ECE-111: Advanced Digital Design Project: Homework 3

Designs	4-bit Johnson Counter, Universal Shift Register
Deadline	October 23, 2024 at 11:59pm
Max. late days	2 (20% grade reduction per day)

Overview

In this homework, you will design a 4-bit Johnson Counter and a Universal Shift Register.

There will be two parts for this homework. In **homework-3a** you will design a synthesizable SystemVerilog model of a 4-bit Johnson Counter. In **homework-3b**, you will develop a synthesizable SystemVerilog code for a Universal Shift Register.

We have provided a folder called **Lab3.zip** which contains the following:

Homework-3a:

1. `johnson_counter.sv` partial design template code
2. `johnson_counter_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 asked for in the assignment tasks. This is to ensure that the testbench is compatible with the design.
2. Initial `load_cnt` signal value is set to 4'b0000 in testbench code provided.
3. For learning purposes, students can update `johnson_counter_testbench.sv` code including initial `load_cnt` value to any other value of choice or any other stimulus under initial block, and more.

Homework-3b:

1. `universal_shift_register.sv` partial design template code
2. `universal_shift_register_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 the template code as long as the primary port list is matching within the previous slide. This is to ensure that the testbench is compatible with the design.

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

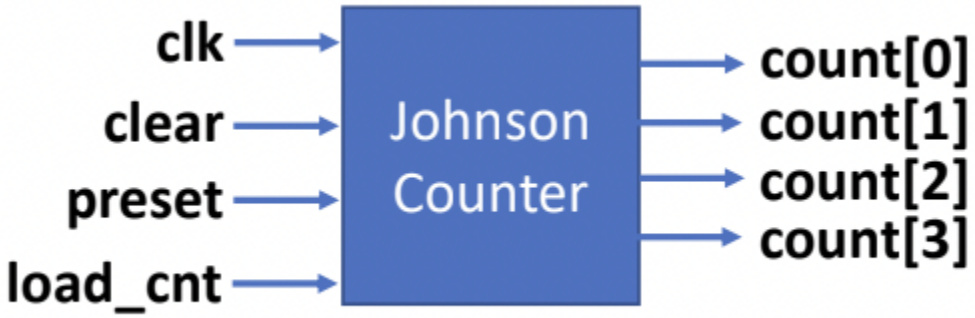
Assignment Tasks

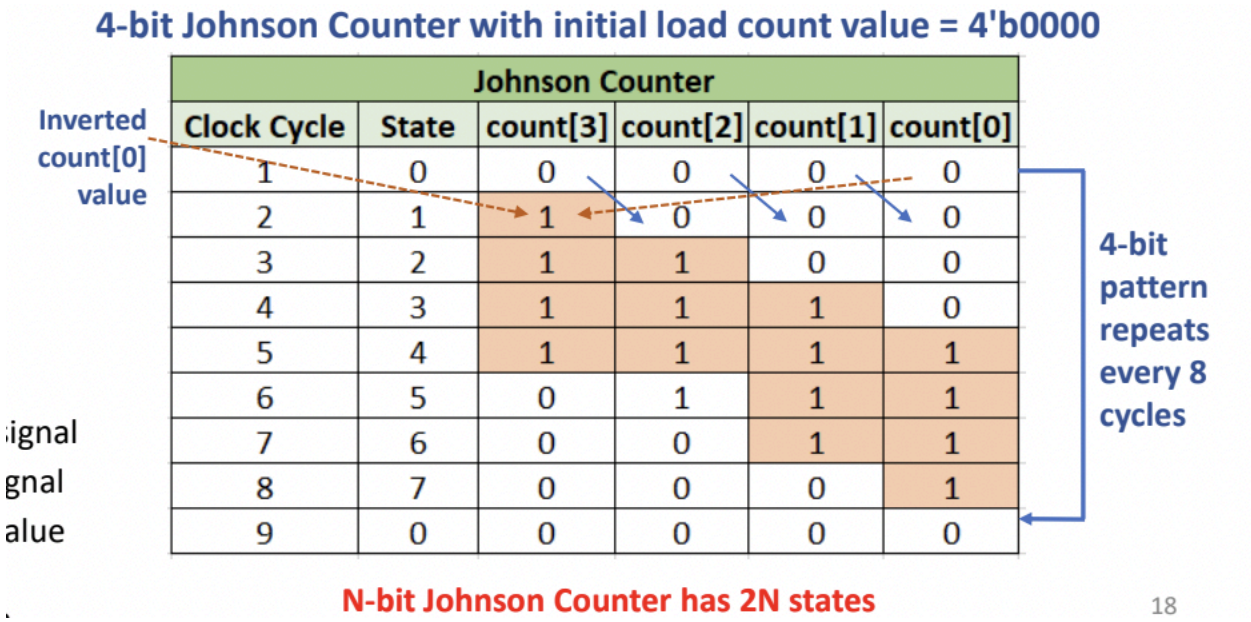
For Homework-3a:

- **Develop Synthesizable SystemVerilog Model for a 4-bit Johnson Counter**
  - Synthesize the Johnson counter.
  - Run simulation using the Johnson counter testbench provided.
  - Review synthesis results (resource usage and RTL netlist/schematic).
  - Review input and output signals in the simulation waveform.
  - Create a SystemVerilog module named as johnson_counter.
  - Use non-blocking assignment statements when implementing johnson_counter.

- **Declare below mentioned Primary Ports for Johnson Counter**
  - Input clk : clock
  - Input clear : asynchronous reset and negedge signal.
  - Input preset : synchronous and active low signal.
  - Input load_cnt[3:0] : Initial count value. Count Value initialized when !preset.
  - Output count[3:0] : output 4-bit count value.

- **Johnson Counter Block Diagram:**





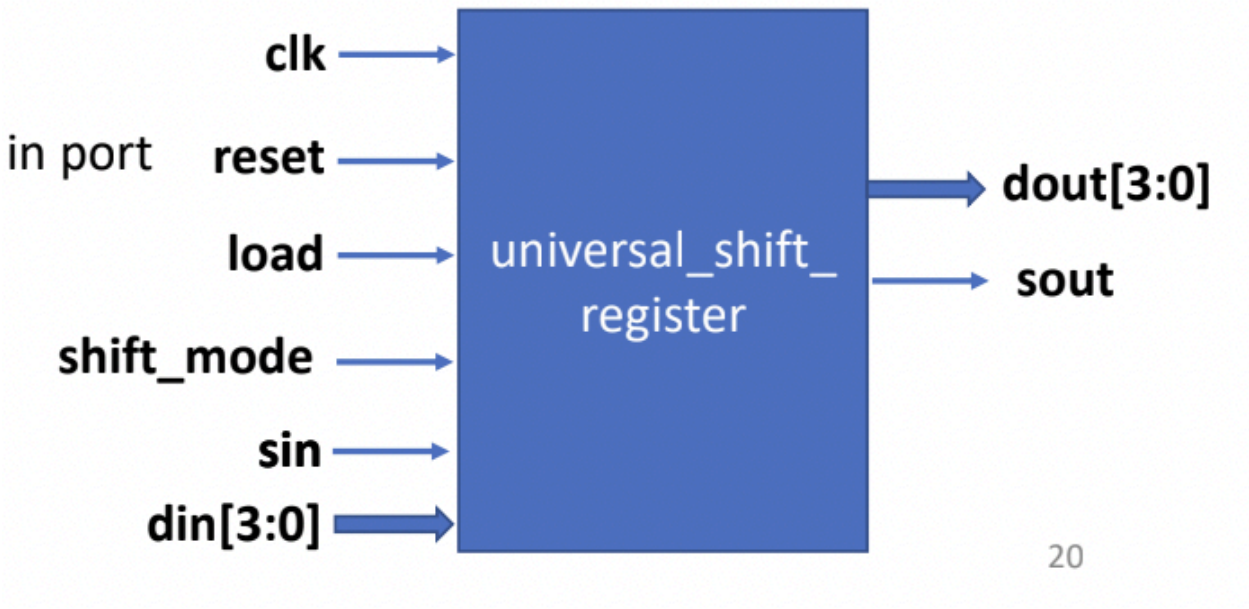
For Homework-3b:

- Develop a synthesizable SystemVerilog model of a 4-bit Universal Shift Register.
  - Universal Shift Register should perform below mentioned 7 operations:
    - PIPO, SIPO-L, SIPO-R, PISO-L, PISO-R, SISO-L, SISO-R
  - Create a module named "universal_shift_register".
  - When implementing shift register for each mode of operation, use non-blocking assignment statements in **case** expression body under clocked always block. See example below:

```
shift_reg[0] <= sin;
shift_reg[1] <= shift_reg[0];
shift_reg[2] <= shift_reg[1];
shift_reg[3] <= shift_reg[2];
```

- Declare below mentioned primary Ports of Universal Shift Register
  - Input clk : clock
  - Input reset : asynchronous reset and posedge signal.
  - Input load : when set to '1' load parallel data through din port into 4-bit shift register.
  - Input din[3:0] : 4-bit parallel data input
  - Output dout[3:0] : 4-bit parallel data output
  - Input sin : 1-bit serial data in
  - Output sout : 1-bit serial data out
  - Input shift_mode[2:0] : selects type of shift operation

- Reference Block Diagram:



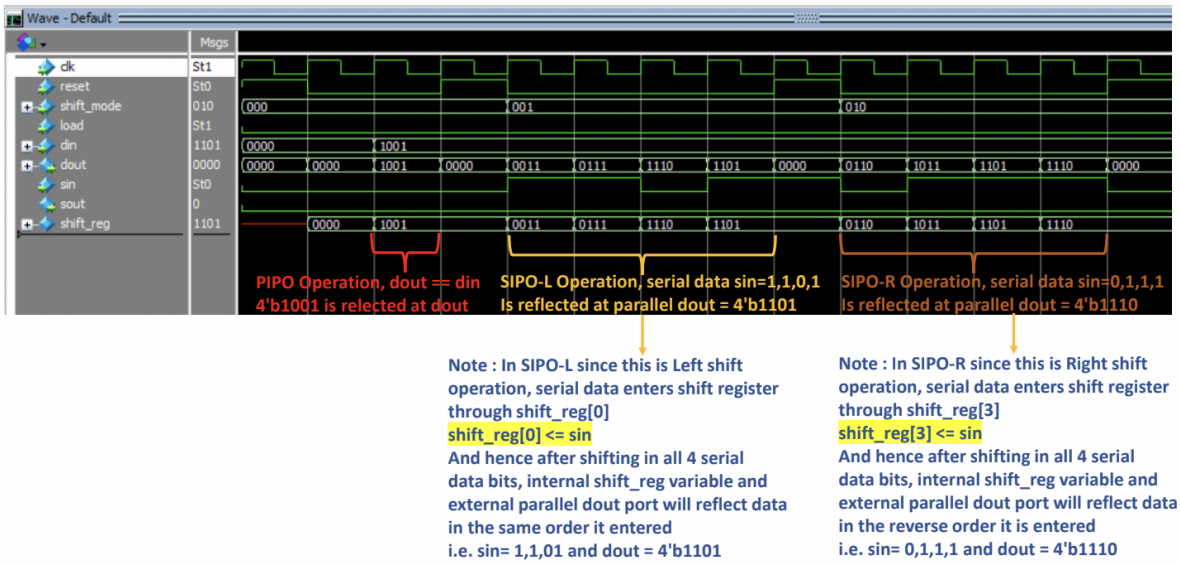
- Implement shift operation based on table below
    - In the table below, the last two columns indicate for each shift operation through which input port data enters in the universal shift register and through which output port data exits the universal shift register.
    - For PIPO, PISO-L and PISO-R, when input load signal is driven to '1' then all bits of the shift register are loaded simultaneously through the din[3:0] input port.

shift_mode	Operation	Value of Input "load"	Input data port	Output data port
3'b000	PIPO (Parallel In Parallel Out)	1	din[3:0]	dout[3:0]
3'b001	SIPO-L (Left Shift Serial In Parallel Out)	0	sin	dout[3:0]
3'b010	SIPO-R (Right Shift Serial In Parallel Out)	0	sin	dout[3:0]
3'b011	PISO-L (Parallel In Left Shift Serial Out)	1	din[3:0]	sout
3'b100	PISO-R (Parallel In Right Shift Serial Out)	1	din[3:0]	sout
3'b101	SISO-L (Left Shift Serial In Left Shift Serial Out)	0	sin	sout
3'b110	SISO-R (Right Shift Serial In Right Shift Serial Out)	0	sin	sout

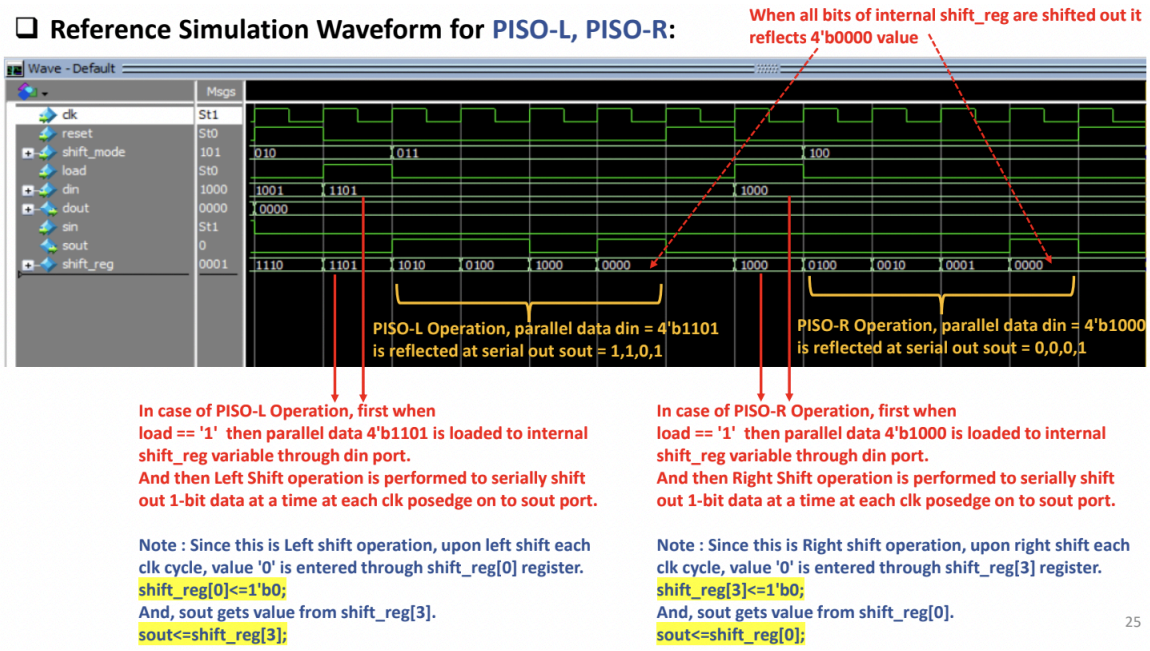
- When implementing each shift operation in SystemVerilog code refer to the table below on which bit position of internal variable shift_reg the data enters and exits.

shift_mode	Operation	Internal Shift Reg assignment	Output dout and sout assignment
3'b000	PIPO (Parallel In Parallel Out)	N/A	dout[3:0] = din[3:0] sout = 0
3'b001	SIPO-L (Left Shift Serial In Parallel Out)	shift_reg[0] = sin	dout[3:0] = shift_reg[3:0] sout = 0
3'b010	SIPO-R (Right Shift Serial In Parallel Out)	shift_reg[3] = sin	dout[3:0] = shift_reg[3:0] sout = 0
3'b011	PISO-L (Parallel In Left Shift Serial Out)	shift_reg[0] = 1'b0	sout = shift_reg[3] dout = 0
3'b100	PISO-R (Parallel In Right Shift Serial Out)	shift_reg[3] = 1'b0	sout = shift_reg[0] dout = 0
3'b101	SISO-L (Left Shift Serial In Left Shift Serial Out)	shift_reg[0] = sin	sout = shift_reg[3] dout = 0
3'b110	SISO-R (Right Shift Serial In Right Shift Serial Out)	shift_reg[3] = sin	sout = shift_reg[0] dout = 0
3'b111	Default Case (No Shifting Operation)	N/A	sout = 0, dout = 0

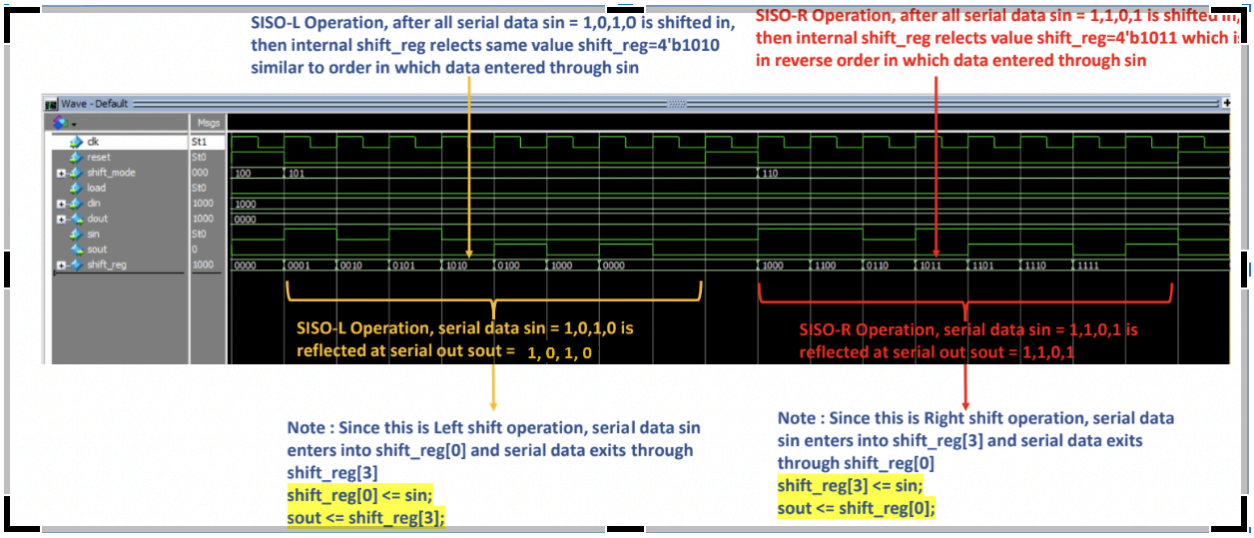
- Reference Simulation Waveform for PIPO, SIPO-L, SIPO-R:



❑ **Reference Simulation Waveform for PISO-L, PISO-R:**



- Reference Simulation Waveform for SISO-L, SISO-R:



Submission Requirements

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

For Homework-3a:

- SystemVerilog design code.
- Synthesis resource usage, schematic generated from RTL netlist viewer and the post-mapping schematics.
- Simulation and transcript snapshots and explain the simulation result to confirm the RTL model developed works as a Johnson counter.

- Explanation of FPGA resource usage in the report is not required.

For Homework-3b:

- SystemVerilog design code
- Synthesis resource usage, schematic generated from RTL netlist viewer and the post-mapping schematics.
- Simulation and transcript snapshots and explain the simulation result to confirm the RTL model developed works as a Universal Shift Register.
- Explanation of FPGA resource usage in the report is not required.