NYU Tandon School of Engineering

Spring 2024, ECE 6443

Midterm Assignment

Instructor: Azeez Bhavnagarwala, email: ajb20@nyu.edu

CA: Charan Chintala, email: cc8473@nyu.edu

Midterm Assignment [released Thursday May 2nd 2024] [due Saturday May 4th 11:59PM]

You are *not allowed to share your solutions* with other colleagues in the class. All uploaded solutions will be checked and graded for originality

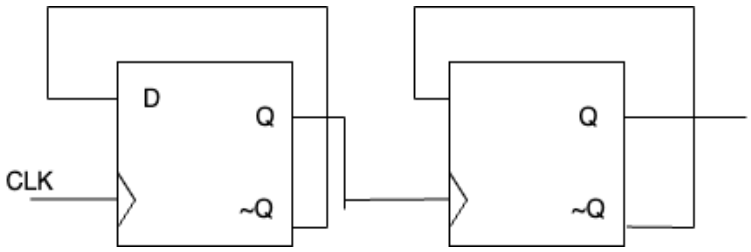
Please use the NYU Brightspace portal to upload your completed Lab.

1. Please provide short answers to the following questions with code:

(i) How does a code infer a latch? Could you provide some RTL code using case and if-else statements that might unintentionally infer latches, and explain how to avoid such inference?

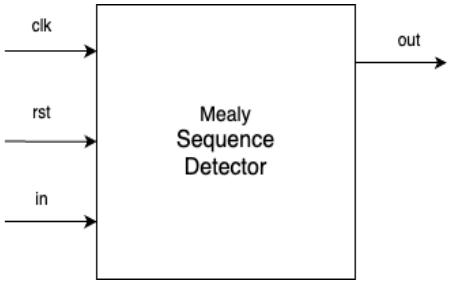
(ii) Write the RTL code to implement a module that counts the number of '1' bits in a given binary input. The module should be parameterized with 'DATA_WIDTH'. Input is 'din' and output to the circuit is 'dout'. Verify the functionality using Testbench, also attach the waveform to the final solution.

2. Write RTL code to implement a frequency divider that divides an input clock signal by 4 using two D flip-flops with asynchronous reset. Also develop a testbench to verify your design, and provide a waveform for justification.



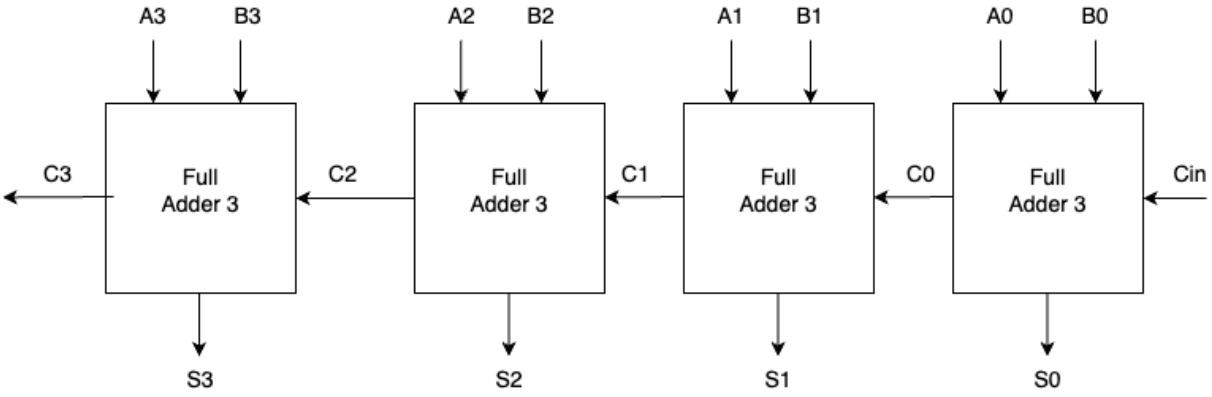
3. Develop RTL code for a sequence detector using a Mealy state machine that can detect overlapping occurrences of the sequence '1010' in a stream of binary input bits. For example, the input stream '101010', the sequence '1010' appears twice. Develop a

testbench to verify your design, and provide a waveform to demonstrate that the machine accurately identifies the pattern ‘10110’. Drawing a state diagram is optional.



4. Write RTL code for a Moore state machine that detects the binary pattern ‘101’ on input. The design should include an asynchronous, active-low reset triggered on the negative edge, and state updates on the rising edge of the clock. Develop a testbench to verify your design, and provide a waveform to demonstrate that the machine accurately identifies the pattern ‘101’, including overlapping patterns. Drawing a state diagram is optional.

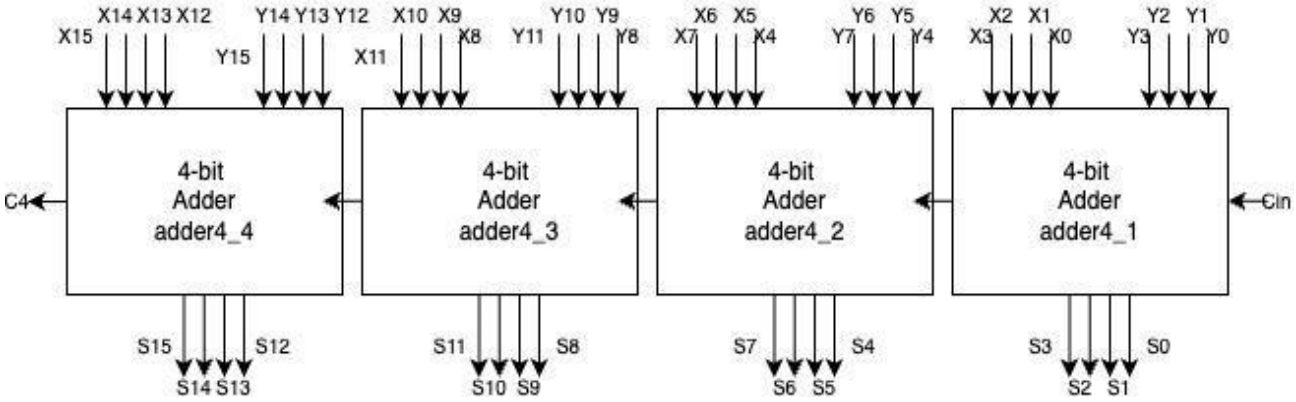
5. Write RTL code for a 4-bit ripple carry adder. The adder should sum two 4-bit operands and output a 4-bit sum along with a carryout. Design a testbench to simulate your ripple carry adder. Provide a waveform to demonstrate the correct operation of adder.



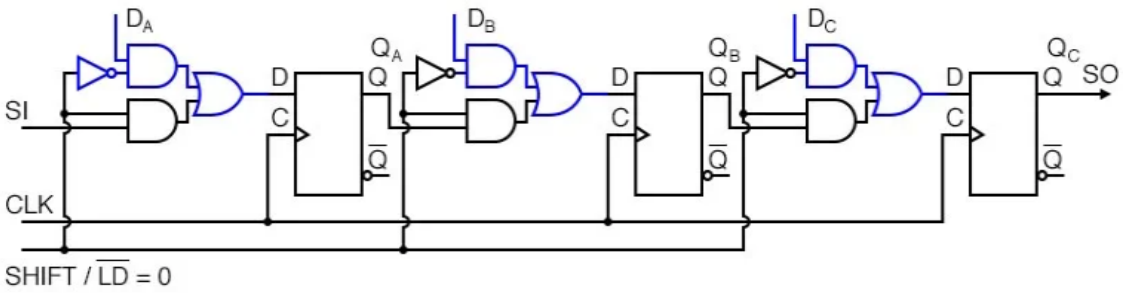
6. Develop RTL code for right rotate on an N-bit vector `'in'` by a number of positions specified by the ‘`rfactor`’. The output should be ‘out’, an N-bit vector that contains the result of rotation. Instead of using concatenate operation, employ a truncate to achieve

rotation effect. Additionally, create a testbench to simulate this module and generate a waveform that demonstrates the functionality of your design across various test cases.

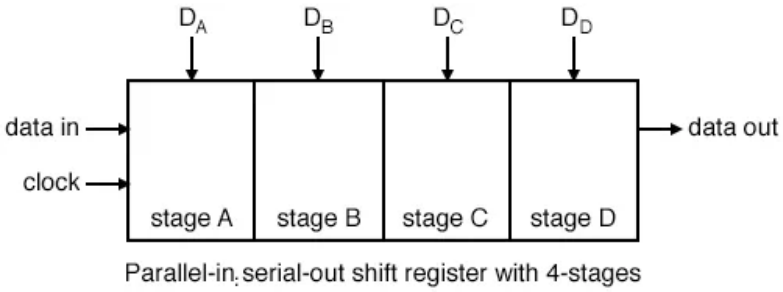
7. Develop RTL code to construct a 16-bit adder by cascading four 4-bit adders modules to perform 16-bit addition operation. Develop a testbench to simulate your 16-bit adder. Provide the waveform output from your simulation to verify correct behavior, showing both the inputs and resulting outputs.



8. Design and Implement a Serial Input Parallel Output (SIPO) register in RTL. Create a testbench to verify your SIPO design. Attach the waveform images that document the functionality of your design.



Parallel-in/ serial-out shift register showing parallel load path



Parallel-in, serial-out shift register with 4-stages

9. RTL code for implementation of adder-subtractor using mode. Where inputs to the circuit are clk, mode, a,b and sum where a, b and sum is 32-bit size. If mode is '0' addition is performed else subtraction is performed. Develop a testbench to verify the functionality of your module. Provide waveforms from your simulation that show the correct operation of the module with changing mode values and different input vectors.

10. Write the RTL code and testbench for the given circuit.