JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

A PROGRAMMABLE AND PARAMETERISABLE RESEEDING LINEAR FEEDBACK SHIFT REGISTER

Dr. Mohammad. Hayath Rajvee[1],
[1]Professor, Dept: of ECE, PBR VITS, Kavali, AP, India.
A.Siva[2], Sk Sameer[3], M. Lakshmi[4], M. Aparna[5].
[2,3,4,5] UG Students, Dept: of ECE PBR VITS, Kavali, AP, India.

Abstract : In this project, we will design and implement a programmable and parameterizable Linear Feedback Shift Register (LFSR) for VLSI IC testing. The LFSR is used in circuit tests for test pattern generation (for exhaustive, pseudorandom, or pseudo-exhaustive testing) and as well as used for signature analysis. The complexity and size of SoCs are increasing at an alarming rate in modern times. There are errors that can occur during field manipulation of the device, attracting Logic Built-in-Self-Test (LBIST) over traditional ATE-based chip tests. A programmable and parameterizable LFSR can be used as test pattern generator for LBIST applications. The proposed design can generate any range of bits of vectors as per the choice of application. Also, the feedback polynomial can be parameterized to generate different length sequences. And LFSR can be configured into three different structural styles such as Fibonacci, Galois and Complete models. A Reseeding technique is introduced to leverage the LFSR to generate higher number of sequences without luring the storage requirements but testing out most of the random pattern resistant faults present in the circuit. The design is verified and analyzed in Xilinx Vivado2018.3 suite.

IndexTerms: Test Pattern Generator, Programmable, Parameterizable, External LFSR, Internal LFSR, Complete LFSR, Reseeding LFSR, Logic BIST.

INTRODUCTION:

A linear feedback shift register (LFSR) is a shift register that accepts as input a linear function of the previous state. This is usually a logical XOR function. The LFSR, like any other shift register, consists of a series of flip-flops. The state of the other bits in the cascade is affected by the tap. The Fibonacci sequence and the Galois sequence are two popular methods of tap selectors. Taps are cascaded and applied to the leftmost bit of the Fibonacci sequence. Each press is XORed with the output of a Galois construct named after the French mathematician Évariste Galois. A Linear[1] Feedback Shift Register (LFSR) is used to generate Pseudo random sequences of bits which can be used in testing a logical circuit. In this work a Test Pattern Generator is implemented which can work as internal LFSR and external LFSR based on the control signal. The main objective of this work is to increase the length of the pseudorandom sequences generated by a Test Pattern generator by combining internal and external LFSR using a control signal in a single module.[2]This paper focus on the design of a reconfigurable Linear Feedback Shift Register (LFSR) for Very Large Scale Integration (VLSI) Integrated Circuit (IC) testing. The advancement in VLSI technology have made chip testing more complicated which has lead to the popularity of Logic Built In Self-Test (LBIST) compared to Automatic Test Equipment (ATE). Logic BIST allows in-built chip testing with the help of an additional hardware structure inside the circuit. The test patterns are not applied by ATE but are generated by inbuilt testing circuits. Thus it reduces testing costs considerably. LFSR is commonly used as a test pattern generator since it is more efficient than binary counters. Reconfigurable LFSR can be used as the test pattern generator inside Logic BIST to improve the fault coverage of IC testing. [3]A Linear Feedback Shift Register (LFSR) is used to generate Pseudo random sequences of bits which can be used in testing a logical circuit. In this work a Test Pattern Generator is implemented which can work as internal LFSR and external LFSR based on the control signal. This module is implemented for different Primitive polynomials from 3bits to 11 bits in Vivado and parameters like utilization, power, and timing are analyzed.[4] This paper presents a new low-power test-data-compression scheme based on linear feedback shift register (LFSR) reseeding. A drawback of compression schemes based on LFSR reseeding is that the unspecified bits are filled with random values, which results in a large number of transitions during scan-in, thereby causing high-power dissipation. [5]In this paper, we describe a new design methodology for LFSR-based test pattern generators (TPG). Multiple seeds are produced by the TPG itself to deal with hard-to-detect faults, and this function is achieved without using a ROM to store the seeds. A reseeding logic is incorporated in the TPG, which loads new seeds into the LFSR whenever specific states are reached. [6]As the size and complexity of systems-on-chips continues to grow, the power dissipation during testing becomes very significant problem. During scan shifting, more transitions occur in the flip-flops compared to what occurs during normal functional operation. [7]The parallel reconfigurable and united architecture of LFSR is represented to improve the speed, flexibility and security of communication system and cipher algorithms. It can be reconfigured to Fibonacci LFSRs and Galois LFSRs according to different applications. The random lengths and feedback taps can be achieved to meet the demands of different applications. [8]A new methodology to increase the encoding efficiency of test compression based on linear feedback

shift registers (LFSRs) is proposed. The proposed method combines LFSR reseeding and bit fixing. [9]LFSR based PN Sequence Generator technique is used for various cryptography applications and for designing encoder, decoder in different communication channel. It is more important to test and verify by implementing on any hardware for getting better efficient result. As FPGAs is used to implement any logical function for faster prototype development, it is necessary to implement the existing design of LFSR on FPGA to test and verify the simulated & synthesis result between different lengths. The total number of random state generated on LFSR depends on the feedback polynomial.[10] Pseudorandom number generators (PRNGs) are important role in cryptography application. Hardware based random number generators become faster. Field Programming Gate Arrays (FPGA) is one of the most valuable devices in hardware industry. This paper presents multi bit linear feedback shift register (LFSR) based PRNGs circuit designed with hardware description languages (HDL).

II EXISTING METHOD:

An LFSR is a shift register that, when clocked, advances the signal through the register from one bit to the next most-significant bit. Some of the outputs are combined in exclusive-OR configuration to form a feedback mechanism. A linear feedback shift register can be formed by performing exclusive-OR on the outputs of two or more of the flip-flops together and feeding those outputs back into the input of one of the flip-flops as shown in Figure.

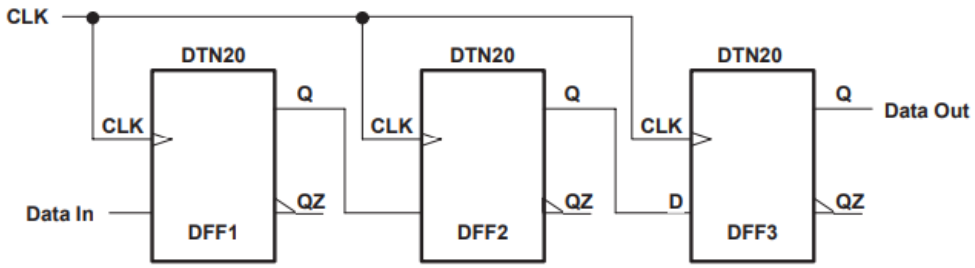


Fig.2.1 A 3-Bit Shift Register

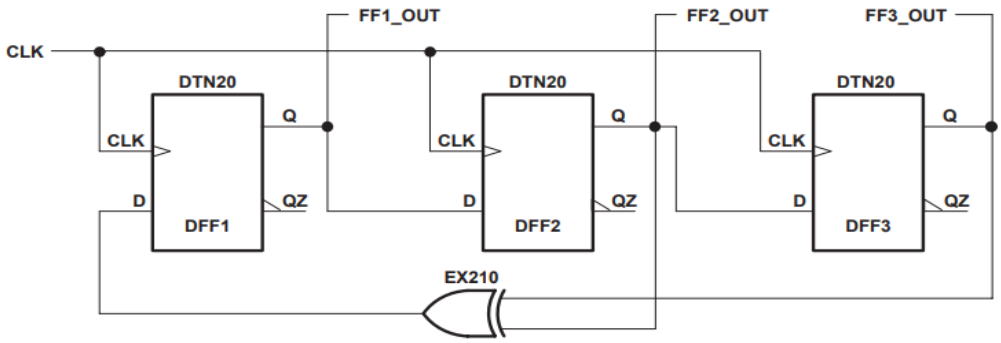


Fig2.2 Linear Feedback Shift Register

LFSRs are simple to construct and are useful for a wide variety of applications, but are often sadly neglected by designers. One of the more common forms of LFSR is formed from a simple shift register with feedback from two or more points, or taps, in the register chain.

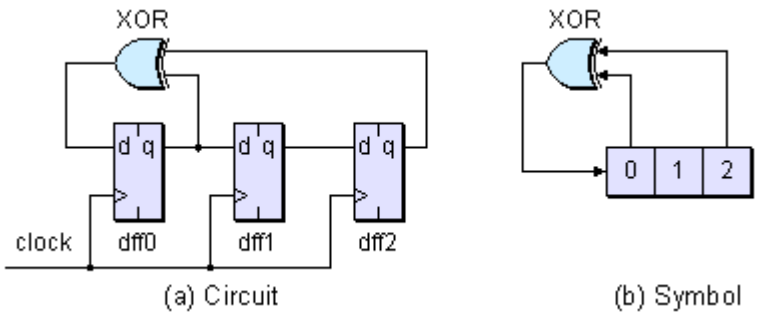


Fig. LFSR with XOR feedback path.

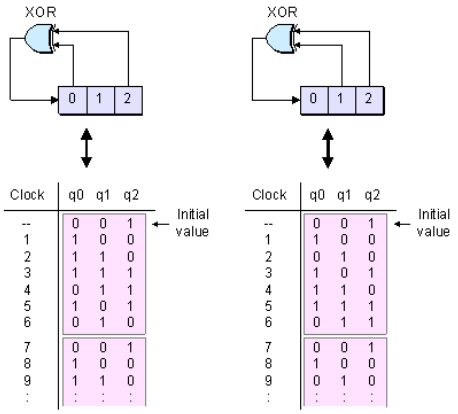


Fig2.3 Comparison of alternative tap selections.

A binary field with 'n' bits can assume 2^n unique values, but a maximal-length LFSR with 'n' register bits will only sequence through (2^n – 1) values. This is because LFSRs with XOR feedback paths will not sequence through the value where all the bits are 0, while their XNOR equivalents will not sequence through the value where all the bits are 1 shown in below fig.

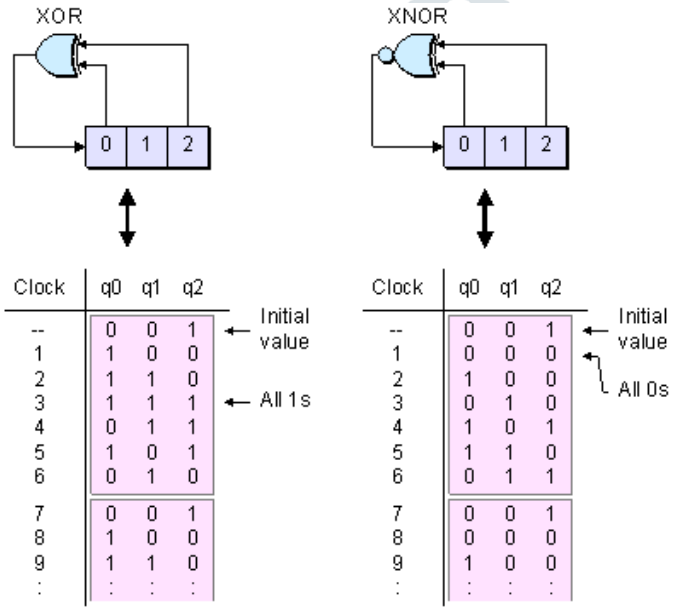


Fig2.4 Comparison of XOR versus XNOR feedback paths.

# of Bits	Length of Loop		Taps
2	3	*	[0,1]
3	7	*	[0,2]
4	15		[0,3]
5	31	*	[1,4]
6	63		[0,5]
7	127	*	[0,6]
8	255		[1,2,3,7]
9	511		[3,8]
10	1,023		[2,9]
11	2,047		[1,10]
12	4,095		[0,3,5,11]
13	8,191	*	[0,2,3,12]
14	16,383		[0,2,4,13]
15	32,767		[0,14]
16	65,535		[1,2,4,15]
17	131,071	*	[2,16]
18	262,143		[6,17]
19	524,287	*	[0,1,4,18]
20	1,048,575		[2,19]
21	2,097,151		[1,20]
22	4,194,303		[0,21]
23	8,388,607		[4,22]
24	16,777,215		[0,2,3,23]
25	33,554,431		[2,24]
26	67,108,863		[0,1,5,25]
27	134,217,727		[0,1,4,26]
28	268,435,455		[2,27]
29	536,870,911		[1,28]
30	1,073,741,823		[0,3,5,29]
31	2,147,483,647	*	[2,30]
32	4,294,967,295		[1,5,6,31]

Table 5 Taps for maximal length LFSRs with 2 to 32 bit

PROPOSED METHOD:

The research adopts the following methodology as shown in figure .1. Investigating into the black box, we can observe 3 types of LFSR structures and reseeding structure.

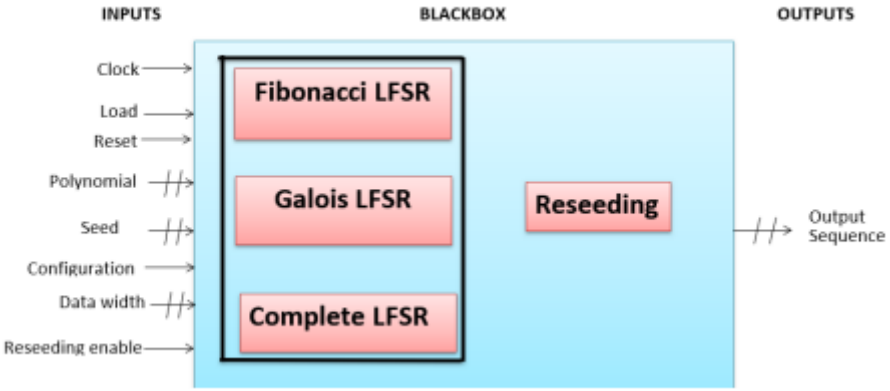


Fig 3.1 Block diagram of proposed work.

A Fibonacci LFSR is also called as standard LFSR or External LFSR. In the fig.2 The LFSR output bit is the rightmost bit. The output bits are bound to the XOR in sequential order and then returned to the leftmost bit. The rightmost bit string is the output stream. The input is affected by the taps, which are the bits of the LFSR state.

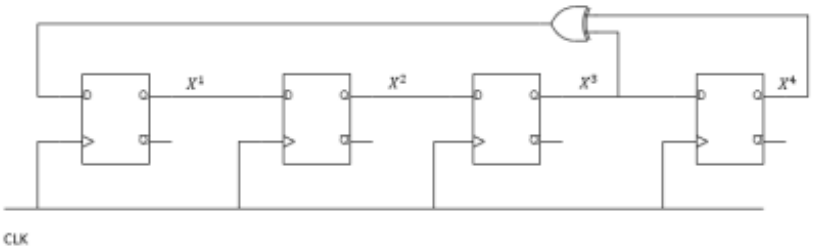


Fig 3.2 Fibonacci LFSR for polynomial x4 + x3 + 1

Modular LFSR is faster in terms of execution compared to the Standard LFSR.

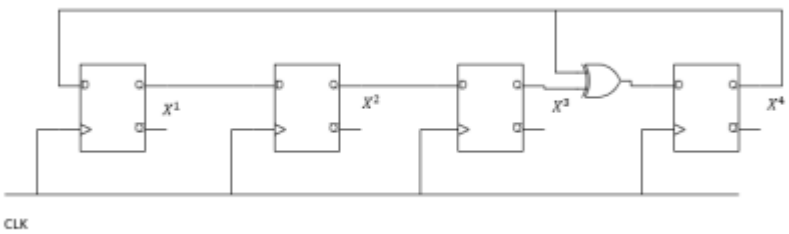


Fig 3.3 Galois LFSR for polynomial x4 + x3 + 1

An LFSR that has been modified to include all null states is referred to as a complete LFSR. 2n states are possible for n bit LFSR with primitive polynomials. For an all-zero input, the XOR gate feeds and shifts a zero value to the LFSR's input, so the presence of an all-zero state can lock the LFSR. The XOR gate is inserted in the final stage of the LFSR, as shown in Figure 4, and the NOR gate with n1 input is used as the zero-detector.

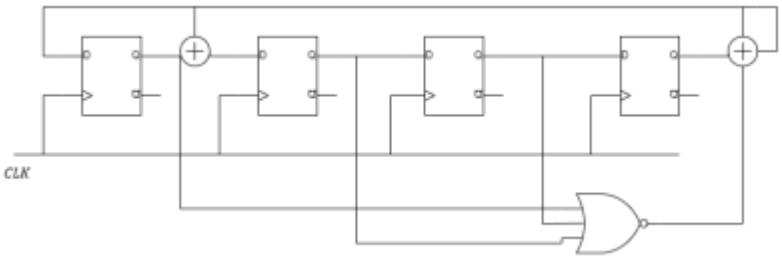


Fig.3.4 Complete LFSR for polynomial x4 + x3 + 1

A reseeding type LFSR is a technique which enables a conventional LFSR to generate the sequences with multiple seeds which help not only in reduction of area overhead to store the patterns, but to test the Random Pattern Resistant Faults (RPR). As illustrated in the Fig.5, a Reseeding LFSR takes the input seed and update the seed value by its own at every cycle hence

generating sequences from multiple seeds in a single run compared to a conventional LFSR which have the capability to generate sequences from a single seed value.

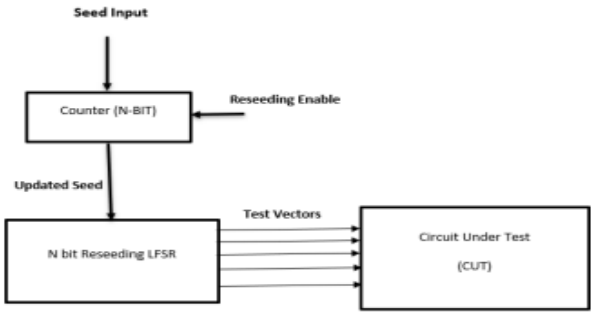


Fig.3.5 Architecture for Reseeding LFSR

Polynomials that produce maximal length ($2^n - 1$) sequence are called primitive polynomials.
The most commonly used linear function of single bits is exclusive-or (XOR). Thus, an LFSR is most often a shift register whose input bit is driven by the XOR of some bits of the overall shift register value.

The mathematics of a cyclic redundancy check, used to provide a quick check against transmission errors, are closely related to those of an LFSR.[1] In general, the arithmetics behind LFSRs makes them very elegant as an object to study and implement. One can produce relatively complex logics with simple building blocks. However, other methods, that are less elegant but perform better, should be considered as well.

RESULT:

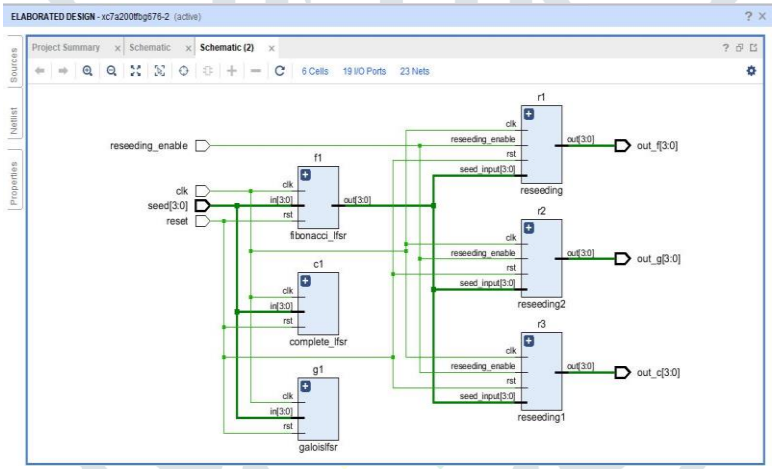


Fig:11 RTL Schematic

The clock, seed and reset are connected to the Fibonacci, complete and galois lfsr's as input.

The Fibonacci output is connected to the reseeding enable it comes from the I/O pads.

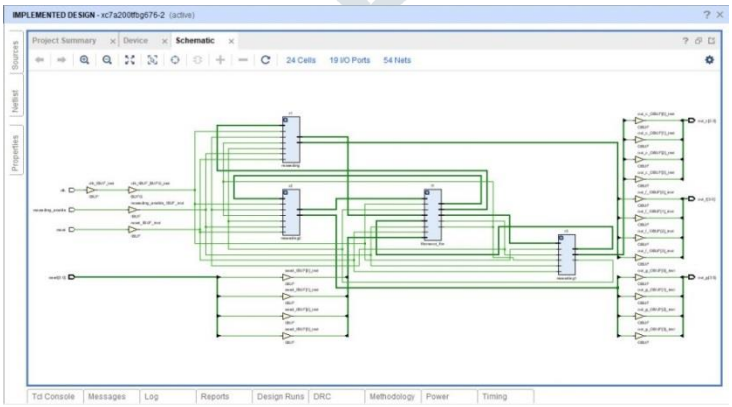


Fig:12 Technology Schematic

The above figure is same as previous figure shows that input and output powers . schematic technology. We can see the number of connected to the circuit

Simulation results:

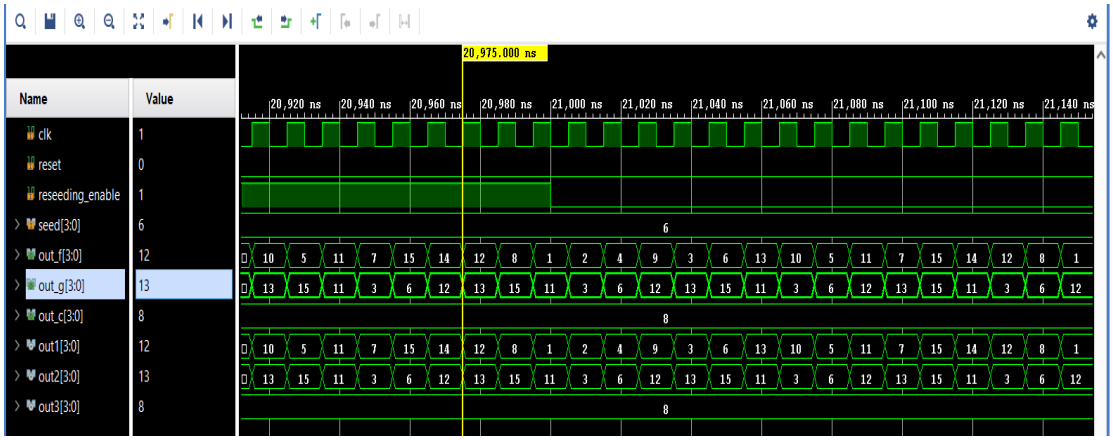


Fig:13 Simulation results for proposed Algorithm

This fighter shows RTL schematic of the above result

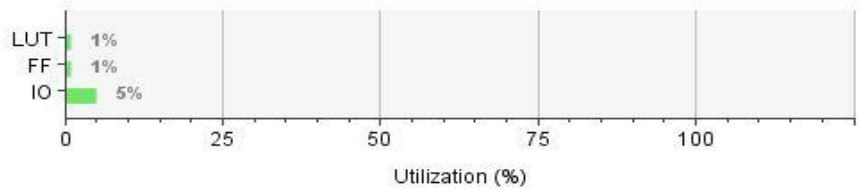
Area, delay and power

Area: The resource LUT is taken as 54(utilization)is available 134600 and the utilization% is 0.04.

We can see the graph LUT and FF is same but IO is increased in compared to those resources.

In this proposed method the area is 54 LUT

Resource	Utilization	Available	Utilization %
LUT	54	134600	0.04
FF	60	269200	0.02
IO	19	400	4.75

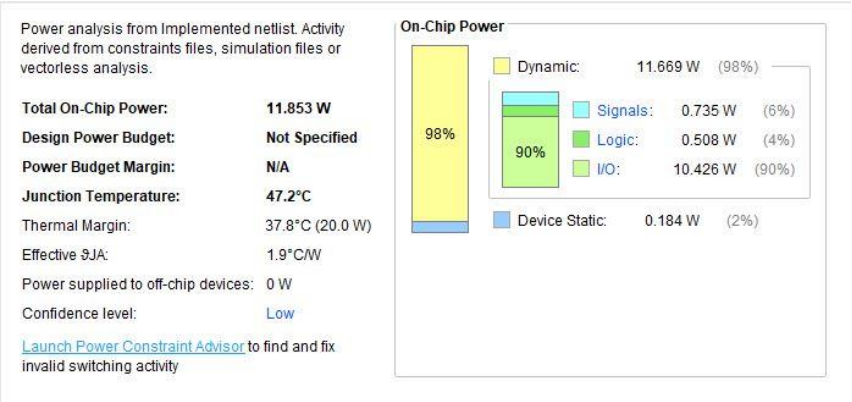


Delay: The proposed method delay is 5.33 ns**.**

```
------------------------------------------------------------------------------------

Max Delay Paths
--------------------------------------------------------------------------------------
Slack:                   inf
  Source:                r2/11/out_reg[3]_C/C
                           (rising edge-triggered cell FDCE)
  Destination:           out_g[3]
                           (output port)
  Path Group:            (none)
  Path Type:             Max at Slow Process Corner
  Data Path Delay:       5.333ns  (logic 2.922ns (54.785%)  route 2.411ns (45.215%))
  Logic Levels:          3  (FDCE=1 LUT3=1 OBUF=1)

    Location             Delay type                Incr(ns)  Path(ns)    Netlist Resource(s)
  -------------------------------------------------------------------    -------------------
    SLICE_X6Y108         FDCE                         0.000     0.000 r  r2/11/out_reg[3]_C/C
    SLICE_X6Y108         FDCE (Prop_fdce_C_Q)         0.433     0.433 r  r2/11/out_reg[3]_C/Q
                         net (fo=2, routed)           0.775     1.208    r2/11/out_reg[3]_C_n_0
    SLICE_X4Y108         LUT3 (Prop_lut3_I2_O)        0.105     1.313 r  r2/11/out_g_OBUF[3]_inst_i_1/O
                         net (fo=3, routed)           1.636     2.949    out_g_OBUF[3]
    P25                  OBUF (Prop_obuf_I_O)         2.384     5.333 r  out_g_OBUF[3]_inst/O
                         net (fo=0)                   0.000     5.333    out_g[3]
    P25                                                       r  out_g[3] (OUT)
  -------------------------------------------------------------------    -------------------
```



Power: The proposed method the total on chip power is 11.853w.

CONCLUSION:

In this work, we have designed and verified a programmable and parameterizable linear feedback shift register. The schematic is verified and simulations are performed in Xilinx Vivado2018.3 suite. The power consumption analysis has been performed in Xilinx Vivado suite three types of structures such as Fibonacci, Galois and complete LFSR modules are designed and verified. Also the data width, polynomial, seed value, configuration of the LFSR was parameterizable and programmable. The LFSR include reseeding scheme, which helps in generating more test patterns without a higher effective area overhead as conventional methods. The complete LFSR consumes less power whereas the Galois LFSR contributes to major power consumption.

REFERENCES:

[1] P. S. Dilip, G. R. Somanathan, and R. Bhakthavatchalu, "Reseeding LFSR for Test Pattern Generation," in 2019 International Conference on Communication and Signal Processing (ICCSP), Apr. 2019, pp. 0921–0925.
[2] C. S. Vikranth, D. Mohammad, G. R. Somanathan, and R. Bhakthavatchalu, "Analysis of a Novel Reseeding Pattern Generator," in 2021 2nd International Conference on Smart Electronics and Communication (ICOSEC), Oct. 2021, pp. 676–682.
[3] K. N. Devika and R. Bhakthavatchalu, "Design of reconfigurable LFSR for VLSI IC testing in ASIC and FPGA," in 2017 International Conference on Communication and Signal Processing (ICCSP), Apr. 2017, pp. 0928–0932.
[4] M. Bushnell and V. Agrawal, Essentials of Electronic Testing for Digital, Memory and Mixed-Signal VLSI Circuits. Springer Science & Business Media, 2004.
[5] A. Sireesha, Mohammad Hayath Rajvee, "Advanced SOC Tracing Technique", International Journal of Engineering Research and Technology(IJERT), ISSN:2278-0181, Vol.1, no.6, pp.1-7, Aug-2012.
[6] S. Dhingra, "Comparison of LFSR and CA for BIST," undefined, 2005, Accessed: Nov. 19, 2021. [Online]. Available: https://www.semanticscholar.org/paper/Comparison-ofLFSR-and-CA-for-BIST-Dhingra/2fc0ca00f0d2d35041ea88b157d18267e7f4f4ca
[7] D. Datta, B. Datta, and H. S. Dutta, "Design and implementation of multi-bit LFSR on FPGA to generate pseudorandom sequence number," in 2017 Devices for Integrated Circuit (Dev IC), Mar. 2017, pp. 346–349.
[8] D. Jeevalakshmi, MD. Hayath Rajvee, "High Speed Floating Point multiplier Units Using Leading Zero Counting And Anticipation Logic For Addition And Subtraction" International Journal Of Computer Science And Communication Networks, ISSN:2249-5789,Vol.2(5),pp.573-576,2012.
[9] A. K. Panda, P. Rajput, and B. Shukla, "FPGA Implementation of 8, 16 and 32 Bit LFSR with Maximum Length Feedback Polynomial Using VHDL," in 2012 International Conference on Communication Systems and Network Technologies, May 2012, pp. 769–773.
[8] H.-S. Kim and S. Kang, "Increasing encoding efficiency of LFSR reseeding-based test compression," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 25, no. 5, pp. 913–917, May 2006.
[10] W. Li and X. Yang, "A Parallel and Reconfigurable United Architecture for Fibonacci and Galois LFSR," in 2015 7th International Conference on Intelligent Human-Machine Systems and Cybernetics, Aug. 2015, vol. 1, pp. 203–206.
[11]https://www.xilinx.com/support/documentation/applicat ion_notes/xapp052.pdf[11] "LFSR Reference – M-Sequence, Linear Feedback Shift Register, Feedback Taps for Maximal Length Sequences." http://in.ncu.edu.tw/~ncume_ee/digilogi/prbs.htm (accessed Nov. 19, 2021).