Indian Institute of Technology (IIT-Kharagpur)

SPRING Semester, 2016
COMPUTER SCIENCE AND ENGINEERING

Switching Circuits Laboratory

Lab–Module E

Verilog Programming of a 4-bit Loadable Linear Feedback Shift Register (LFSR) with all Zero State

Objective of the Design: The objective is to design a synchronous sequential circuit using 4 D-FlipFlops, which when loaded by an initial vector (called seed, and which is non-zero) cycles through all the 15 non-zero binary vectors before returning to the initial vector (the seed). Thus this can be used as a modulo 15 counter, without using any adder! The only combinational gates required are XORs. They form a very useful structure, commonly called as *Linear Feedback Shift Registers* (LFSRs) and have extensive use in digital circuits, communication theory, etc.

Details of the Circuit

The design is a synchronous sequential circuit using D-FlipFlops as the memory elements. The design has 4 D-FlipFlops (DFFs) connected as shown in the **Fig. 1**. The 4-stages of the sequential circuit can be loaded by an initialization *seed* by using 4 multiplexers as shown in the diagram.

The select line of the multiplexer (*sel*) when made 0, the seed vector bits are passed into the inputs of the DFFs. For all the 3 right DFFs, when the sel bit is 1, the output of the left DFF is shifted in to the input of the DFF. For the left-most DFF (as there nothing to its left!), the bit to be shifted in when the sel bit is 1, is computed by XORing the outputs of the right two DFFs.

The wire names are clearly shown in the diagram to help your design. Also note that DFFs will have an *asynchronous reset*, rst and the input clock is to be divided and made slow to see the outputs in the LEDs of the Nexys-3 Boards.

For this proceed in the following steps:

1. Design the circuit using Verilog-HDL in a modular fashion. Your design should have a **top-level** module which instantiates separate sub-modules for:

(a) D-FlipFlop (DFF)
(b) Multiplexer (MUX)
(c) Clock-Divider (CLKDiv)

The inputs of the design are: 1) 4-bit seed 2) rst 3) sel, and 3) clk (generated internally in the board and later divided by your CLKDiv module). The output will be a 4-bit state of the 4-DFFs, denoted by *state[3:0]*={w2,w3,w4,w5}.

2. Write an UCF file for the design, ensuring that the 4-bit seed, rst, and sel inputs are to be provided from the switches of your board. The output state is to be visualised in the LEDs of your board.

3. Download the design's bit file onto the FPGA board and show a demonstration of the evolution of the LFSR. Load the LFSR by the non-zero seed `1111`, and show that it comes back to the same state after 15 clock ticks. The state transitions of the LFSR are as follows: `1111` → `0111` → `0011` → `0001` → `1000` → `0100` → `0010` → `1001` → `1100` → `0110` → `1011` → `0101` → `1010` → `1101` → `1110` → `1111`

4. Note that the sequential machine has a fixed point, when the input is `0000`. In the class, we discussed on how to convert this circuit to also include this state in the transition of the machine. Augment your circuit, and the corresponding Verilog code to realize the modified counter.

This counter is also called *De-Bruijn* Sequence Generator or counter.

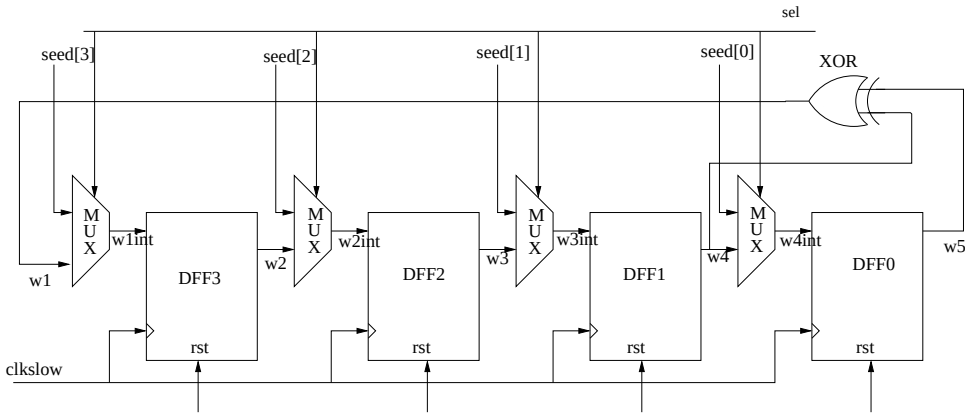

Figure 1: The 4-bit LFSR