ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

On-Chip Test Generation system for conducting functional broadside tests and achieves hightransition fault coverage for testable circuits.

Pasupuleti Gopalarao
MTech Student
Dept of ECE
Anurag engineering college

L.Nageswara rao
Assistant professor
Dept of ECE
Anurag engineering college

M.Basha, MTech
HoD & Associate Professor
Dept of ECE
Anurag engineering college

Abstract:

Very-large-scale integration (VLSI) is the process of creating an integrated circuit (IC) by combining thousands of transistors into a single chip.Functional verification, in electronic design automation, is the task of verifying that the logic design conforms to specification. In everyday terms, functional verification attempts to answer the question "Does this proposed design do what is intended?" This is a complex task, and takes the majority of time and effort in most large electronic system design projects. Functional verification is a part of more encompassing design verification, which, besides functional verification, considers non-functional aspects like timing, layout and power. During these functional tests power dissipation doesn't exceed that possible during functional operation. Built-in generation of functional broadside tests is done using a fixed hardware structure to achieve higher fault coverage. The structure is added to a given circuit that needs to be tested. On-chip test generation avoids the need to compute reachable states offline by generating the reachable states during test application.

Keywords: ***Functional tests, Board, Chip, transition faults, fault coverage.***

Introduction:

Testing a circuit is necessary to make sure that all design errors have been fixed. Over testing occurs due to nonfunctional operation conditions created by unreachable scanning states. Tests applied under nonfunctional operation conditions, which are made possible by scanning in an unreachable state, may lead to unnecessary yield loss. Slow paths that cannot be sensitized may cause circuit to fail and when current demands higher than possible cause voltage drop leading to the circuit failure.

Functional verification is very difficult because of the sheer volume of possible testcases that exist in even a simple design. Frequently there are more than 10^80 possible tests to comprehensively verify a design - a number that is impossible to achieve in a lifetime.[according to whom?] This effort is equivalent to program verification, and is NP-hard or even worse - and no solution has been found that works well in all cases. However, it can be attacked by many methods. None of them are perfect, but each can be helpful in certain circumstances:

- Logic simulation simulates the logic before it is built.
- Simulation acceleration applies special purpose hardware to the logic simulation problem.
- Emulation builds a version of system using programmable logic. This is expensive, and still much slower than the real hardware, but orders of magnitude faster than simulation. It can be used, for example, to boot the operating system on a processor.
- Formal verification attempts to prove mathematically that certain requirements (also expressed formally) are met, or that certain undesired behaviors (such as deadlock) cannot occur.

- Intelligent verification uses automation to adapt the testbench to changes in the register transfer level code.
- HDL-specific versions of lint, and other heuristics, are used to find common problems.

The most common method for delivering test data from chip inputs to internal circuits under test (CUTs, for short), and observing their outputs, is called scan-design. In scan-design, registers (flip-flops or latches) in the design are connected in one or more scan chains, which are used to gain access to internal nodes of the chip. Test patterns are shifted in via the scan chain(s), functional clock signals are pulsed to test the circuit during the "capture cycle(s)", and the results are then shifted out to chip output pins and compared against the expected "good machine" results.

Straightforward application of scan techniques can result in large vector sets with corresponding long tester time and memory requirements. Test compression techniques address this problem, by decompressing the scan input on chip and compressing the test output. Large gains are possible since any particular test vector usually only needs to set and/or examine a small fraction of the scan chain bits. The output of a scan design may be provided in forms such as Serial Vector Format (SVF), to be executed by test equipment.

Functional broadside tests allow only reachable states as scan-in states. As broadside tests are two pattern tests, the circuit undergoes state transitions that are possible during functional operations. Delay faults can effect functional operation and current demands don’t exceed the limit. This avoids overtesting. Power dissipation also don’t exceed that possible.

Functional and pseudo functional scan based tests compute reachable states offline. Pseudo functional tests use functional constraints to avoid arbitrary states. These tests are not sufficient for avoiding unreachable states. Delay fault coverage in pseudo functional scan based tests and arbitrary broadside tests is higher than the fault coverage in functional broadside tests due to the fact that functional broadside tests doesn’t allow unreachable states where as other methods allow them. Moreover, the need for higher fault coverage is the one of the reasons to cause overtesting and power dissipation exceed more than possible during functional operation.

In the proposed method we use the on-chip generation of functional broadside tests where on-chip generation has the advantages of speed and reduces test data volume. On-chip generation doesn’t impose any constraints on delay faults as in pseudo functional scan based tests. Functional broadside tests ensure that scan-in state is only a reachable state. The reachable states are generated during test application by the circuit. This avoids computing them offline.

If a primary input sequence A is applied in functional mode starting from a reachable state, all the states traversed under A are reachable states. Any one of these states is used as initial state. For the detection for set of faults F we need |F| different reachable states. Typically it is small fraction of A. So, primary input sequence A doesn’t need to take the circuit through all reachable states but only a number relative to |F| in order to be effective.

Linear Feedback Shift Registers

The LFSR are the basic building blocks of the pseudo random test pattern generators. In unbiased pseudo random testing, the outputs from the LFSR is fed directly to the CUT and thus the no. of LFSR stages required is equal to the number of inputs to the CUT. For a weighted pseudo random testing we however require much more LFSR stages than the inputs to the CUT. This is so because each weighted bit usually requires more than one equi-probable bit coming in from an LFSR stage for the generation of its weighted bit. Now we assume that for both the unbiased and the weighted case we have the total number of LFSR shift registers required for each of the CUTs.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

LFSR Design

The hardware used for generating a primary input sequence consists of a Linear Feedback Shift Register and a small number of gates. Gates are used to modify the random sequence in order to avoid repeated synchronization that is the sequence takes the circuit to repeat the same or similar states. In addition to this, a single gate is used for determining the tests to be applied based on primary input sequence.

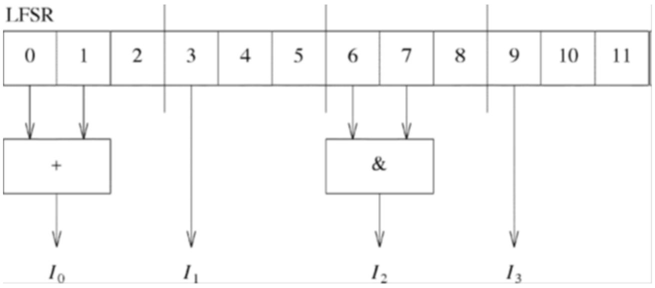


Fig: .On-chip generation of A

The fixed hardware structure needs to be tailored to the given circuit through these parameters.

1) The number of LFSR bits.

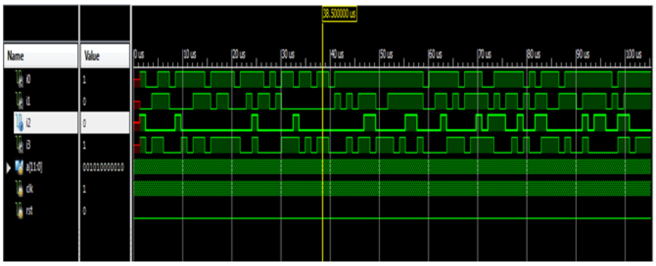
2) Seeds for the LFSR to generate different primary input sequences and several subsets of tests.

3) The length of primary input sequence.

4) The specific gates used to modify LFSR sequence into primary input sequence A.

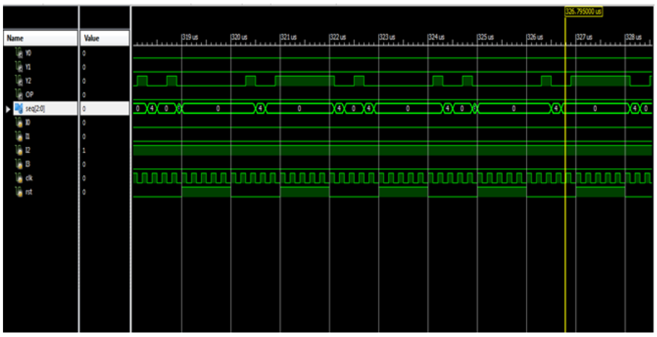
5) Specific gates for selecting the type of functional broadside tests that will be applied to the circuit based on primary input sequence A.

Simulation Results



Name	Value
i0	1
i1	0
i2	0
i3	1
a[11:0]	001010000010
clk	1
rst	0

Fig:sequence generator outputs.



Name	Value
Y0	0
Y1	0
Y2	0
OP	0
seq[2:0]	0
I0	0
I1	0
I2	1
I3	0
clk	0
rst	0

Fig: S_27 Functional Broad Side Test Sequences

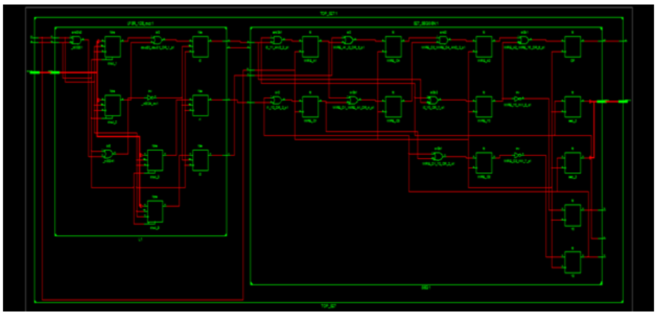
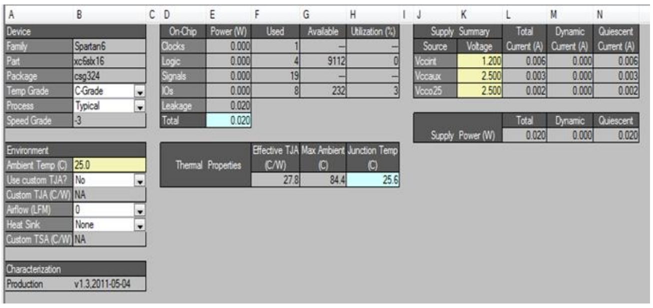


Fig: RTL schematic diagram

Power report:



Device	
Family	Spartan6
Part	xc6slx16
Package	csg324
Temp Grade	C-Grade
Process	Typical
Speed Grade	-3

On-Chip	Power (W)	Used	Available	Utilization (%)
Clocks	0.000	1	---	---
Logic	0.000	4	9112	0
Signals	0.000	19	---	---
IOs	0.000	8	232	3
Leakage	0.020			
Total	0.020			

Supply Summary		Total	Dynamic	Quiescent
Source	Voltage	Current (A)	Current (A)	Current (A)
Vccint	1.200	0.006	0.000	0.006
Vccaux	2.500	0.003	0.000	0.003
Vcco25	2.500	0.002	0.000	0.002

	Total	Dynamic	Quiescent
Supply Power (W)	0.020	0.000	0.020

Environment	
Ambient Temp (C)	25.0
Use custom TJA?	No
Custom TJA (C/W)	NA
Airflow (LFM)	0
Heat Sink	None
Custom TSA (C/W)	NA

Thermal Properties	Effective TJA (C/W)	Max Ambient (C)	Junction Temp (C)
	27.8	84.4	25.6

Characterization	
Production	v1.3,2011-05-04

Conclusion

A fixed and simple hardware structure is implemented to conduct functional broadside tests on-chip. The hardware is implemented using the primary input sequence to a circuit with known reachable state to achieve additional reachable states. Random sequences are chosen to avoid repeated synchronization. Two pattern tests are done to achieve higher fault coverage. The parameters that are tailored the circuit under test to conduct functional broadside tests on-chip are length of LFSR, seeds for LFSR, length of primary sequence, gates for modifying the sequence and gates for selecting tests based on primary input sequence.

References:

[1] Irith Pomeranz, Built-In Generation of Functional Broadside Tests Using a Fixed Hardware Structure, IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS, VOL. 21, NO. 1, JANUARY 2013

[2]M.Pavan Kumar Reddy & K.Bala, Design and Characterization of Parallel Prefix Adders, IJMETMR (www.ijmetmr.com), Volume No: 1(2014), Issue No: 10 (October)

[3] A. Stroele, "A self test approach using accumulators as test pattern generators," in Proc. Int. Symp. Circuits Syst., 1995, pp.2120–2123.

[4] H. J. Wunderlich, "Multiple distributions for biased random test patterns," in Proc. IEEE Int. Test Conf., 1988, pp. 236–244.

[5] I. Pomeranz and S. M. Reddy, "3 weight pseudo-random test generation based on a deterministic test set for combinational and sequential circuits," IEEE Trans. Comput.- Aided Des. Integr. Circuits Syst., vol.12, no. 7, pp. 1050–1058, Jul. 1993.

[6] K. Radecka, J. Rajski, and J. Tyszer, "Arithmetic built-in self-test for DSP cores," IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., vol. 16, no. 11, pp. 1358–1369, Nov. 1997.

[7] J. Rajski and J. Tyszer, Arithmetic Built-In Self Test For Embedded Systems. Upper Saddle River, NJ: Prentice Hall PTR, 1998.

[8] S. Wang, "Low hardware overhead scan based 3-weight weighted random BIST," in Proc. IEEE Int. Test Conf., 2001, pp. 868–877.

[9] S. Zhang, S. C. Seth, and B. B. Bhattacharya, "Efficient test compaction for pseudo-random testing," in Proc. 14th Asian Test Symp., 2005, pp. 337–342.

[10] J. Savir, "Distributed generation of weighted random patterns," IEEE Trans. Comput., vol. 48, no. 12, pp. 1364–1368, Dec. 1999.

[11] I. Voyiatzis, D. Gizopoulos, and A. Paschalis, "Accumulator-based weighted pattern generation," presented at the IEEE Int. Line Test Symp., Saint Raphael, French Riviera, France, Jul. 2005.

[12] F. Brglez and H. Fujiwara, "A neutral netlist of 10 combinational benchmarks circuits and a target translator in FORTRAN," presented at the Int. Symp. Circuits Syst., Kyoto, Japan, 1985.

Authors:

Pasupuleti Gopalarao