**16-Bit Pseudo Random number**

**Generator (PRNG) that generates Fibonacci series**

**Course project**

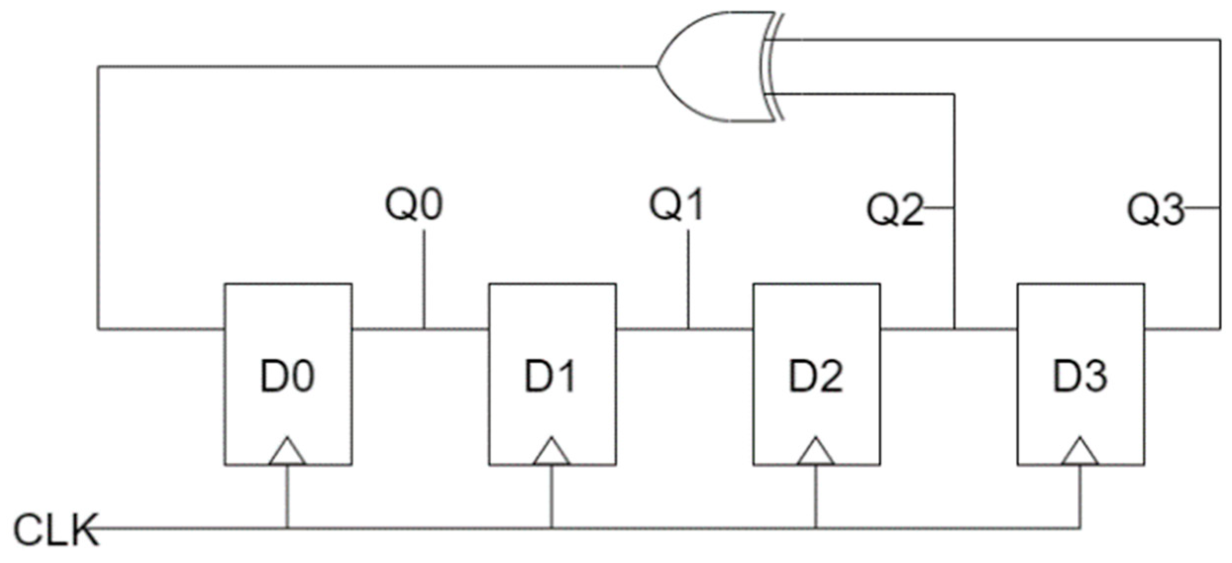
|  |  |
| --- | --- |
| Name | USN |
| Sarvesh S Inamdar | 01FE20BEC234 |
| Sakshi R Halbhavi | 01FE20BEC233 |

***Introduction***

* Logic built-in self-test (LBIST) is a form of built in self-test (BIST) in which the logic inside a chip can be tested on-chip itself without any expensive Automatic Test Equipment (ATE). A BIST engine is built inside the chip.
* In general, LBIST uses Pseudo Random Number Generator (PRNG) which is an application of LFSR.
* A Linear Feedback Shift Register (LFSR) is used to generate the inputs to the device’s internal scan chain, initiate a functional cycle to capture the response of the device, and then compact the captured response using a Multiple Input Signature Register (MISR).
* The compacted response that comes out of the MISR is called the signature output.
* Any corruption in the output signature indicates a defect in the device.
* Linear Feedback Shift Register (LFSR) are widely used to design PRNG with the seed implemented to the D-Flip Flop.
* Typically a Standard form LFSR is used for generating pseudo random patterns which acts as the test input.

**LFSR [Linear Feedback Shift Register]**

* The LFSR model is that of a finite state machine comprising storage elements and modulo two adders (XOR gates) connected in feedback loops.

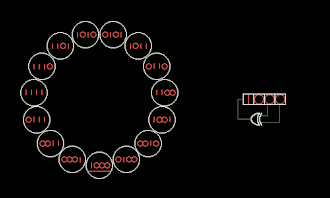


***Figure****:- A basic 4 Bit LFSR circuit consisting of D flipflops*

* Similarly, we have to design a 16 Bit LFSR which generates Fibonacci series.

**4 Bit Fibonacci LFSR**

* A Fibonacci LFSR (Linear Feedback Shift Register) is a type of shift register used for generating pseudorandom binary sequences. The term "Fibonacci" refers to the fact that the feedback function used in the shift register is based on the Fibonacci sequence.
* In a Fibonacci LFSR, the feedback function is based on the exclusive-OR (XOR) operation between specific flip-flop outputs.



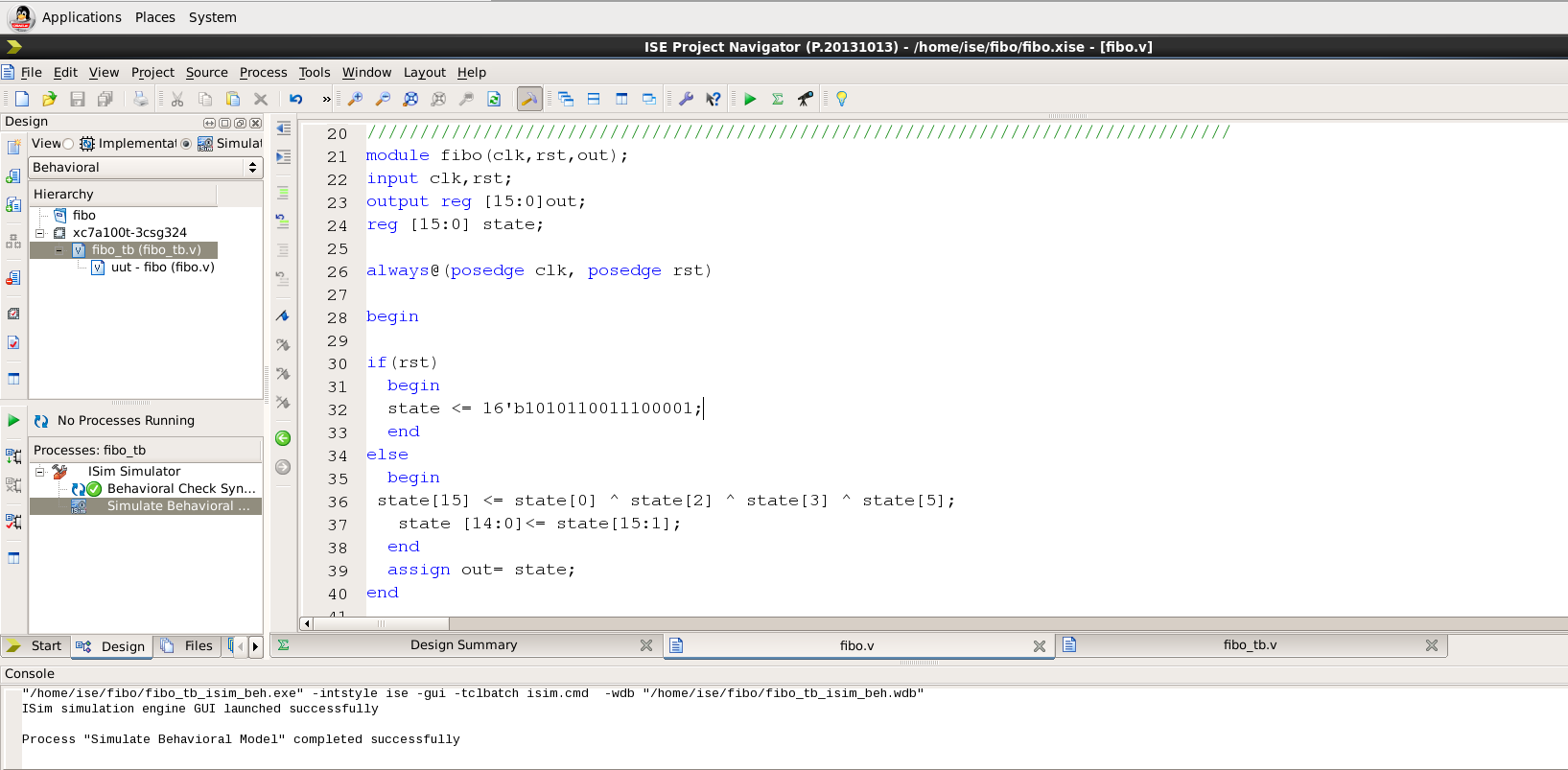
***Figure****:-* *A 4-bit Fibonacci LFSR with its state diagram. The XOR gate provides feedback to the register that shifts bits from left to right.*

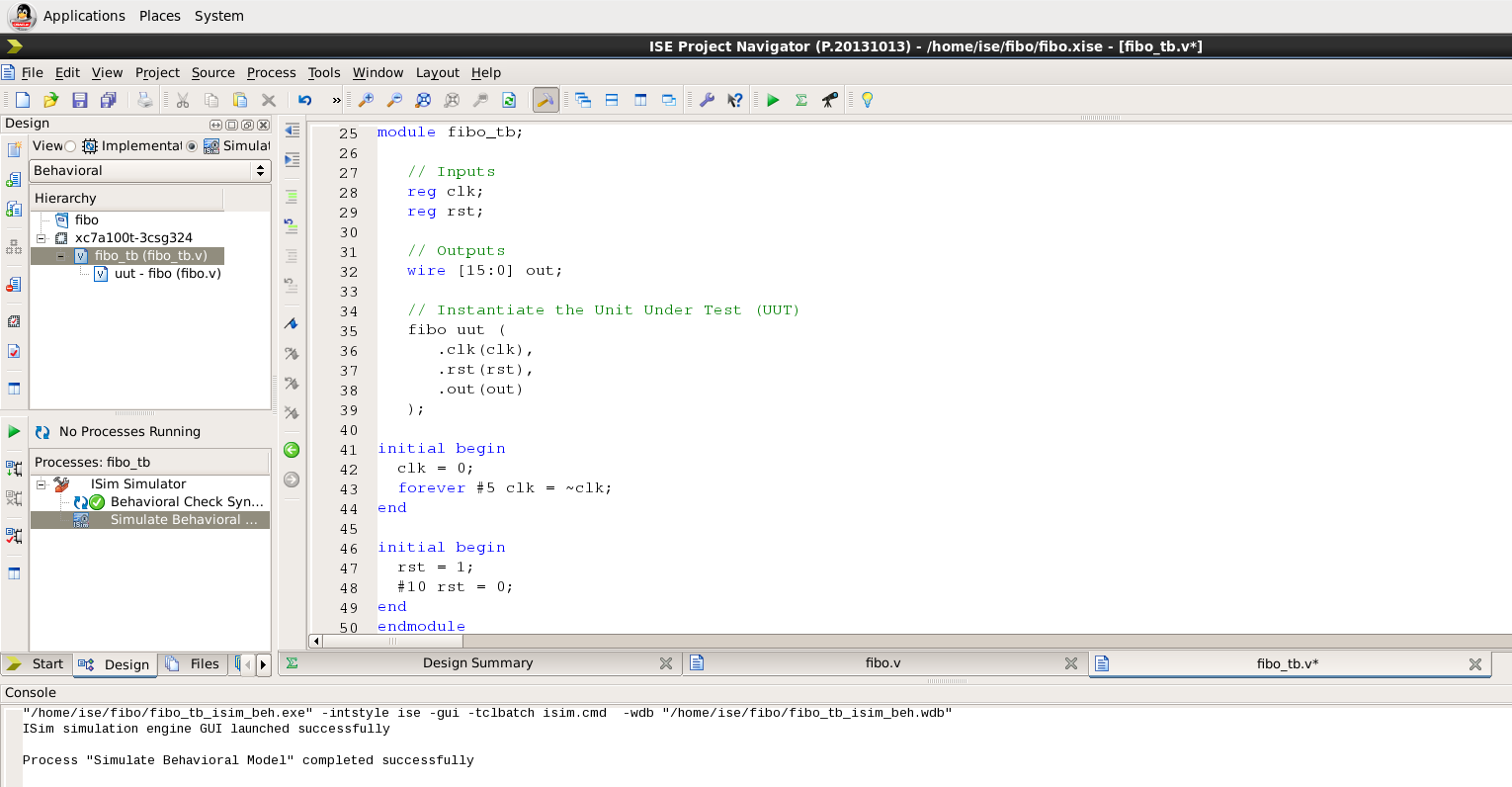
* The maximal sequence consists of every possible state except the "0000" state.
* An LFSR with a well-chosen feedback function can produce a sequence of bits which appears random and which has a very long cycle

**16 Bit Fibonacci LFSR**

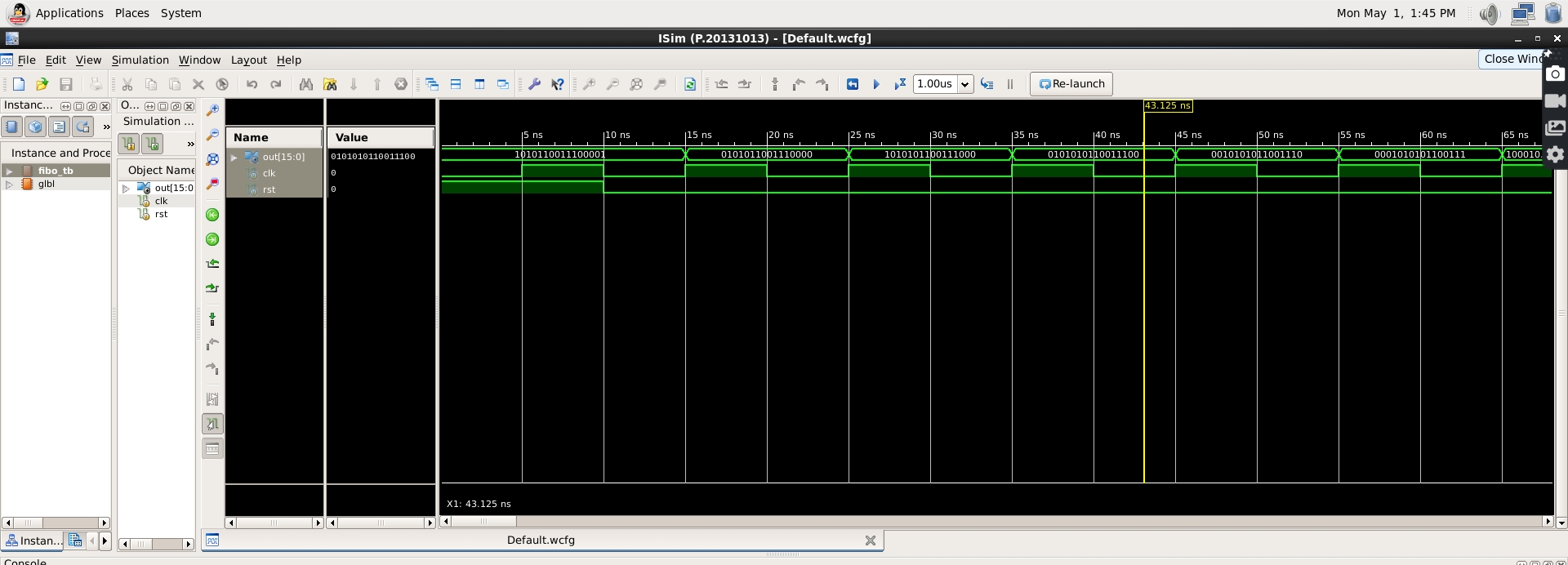
* A 16-bit Fibonacci LFSR is a sequential logic circuit consisting of 16 flip-flops and a feedback function that is based on the Fibonacci sequence.
* The LFSR generates a pseudorandom binary sequence that repeats after 2^n-1 clock cycles.(n=16)
* The basic building block of a 16-bit LFSR is a D-type flip-flop, which stores a single bit of data and can be used to delay a signal by one clock cycle.
* The 16 flip-flops are connected in series, forming a shift register that can be shifted to the right on each clock cycle
* The feedback function for a 16-bit Fibonacci LFSR is based on a tap sequence that is derived from the Fibonacci sequence
* The tap sequence for a 16-bit LFSR is 1000100000010001, which corresponds to the Fibonacci number 65537 (which has a binary representation of 10000000000000001).
* To generate the next bit of the pseudorandom sequence, the values stored in the 16 flip-flops are XORed together according to the tap sequence
* Specifically, the outputs of the 16th, 14th, 13th, and 11th flip-flops are fed into a 4-input XOR gate, and the output of the XOR gate is fed back into the input of the first flip-flop.
* The bit positions that affect the next state are called the taps. In the previous diagram the taps are [16,14,13,11].
* The arrangement of taps for feedback in an LFSR can be expressed in finite field arithmetic as a polynomial mod is given by m(x) = 1 + h1x2 + h2x2 + :::::::hn-1xn-1 + hnxn.
* This means that the coefficients of the polynomial must be 1's or 0's. This is called the feedback polynomial
* For example, if the taps are at the 16th, 14th, 13th and 11th bits (as shown), the feedback polynomial is x16 + x14 + x13 + x11 + 1 = 0.

**Verilog code for 16 Bit Fibonacci LFSR**

**Testbench**

****

**Simulated Output**

****