Question: 05
DATE: October 6, 2017
Sabbir Ahmed

1 Background

Implement the linear feedback shift register shown on `https://en.wikipedia.org/wiki/Linear_feedback_shift_register`.

The core 16-bit register must be implemented with a 16-bit signal. The the shift should be implemented using a concatenation operator to right-shift with the left-most bit loaded from the output of the xor gates. Verify that linear-feedback shift-register produces all possible states of a 16-bit register excluding the zero state before repeating its seqeunce.

To do this, use a Verilog Test Bench to save the output to a file every clock cycle for at least two full cycles of the pattern. Write C/Python/Java code to verify that the pattern repeats identically at least twice and that no number is repeated or missed during one cycle of the pattern. You should submit all code and output files.

2 Implementation

The odd and even indices of 'x' were swapped in a simple while loop to create the output 'u'.

The module implementation along with its testbench can be found in the `scripts' directory. A sample of the waveform generated is provided:

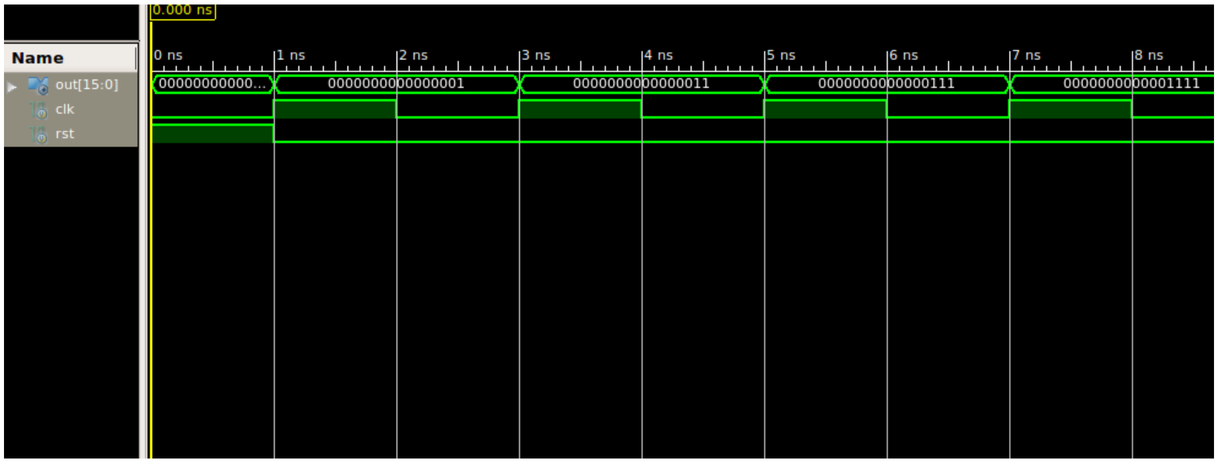

Figure 1: Waveform Generated from Part 4 Test Bench

The following output is dumped from the test bench as well, demonstrating the swapping of the bits:

```
x = 0101010101010101010101010101010101010101010101010101010101010101 when clk = 1
u = 1010101010101010101010101010101010101010101010101010101010101010 when clk = 1
x = 1101110111011101110111011101110111011101110111011101110111011101 when clk = 1
u = 1110111011101110111011101110111011101110111011101110111011101110 when clk = 1
```

```
x = 110111011101110111011101110111011101110111011101 when clk = 0
u = 111011101110111011101110111011101110111011101110 when clk = 0
x = 001000100010001000100010001000100010001000100010 when clk = 1
u = 000100010001000100010001000100010001000100010001 when clk = 1
```