ESE 461: Design Automation for Integrated Circuit Systems

Lab1: Basic Verilog Practice

Due: Sep 12 5:00pm, Submit by Github

Sec-1. Combinational Logic

Problem 1: Implement the logic in Fig.1.

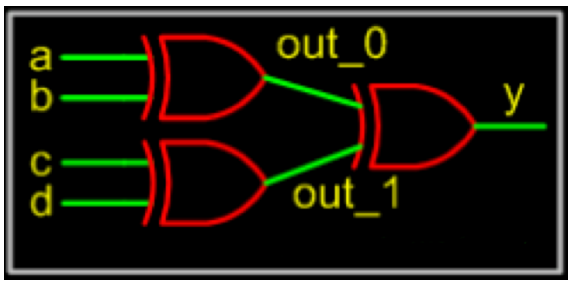

Fig.1

Problem 2: Implement the logic for 1-bit Full Adder in Fig.2.

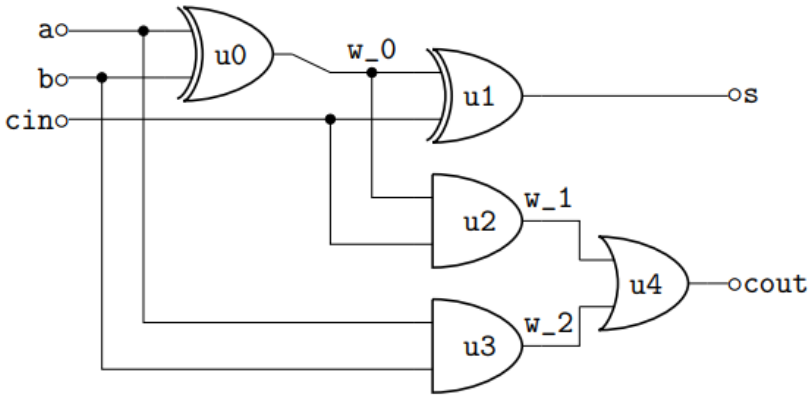

Fig.2

Sec-2. Sequential Logic

Problem 3: Implement the D Flip-Flop shown in Fig.3. The D flip-flop is a widely used type of flip-flop. It is also known as a data or delay flip-flop. The D flip-flop captures the value of the D-input at a definite portion of the clock cycle (the rising or falling edge of the clock). That captured value becomes the Q output. At other times, the output Q does not change. The D flip-flop can be

viewed as a memory cell or a delay line. **In your design, set the active edge in a flip-flop be rising.**

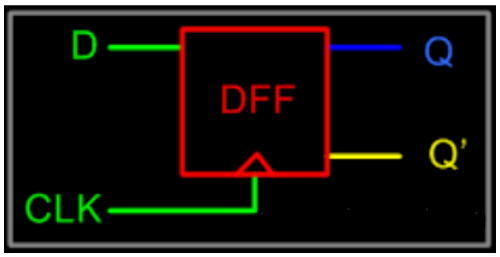

Fig.3

Problem 4: Implement the SR Latch, the circuit using NOR gate

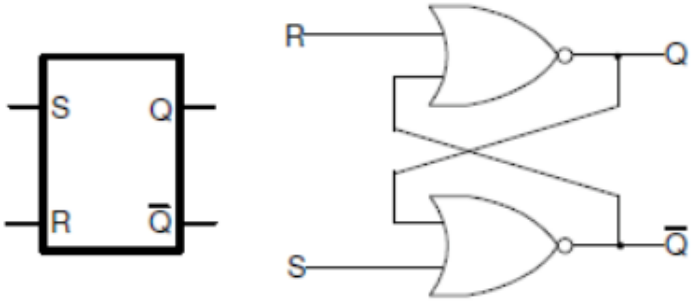

Fig.4

Problem 5: Implement the Finite State Machine – Arbiter in fig.5. The FSM has got the following states:

- IDLE: In this state, FSM waits for the assertion of req_0 or req_1 and drive both gnt_0 and gnt_1 to inactive state. This is the default state of FSM, it is entered after the reset and also during the fault recovery condition.
- GNT0: FSM enters this state when req_0 is asserted and remains here as long as req_0 is asserted. When req_0 is de-asserted, FSM returns to the IDLE state.
- GNT1: FSM enters this state when req_1 is asserted and remains there as long as req_1 is asserted. When req_1 is de-asserted, FSM returns to the IDLE state.

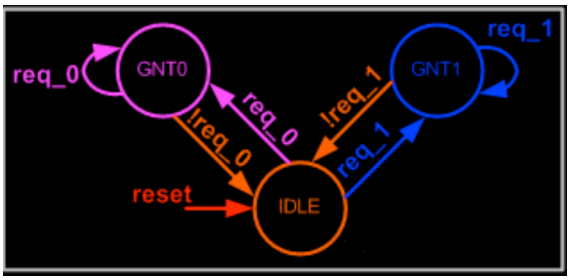

Fig.5

Sec-3 Submission file

Please submit your lab assignment on Github:

1. verilog module code
2. testbench
3. lab1-report.pdf