If the size of a clock skew is small, i.e., within the range of hold time requirement, a hold time violation will occur.

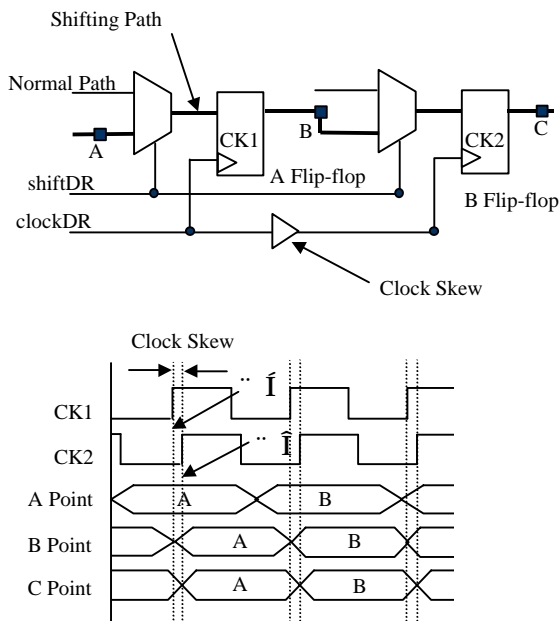


Fig 4. Operation after Re-Routing

To resolve this situation, it is suggested that the *clockDR* network should be routed as shown in Figure 4, that is, the order of *clockDR* connection should be reversed. Then, data A is captured at the rising edge of CK1 (\uparrow), and propagated to the flip-flop B at the rising edge of CK2 (\uparrow).

Because of a recent design trend toward deep sub-micron, the width of the signal line continues to decrease and thus, the resistance of the signal line increases. Slope waveform degradation can happen due to the resistance shielding effect, which occurs when the resistance of each signal line is increased. Boundary scan has long nets, *clockDR*, *shiftDR*, *updateDR*, *Mode*, and *Reset*, which span the whole boundary of a chip. Boundary scan has the problem of slope waveform degradation. The problem can be removed by adding buffers as shown in Figure 5. From the experiments with SPICE simulations, we found out that *nid* cell (non-inverting buffer with 1X driver) should be added to each net whenever it feeds 10 fan-outs for 0.35um technology, and *nid2* cell (non-inverting buffer with 2X driver) for 0.25um technology.

Ground bouncing can occur when inductance and/or resistance in the power distribution pathways injects a voltage drop across internal nodes that are supposedly referenced to the same value (power or ground). This has effect of superimposing a voltage fluctuation on a signal. If this signal is TCK and the fluctuation is large enough to create a new clock cycle, the synchronization with the TAP state can be lost [4] Ground bouncing occurs on the falling edge of TCK during update-DR or update-IR state in which all the outputs can switch simultaneously. After all, ground bouncing problem is related to the number of SSOs (Simultaneous Switching Outputs). By adding delays to the *updateDR* and *Mode* net whenever they feed 10 fan-outs that are result from SPICE simulations, The number of SSOs in the boundary scan operation can be reduced. Figure 5 shows how the delays are added to the *updateDR* and *Mode* net.

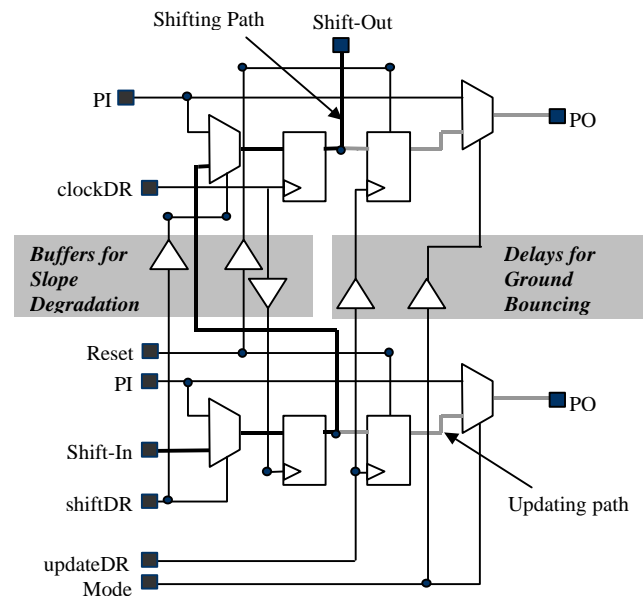


Fig 5. Structure when the Delays Are Inserted

The circuit in the experiments has 208 pins and it consists of multiplexers and flop-flops as shown in Figure 6.

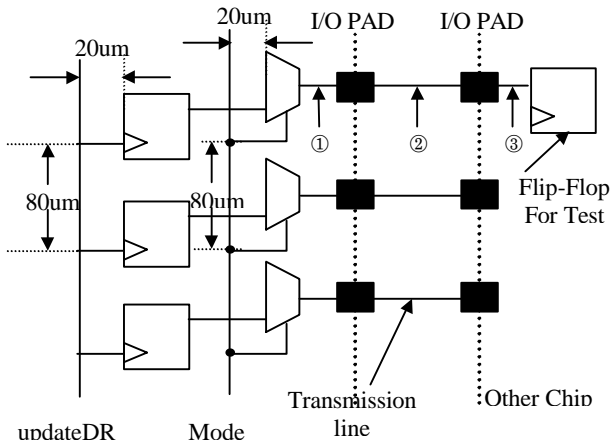


Fig 6. The Circuit Used in the Experiments

Figure 7 shows the slope waveform degradation with SPICE simulations when no delays are inserted. The top and bottom graph show the slope waveform degradation of the *Mode* net and *updateDR* net, respectively, at the measuring points of the 1st, 104th, 156th, 208th fan-outs from the TDI pin. Figure 7 shows that the slope waveform degradation is approximately 20-30ns, if no delays are inserted at those nets.

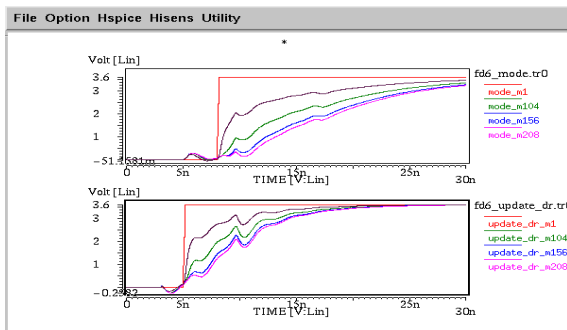


Fig 7. Slope Waveform Degradation If No Delays Are Inserted at the Mode and updateDR Net

Figure 8 shows the slope waveform degradation and the ground bouncing noise when a nid cell is added to each net whenever it feeds 10 flip-flops. The two upper and the two lower graphs show the slope waveform degradation and ground bouncing noise at the *Mode* net and the *updateDR* net, respectively. The measuring points of the ground bouncing noise experiment are ①, ② and ③ as shown in Figure 6. And the measuring points of the slope waveform degradation experiments are the 1st, 104th, 156th, 208th fan-outs from the TDI pin. The experiments show that when a nid cell is added to each net whenever it feeds 10 flip-flops, the slope waveform degradation and ground bouncing noise problem can be prevented.

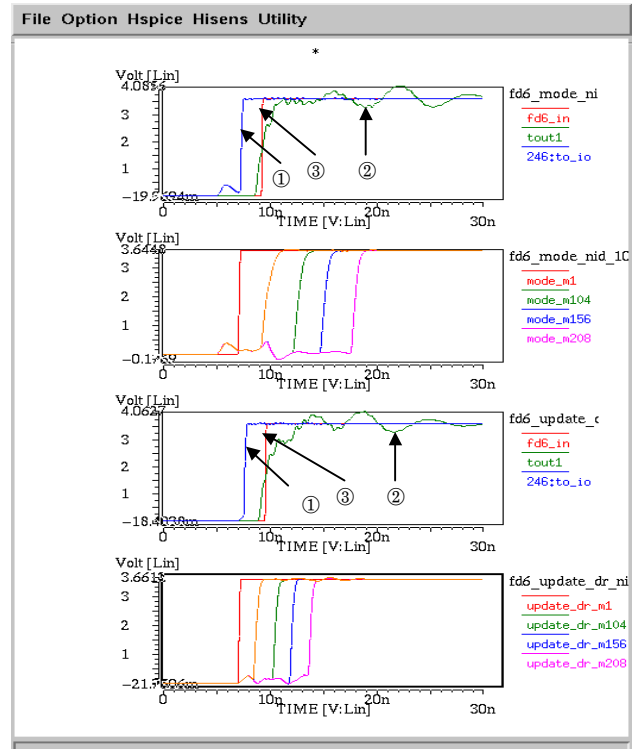


Fig 8. Slope Waveform Degradation and Ground Bouncing Noise If the Delays Are Inserted at the Mode and updateDR Net.

Design and Verification Flow

Figure 9 shows the boundary scan design and verification flow. This flow consists of 4 steps; boundary scan insertion, compliance check, buffer chain insertion, and timing simulation.

First, The boundary scan insertion step requires gate-level netlist and the port order information of the internal logic. The output of this step is boundary scan inserted gate-level netlist. Second, the compliance check step requires the boundary scan inserted gate-level netlist generated from the previous step and the boundary scan circuit information such as loaded instruction and TAP controller state. The outputs of this step are BSDL (Boundary Scan Description Language) description and functional vector. Third, the buffer chain insertion step requires package bonding pad order information and net names for buffer insertion and the boundary scan inserted gate-level netlist. The outputs of this step are boundary scan inserted gate-level netlist with buffer chains and *BSRLIST*, which will be used placement and routing tool to locate boundary scan registers and to insert delays near the corresponding I/O pads automatically. Finally, timing simulation verifies the functional vector with delay.

Test-Compiler™ is used for boundary scan insertion, TestGen™ is used for compliance check. Apollo™ is used for placement and routing, and SADAS is used for buffer chain insertion.

The main goal of this flow is to remove the post-layout problems described above at the pre-layout step using the buffer chain method.

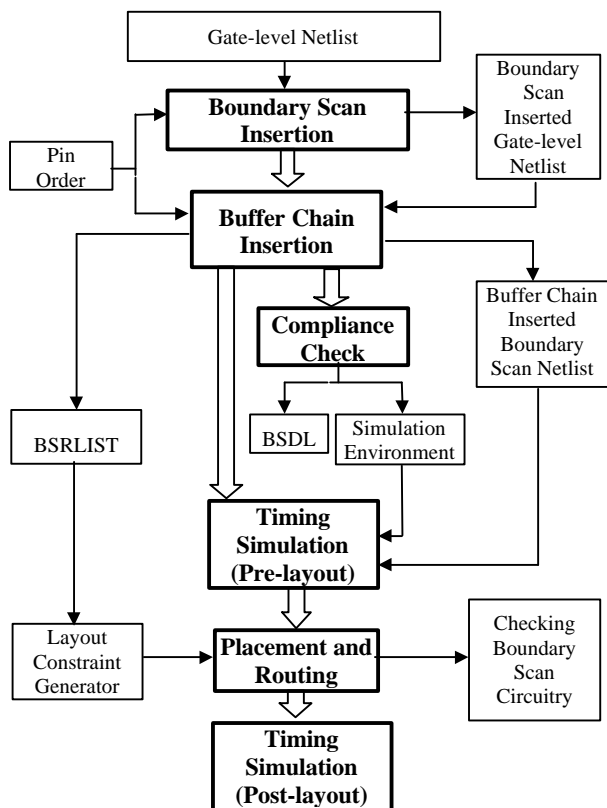


Fig 9. The Boundary Scan Design and Verification Flow Diagram

III. RESULTS

The proposed design and verification flow has been applied to a real device. Table 1 shows the circuit information of the device.

Table 1. Device Information

Chip Size	13000um x 13000um
Gate Counts	1800Kgates
# of Pins	304
# of BSRs	604
# of Power / Ground Pad	129
# of Inserted Buffers	190
Technology	0.35um, 3.3V

Figure 10 shows the slope waveform degradation of all nets in the boundary scan circuit after layout. In case of 0.35um technology, the slope waveform degradation is less than 2.0ns and ground bouncing is reduced. The number of SSOs is about 20 pins/1ns at the most critical point. This device is compliant for the IEEE Std 1149.1 and is performed pre-layout and post-layout timing simulations with the boundary scan functional vector

successfully. Although the device is very large and complicated, all the problems associated with the boundary scan design are resolved successfully.

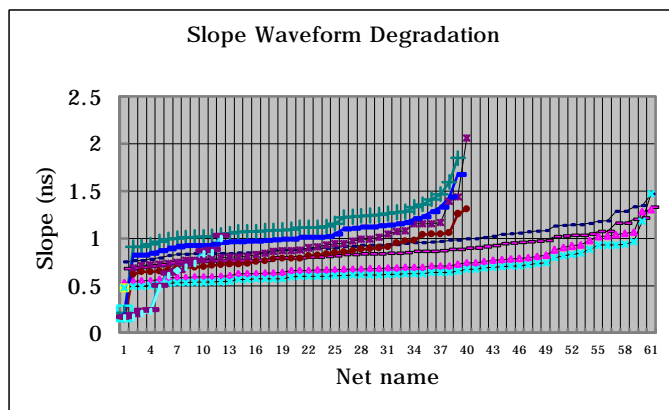


Fig 10. The Slope Waveform Degradation in a Real Device

IV. CONCLUSIONS

We have explained the problems in boundary scan design and proposed solution using a buffer chain method. The buffer chain method is easy to implement and provides total solution of the boundary scan problems: invalid data shifting, slope waveform degradation and ground bouncing. The boundary scan design and verification flow has been adopted to a real device without any design iteration.

Acknowledgements

The authors wish to thank Mr. Chi-Ho Cha of Design Kit Group in Samsung Electronics, for his help in the experiments

REFERENCES

- [1] IEEE Std 1149.1-1990, "IEEE Standard Test Access Port and Boundary-Scan Architecture," The Institute of Electrical and Electronics Engineers, Inc, Feb. 1990
- [2] H.Bleeker and P.Eijnden and F. de Jong "Boundary-Scan Test - A Practical Approach," Kluwer Academic Publishers, 1993
- [3] C.M.Mauder and R.E.Tulloss, "The Test Access Port And Boundary Scan Architecture," IEEE Computer Society Press Tutorial, 1990
- [4] "STD90/MDL110 0.35um 3V CMOS Standard Cell Library for Pure Logic / MDL Products Databook," SEC-ASIC, 1997
- [5] Kenneth P. Parker, "The Boundary-Scan Hand Book," Kluwer Academic Publishers , pp. 167-169, 1992
- [6] Markus F. Robinson "Technology Independent Boundary Scan Synthesis (Design Flow Issues)," The Institute of Electrical and Electronics Engineers, Inc., pp. 416-421, 1993