ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 10, October 2014

Efficient Implementation of Fault Coverage Circuit for High Speed and Low Power Applications

A.S.R.N.Raju [1], C .Malleswar [2]

M.Tech (Scholar), Dept. of ECE, Vasavi Inistute of Engineering & Technology, Andhra Pradesh, India[1]

HOD, Dept. of ECE, Vasavi Inistute of Engineering & Tech Technology, Andhra Pradesh, India [2]

ABSTRACT: Test Pattern generation has long been carried out by using Linear Feedback Shift Registers (LFSR's). LFSR's are a series of flip-flop's connected in series with feedback taps defined by the generator polynomial. The seed value is loaded into the outputs of the flip-flops. The only input required to generate a random sequence is an external clock where each clock pulse can produce a unique pattern at the output of the flip-flops. This random sequence at the output of the flip-flops can be used as a test pattern. The number of inputs required by the circuit under test must match with the number of flip-flop outputs of the LFSR. To reduce the power by maintaining the fault coverage in these project three intermediate patterns between the random patterns is generated. The goal of having intermediate patterns is to reduce the transitional activities of Primary Inputs (PI) which eventually reduces the switching activities inside the Circuit under Test (CUT) and hence power consumption is also reduced without any penalty in the hardware resources. The experimental results for c17 benchmark, with and without fault confirm the fault coverage of the circuit being tested. In the paper the power is mention that 14mw. Now the proposed system has to reduce it to less than 14mw i.e. nearly 12mw. At the same time we will reduce the device utilization also.

KEYWORDS: Cognitive Radio, Spectrum Sensing, Efficient Communication, System Security.

I.INTRODUCTION

Very-large-scale integration (VLSI) is the process of creating integrated circuits by combining thousands of transistor-based circuits into a single chip. VLSI began in the 1970s when complex semiconductor and communication technologies were being developed. The microprocessor is a VLSI device [1-3]. The term is no longer as common as it once was, as chips have increased in complexity into the hundreds of millions of transistors. The first semiconductor chips held one transistor each. Subsequent advances added more and more transistors, and, as a consequence, more individual functions or systems were integrated over time. The first integrated circuits held only a few devices, perhaps as many as ten diodes, transistors, resistors and capacitors, making it possible to fabricate one or more logic gates on a single device. Now known retrospectively as "small-scale integration" (SSI), improvements in technique led to devices with hundreds of logic gates, known as large-scale integration (LSI), i.e. systems with at least a thousand logic gates. Current technology has moved far past this mark and today's microprocessors have many millions of gates and hundreds of millions of individual transistors. At one time, there was an effort to name and calibrate various levels of large-scale integration above VLSI. Terms like Ultra-large-scale Integration (ULSI) were used. But the huge number of gates and transistors available on common devices has rendered such fine distinctions moot. Terms suggesting greater than VLSI levels of integration are no longer in widespread use. Even VLSI is now somewhat quaint, given the common assumption that all microprocessors are VLSI or better [4,5].

As of early 2008, billion-transistor processors are commercially available, an example of which is Intel's Montecito Itanium chip. This is expected to become more commonplace as semiconductor fabrication moves from the current generation of 65 nm processes to the next 45 nm generations (while experiencing new challenges such as increased variation across process corners). Another notable example is NVIDIA's 280 series GPU.

10.15662/ijareeie.2014.0310052
Copyright to IJAREEIE

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 10, October 2014

This microprocessor is unique in the fact that its 1.4 Billion transistor count, capable of a teraflop of performance, is almost entirely dedicated to logic (Itanium's transistor count is largely due to the 24MB L3 cache) [1]. Current designs, as opposed to the earliest devices, use extensive design automation and automated logic synthesis to lay out the transistors, enabling higher levels of complexity in the resulting logic functionality. Certain high-performance logic blocks like the SRAM cell, however, are still designed by hand to ensure the highest efficiency (sometimes by bending or breaking established design rules to obtain the last bit of performance by trading stability). The main challenging areas in VLSI are performance, cost, and power dissipation. Due to switching i.e. the power consumed testing, due to short circuit current flow and charging of load area, reliability and power. The demand for portable computing devices and communications system are increasing rapidly. These applications require low power dissipation VLSI circuits. The power dissipation during test mode is 200% P more than in normal mode. Hence it is important aspect to optimize power during testing. Power optimization is one of the main challenges. Test Pattern generation has long been carried out by using conventional Linear Feedback Shift Registers (LFSR's5). LFSR's are a series of flip-flop's connected in series with feedback taps defined by the generator polynomial. The seed value is loaded into the outputs of the flip-flops. The only input required to generate a random sequence is an external clock where each clock pulse can produce a unique pattern at the output of the flip-flops. This random sequence at the output of the flip-flops can be used as a test pattern. The number of inputs required by the circuit under test must match with the number of flip-flop outputs of the LFSR. This test pattern is run on the circuit under test for desired fault coverage. The power consumed by the chip under test is a measure of the switching activity of the logic inside the chip which depends largely on the randomness of the applied input stimulus. Reduced correlation between the successive vectors of the applied stimulus into the circuit under test can result in much higher power consumption by the device than the budgeted power. A new low power pattern generation technique is implemented using a modified conventional Linear Feedback Shift Register.

II. ARCHITECTURE OF THE PROPOSED MODEL

The main challenging areas in VLSI are performance, cost, power dissipation is due to switching i.e. the power consumed testing, due to short circuit current flow and charging of load area, reliability and power [1-3]. The demand for portable computing devices and communications system are increasing rapidly. These applications require low power dissipation VLSI circuits. The power dissipation during test mode is 200% P more than in normal mode. Hence it is important aspect to optimize power during testing. Power optimization is one of the main challenges

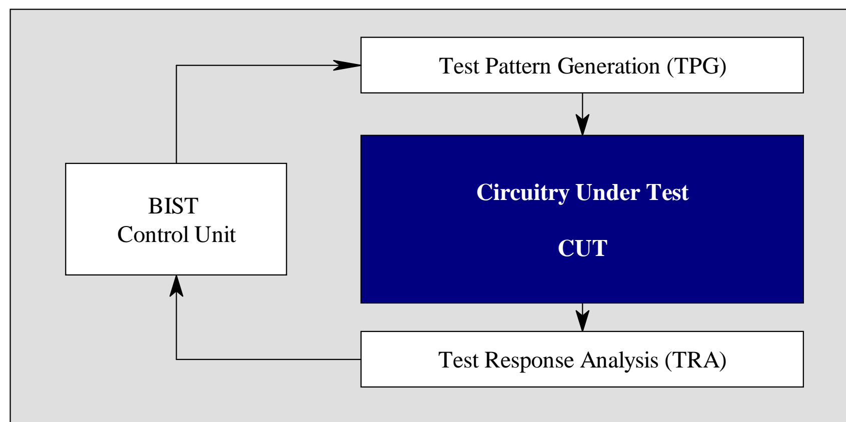

Fig 1 Test Pattern Generator

It generates test pattern for CUT. It will be dedicated circuit or a micro processor. Pattern generated may be pseudo random numbers or deterministic sequence. Here we are using a Linear Feedback Shift Register for generating random number. The Architecture for LFSR is as shown below.

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 10, October 2014

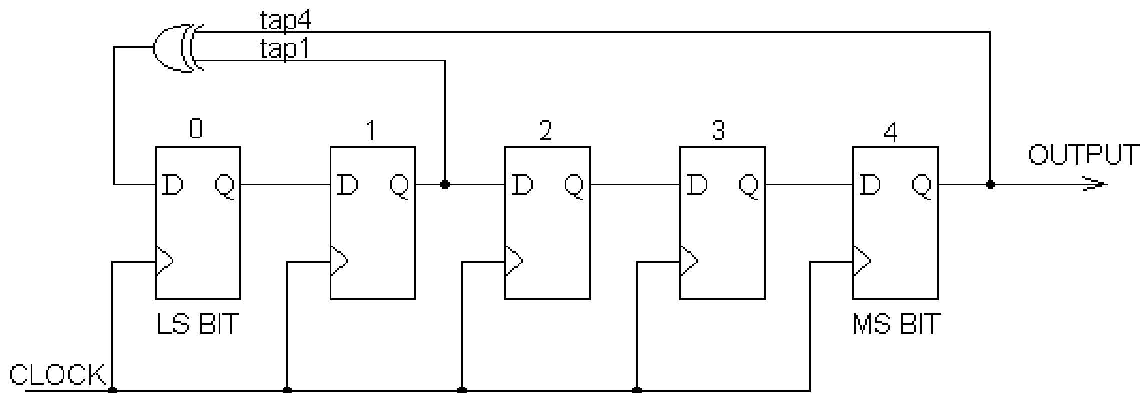

Fig 2 The Architecture for LFSR

Tapping can be taken as we wish but as per taping change the LFSR output generate will change & as we change in no of flip-flop the probability of repetition of random number will reduce. The initial value loading to the LFSR is known as seed value.

Test Response Analyzer (TRA): TRA will check the output of MISR & verify with the input of LFSR & give the result as error or not.
BIST Control Unit: Control unit is used to control all the operations. Mainly control unit will do configuration of CUT in test mode/Normal mode, feed seed value to LFSR, Control MISR & TRA. It will generate interrupt if an error occurs. You can clear interrupt by interrupt_clear_i signal.
Circuit under Test (CUT): CUT is the circuit or chip in which we are going to apply BIST for testing stuck at zero or stuck at one error.
Need for using BIST technique: Today's highly integrated multi-layer boards with fine-pitch ICs are virtually impossible to be accessed physically for testing. Traditional board test methods which include functional test, only accesses the board's primary I/Os, providing limited coverage and poor diagnostics for board-network fault. In circuit testing, another traditional test method works by physically accessing each wire on the board via costly "bed of nails" probes and testers. To identify reliable testing methods which will reduce the cost of test equipment, a research to verify each VLSI testing problems has been conducted. The major problems detected so far are as follows:

- Test generation problems
- Gate to I/O pin ratio

- **Test Generation Problems**

The large number of gates in VLSI circuits has pushed computer automatic-test-generation times to weeks or months of computation. The numbers of test patterns are becoming too large to be handled by an external tester and this has resulted in high computation costs and has outstripped reasonable available time for production testing.

- **The Gate to I/O Pin Ratio Problem**

As ICs grow in gate counts, it is no longer true that most gate nodes are directly accessible by one of the pins on the package. This makes testing of internal nodes more difficult as they could neither no longer be easily controlled by signal from an input pin (controllability) nor easily observed at an output pin (observe ability). Pin counts go at a much slower rate than gate counts, which worsens the controllability and observe ability of internal gate nodes.

III.SYNTHESIS IN DESIGN PROCESS

Verilog HDL is a hardware description language that allows a designer to model a circuit at different levels of abstraction, ranging from the gate level, register-transfer level, behavioral level to the algorithmic level. Thus a circuit can be described in many different ways, not all of which may be synthesizable/Compounding this is the fact that Verilog HDL was designed primarily as a simulation language and not as a language for synthesis. Consequently, there

10.15662/ijareeie.2014.0310052
Copyright to IJAREEIE

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 10, October 2014

are many constructs in Verilog HDL that have no hardware counterpart, for example, the $display system call. Also there is no standardized subset of Verilog HDL for register-transfer level synthesis.

Because of these problems, different synthesis systems support different Verilog HDL subsets for synthesis. Since there is no single object in Verilog HDL that means a latch or a flip-flop, each synthesis system may provide different mechanisms to model a flip-flop or a latch. Each synthesis system therefore defines its own subset of Verilog HDL including its own modeling style.

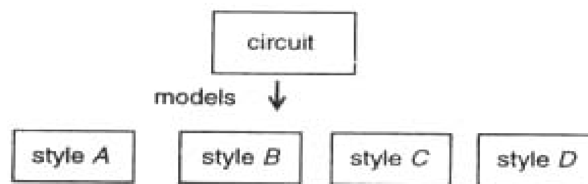

Fig.3. Same behavior different styles

Figure 4shows a circuit that is described in many different ways using Verilog HDL. A synthesis system that supports synthesis of styles A and B may not support that of style C. This implies that typically synthesis models are non-portable across different synthesis models are non-portable across different synthesis systems. Style D may not be synth & sizable at all.

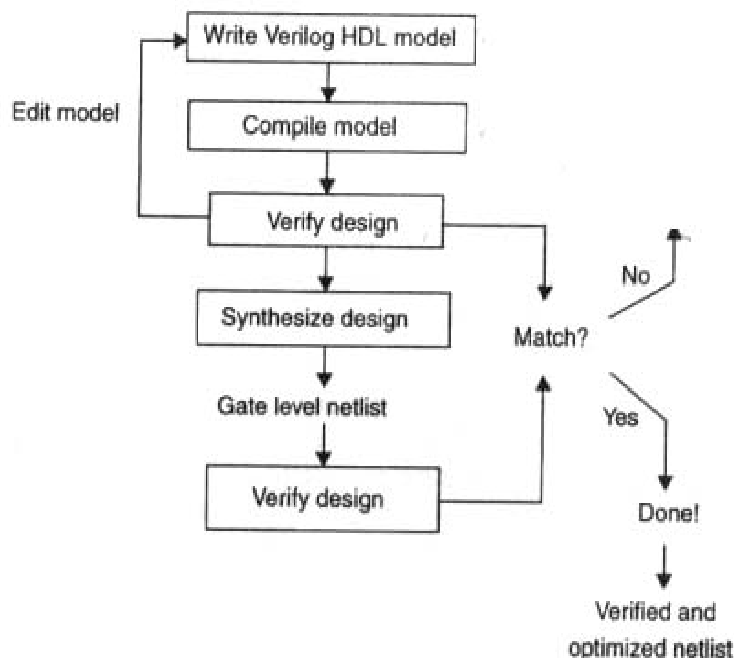

Fig 4. Typical design process

This limitation creates a severe handicap because now the designer not only has to understand Verilog HDL, but also has to understand the synthesis-specific modeling style before a synihcsizable model can be written. The typical design process shown in Figure 5.4 can not always be followed for Veri!og HDL synthesis.

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 10, October 2014

IV. RESULT AND DISCUSSION

The obtained results from the above mentioned methodology are presented from Fig.5 to Fig.7. Fig.5. represents the schematic diagram of the representation of the proposed model. The pin configuration and the operation is as discussed above.

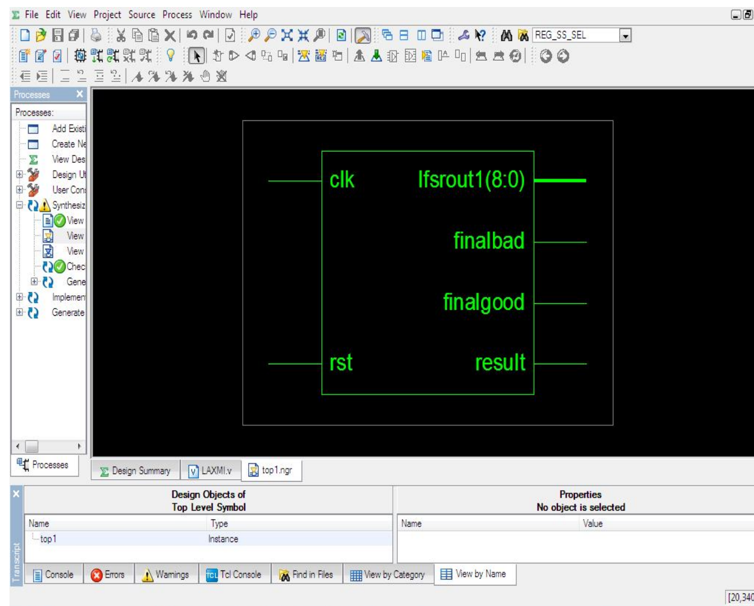

Fig. 5. Schematic Diagram

Fig.6 represents the RTL schematic of the proposed model. The resultant out wave forms for the concerned inputs are given in the Fig.7. The inputs and the corresponding outputs can be seen from this and the operation can be understood and vaidated later.

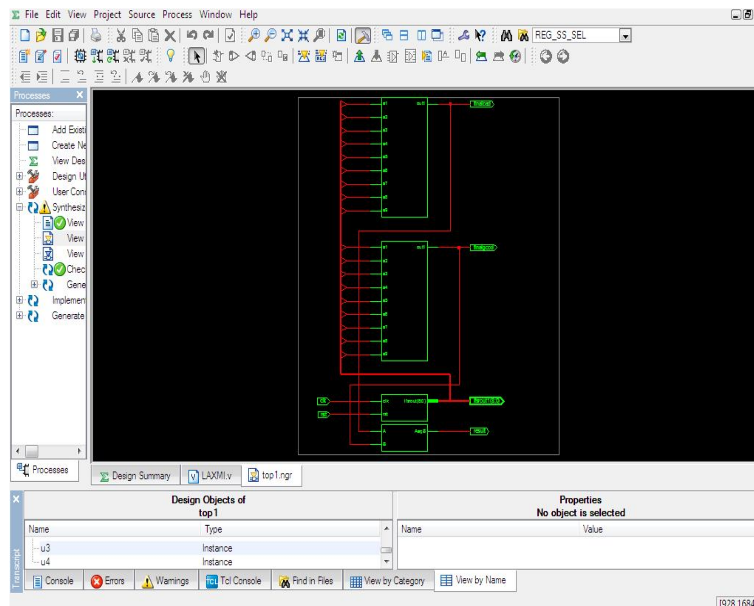

Fig. 6. RTL Schematic

Copyright to IJAREEIE

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 10, October 2014

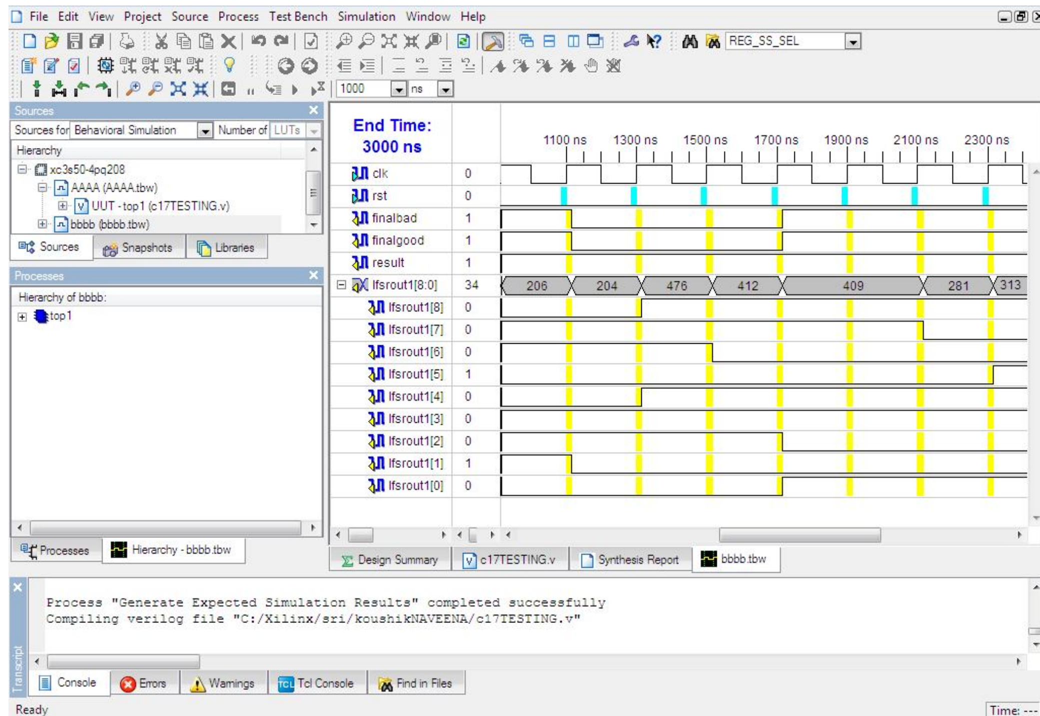

Fig. 7. Wave forms

VI.CONCLUSION

The proposed approach shows the concept of reducing the transitions in the test pattern generated. The transition is reduced by increasing the correlation between the successive bits. The simulation results shows that how the patterns are generated for the applied seed vector. This paper presents the implementation with regard to verilog language. Synthesizing and implementation (i.e. Translate, Map and Place and Route) of the code is carried out on Xilinx - Project Navigator, ISE 8.2i suite. The power reports shows that the proposed low power lfsr consumes less power (12 mw) during testing by taking the benchmark circuit C17 . In future there is a chance to reduce the power somewhat more by doing modifications in the proposed architecture.

.

REFERENCES

[1] Michael L.Bushnell, Vishwani D.Agawal," Essentials of electronic testing for digital, memory and mixed-signal VLSI circuits," Kluwer Academic Publishers, 2000.

[2] Mohammad Tehranipoor, Mehrdad Nourani, Nisar Ahmed," Low-Transition LFSR for BIST-Based Applications," 14th Asian Test Symposium, pp. 138- 143, 18-21 Dec. 2005..

[3] F. Corno, P. Prinetto, M. Rebaudengo, M. Sonza Reorda," A Test Pattern Generation methodology for low power onsumption," pp.1-5, 2008.

[4] P.Glard et al "Survey 0 F Low-Power Testing Of Vlsi Circuits"IEEE Design & Test Of Computers, vol. 19, no. 3. (2002), pp. 80-90.

[5] Dr.K.Gunavathi, Mr. K. ParamasivaM, Ms.P.Subashini Lavanya, M.Umamageswaran," A novel BIST TPG for testing of VLSI circuits", IEEE International Conference on Industrial and Information Systems, pp. 8 - 11, August 2006.