ECE-111: Advanced Digital Design Project: Homework 4

Designs: Linear Feedback Shift Register (LFSR), Barrel Shifter, Gray to Binary Code Convertor

Overview

There will be three parts for this homework. In **homework-4a** you will design a synthesizable SystemVerilog Model of a Linear Feedback Shift Register (LFSR). In **homework-4b**, you will develop a synthesizable SystemVerilog code for a Barrel Shifter. In **homework-4c** you will design a synthesizable SystemVerilog Model of a Gray to Binary Code Convertor.

We have provided a folder called **Lab4.zip** which contains the followings:

Homework-4a:

1. lfsr.sv partial design template code
2. lfsr_testbench.sv full code

Note:

1. It is not mandatory to use design template code provided in the lab folder. Students can implement their design module from scratch without referring to template code as long as the primary port list is matching with what is given. This is to ensure testbench is compatible with design.
2. For learning purposes, students can change the stimulus in the initial block in the testbench file.

Homework-4b:

1. barrel_shifter.sv partial design template code
2. barrel_shifter_testbench.sv full code
3. mux_2x1_behavioral.sv

Note:

1. It is not mandatory to use design template code provided in the lab folder. Students can implement their design module from scratch without referring to template code as long as the primary port list is matching with what is given. This is to ensure testbench is compatible with design.

2. For learning purposes, students can change the stimulus in the initial block in the testbench file.

Homework-4c:

1. gray_to_binary_code_convertor.sv template code for design
2. gray_to_binary_code_convertor_testbench.sv full code

Note:

1. For learning purposes, students can change the stimulus in the initial block in the testbench file.

Assignment Tasks

This assignment requires you to complete the following tasks:

Recommended Task:

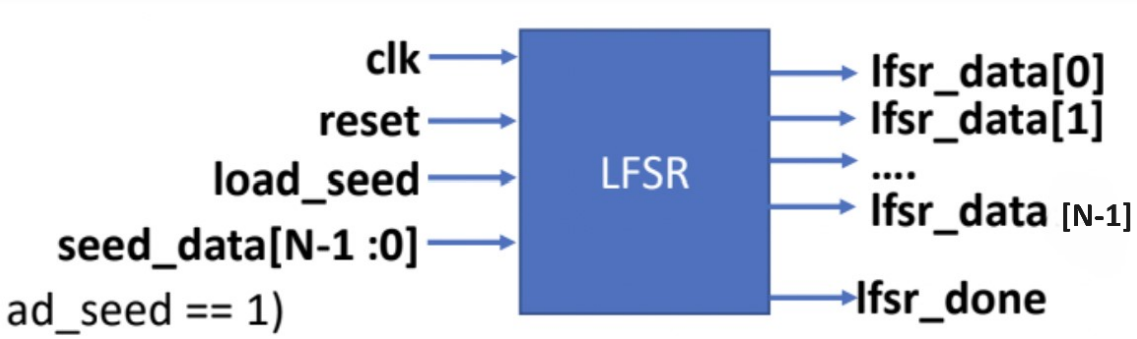
- Review Discussion 4 video and slides for a more detailed overview before starting the assignment.

For homework-4a:

- **Develop SystemVerilog RTL model for N-bit Linear Feedback Shift Register (LFSR):**
  - Value N can be either 2,3,4,5,6,7,8 (we expect you to use a parameter).
  - In the same SystemVerilog module implement max-length LSFR pseudo-random generation polynomials to support each of the above-mentioned N values.
  - Use xor implementation for LFSR SystemVerilog RTL model development.
  - Synthesize LFSR and run simulation using LSFR testbench provided
    - Testbench provided for 4-bit LSFR. In testbench file parameter N can be set to some other value say 5 and run simulation.
  - Review synthesis results (resource usage and RTL netlist/schematic).
  - Review input and output signals in simulation waveform.
  - Assume below mentioned (in block diagram) primary port names and SystemVerilog RTL module name as lfsr.

- **Primary Ports for LFSR module**
  - Input clk (clock).
  - Input reset (asynchronous reset and negedge signal).
  - Input load_seed (synchronous and active high signal).
  - Input seed_data(initial seed data gets loaded when load_seed == 1).
  - Output lfsr_data (output of each Flipflop on shift register).
  - Output lfsr_done (once N-bit pattern generation is completed and before the repetition of pattern start, generate 1 cycle pulse for lfsr_done).

- **Here is the block diagram:**



Bits(n)	Feedback Polynomial	Max Length / Period ($2^N - 1$)
2	$x^2 + x^1 + 1$	3
3	$x^3 + x^2 + 1$	7
4	$x^4 + x^3 + 1$	15
5	$x^5 + x^3 + 1$	31
6	$x^6 + x^5 + 1$	63
7	$x^7 + x^6 + 1$	127
8	$x^8 + x^6 + x^5 + x^4 + 1$	255

For Homework-4b:

- **Develop SystemVerilog RTL model for 4-bit Barrel Shifter using 2x1 Mux behavioral model.**
  - The same Barrel Shifter RTL model should support logical shift left, logical shift right, rotate left and rotate right operations.
  - Use 2x1 Mux behavioral RTL model (will be provided) to design barrel shifter (Do not use 4x1 Mux model).
  - Synthesize Barrel Shifter and review synthesis results (resource usage and RTL netlist/schematic).

- Run simulation using the testbench provided and review the waveform to confirm:
  - left shift, right shift, left rotate and right rotate operations of the barrel shifter RTL model.
- Assume below mentioned (in block diagram) primary port names and the SystemVerilog RTL module name as barrel shifter.

- **Primary Ports for Barrel Shifter**
  - select : to select between the shift or rotate operation.
    - select == 0 for shift operation
    - select == 1 for rotate operation
  - din : 4 bit input data
  - dout : 4-bit output data
  - direction : move bits in either left or right direction.
    - direction == 0, move bits to the right.
    - direction == 1, move bits to the left.
  - shift_value : bit positions to be shifted
    - shift_value == 00, no shift operations
    - shift_value == 01, move bits by 1-bit position
    - shift_value == 10, move bits by 2-bit positions
    - shift_value == 11, move bits by 3-bit positions

- **Block diagram for barrel shifter:**

N-Bit Input Data	Number of 2x1 Muxes : N x $\log_2$ N	Number of multiplexing stages (number of chains)
2	2 x $\log_2$ 2 = 2	1 chains (each chain with 2 2x1 Muxes)
4	4 x $\log_2$ 4 = 8	2 chains (each chain with 4 2x1 Muxes)
8	8 x $\log_2$ 8 = 24	3 chains (each chain with 8 2x1 Muxes)
16	16 x $\log_2$ 16 = 64	4 chains (each chain with 16 2x1 Muxes)
32	32 x $\log_2$ 32 = 160	5 chains (each chain with 32 2x1 Muxes)

- **Ensure simulation results of Barrel Shifter RTL Model is as per the below mentioned truth table**

select	direction	shift_value	Operation
0	0 or 1	00	No Shift operation
0	0	01	**LSR>>1** (Logical Shift Right by 1-bit position)
0	0	10	**LSR>>2** (Logical Shift Right by 2-bit position)
0	0	11	**LSR>>3** (Logical Shift Right by 3-bit position)
0	1	01	**LSL<<1** (Logical Shift Left by 1-bit position)
0	1	10	**LSL<<2** (Logical Shift Left by 2-bit position)
0	1	11	**LSL<<3** (Logical Shift Left by 3-bit position)
1	0 or 1	00	No Rotate operation
1	0	01	**ROR#1** (Rotate Right by 1-bit position)
1	0	10	**ROR#2** (Rotate Right by 2-bit position)
1	0	11	**ROR#3** (Rotate Right by 3-bit position)
1	1	01	**ROL#1** (Rotate Left by 1-bit position)
1	1	10	**ROL#2** (Rotate Left by 2-bit position)
1	1	11	**ROL#3** (Rotate Left by 3-bit position)

Note: For Barrel Shifter testbench already has stimulus for each of the rows above in truth table

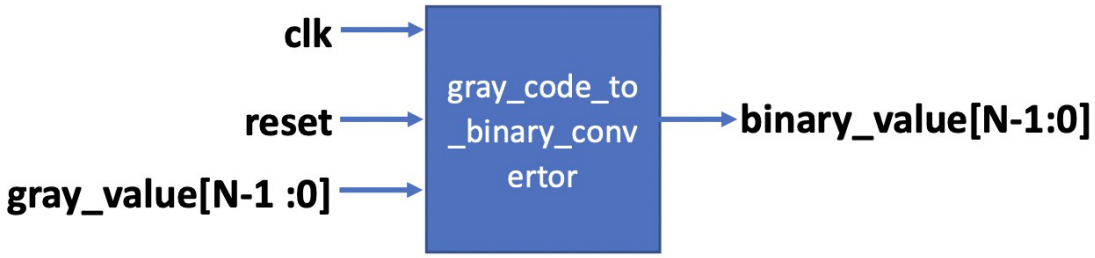
For homework-4c:

- **Develop SystemVerilog RTL model for N-bit Gray to Binary Code Converter**
  - Use the function to implement gray to binary code conversion and add a flipflop at the output of the converter.
  - Synthesize the design and run simulation using the testbench provided.
  - Review synthesis results (resource usage and RTL netlist/schematic).
  - Testbench is provided for 4-bit gray to binary value conversion.
  - Review input and output signals in simulation waveform.
  - Assume below mentioned primary port names and SystemVerilog RTL module name as gray_code_to_binary_convertor.

- **Primary Ports for gray_code_to_binary_convertor module**
  - Input clk : posedge clock
  - Input rstn : reset should be an asynchronous negedge signal.
  - Input gray_value : N-bit input gray code signal.
  - Output binary_value : N-bit output converted binary value signal.

Note: Add a flipflop at the output of gray to binary code convertor logic.

- Block Diagram:



Submission Requirements

Submit a report on Gradescope in PDF format which includes the following :

For homework-4a:

- SystemVerilog design code.
- Synthesis resource usage and schematic generated from RTL netlist viewer.
- Simulation and transcript snapshots and explain the simulation results to confirm that the RTL model developed works as a LFSR.
- Post-Mapping schematic is optional to submit.
- Explanation of FPGA resource usage in the report is not required.

For homework-4b:

- SystemVerilog design code.
- Synthesis resource usage and schematic generated from RTL netlist viewer.
- Simulation and transcript snapshots and explain the simulation results to confirm that the RTL model developed works as a Barrel Shifter.
- Post-Mapping schematic is optional to submit.
- Explanation of FPGA resource usage in the report is not required.

For homework-4c:

- SystemVerilog design code.
- Synthesis resource usage and schematic generated from RTL netlist viewer.
- Simulation and transcript snapshots and explain the simulation results to confirm that the RTL model developed works as a gray to binary code converter.
- Post-Mapping schematic is optional to submit.
- Explanation of FPGA resource usage in the report is not required.