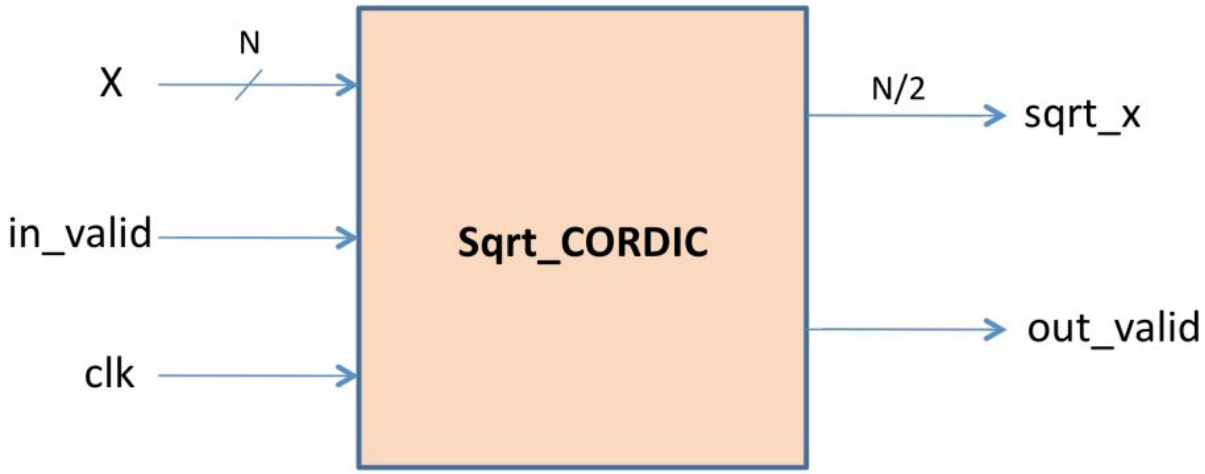
# Midterm Exam ECE 448
# Spring 2016
# Wednesday,
# March 16
# 15 points

Instructions:
Zip all your deliverables into an archive <last_name>.zip and submit it through Blackboard no later than Wednesday, March 16, 10:20 PM EDT.

Develop a VHDL description of the Sqrt_CORDIC circuit, which calculates the integer square root of an input value using a CORDIC approach. The equation for the output sqrt_x is given by:

$$\text{sqrt\_x} = \lfloor \sqrt{x} \rfloor$$

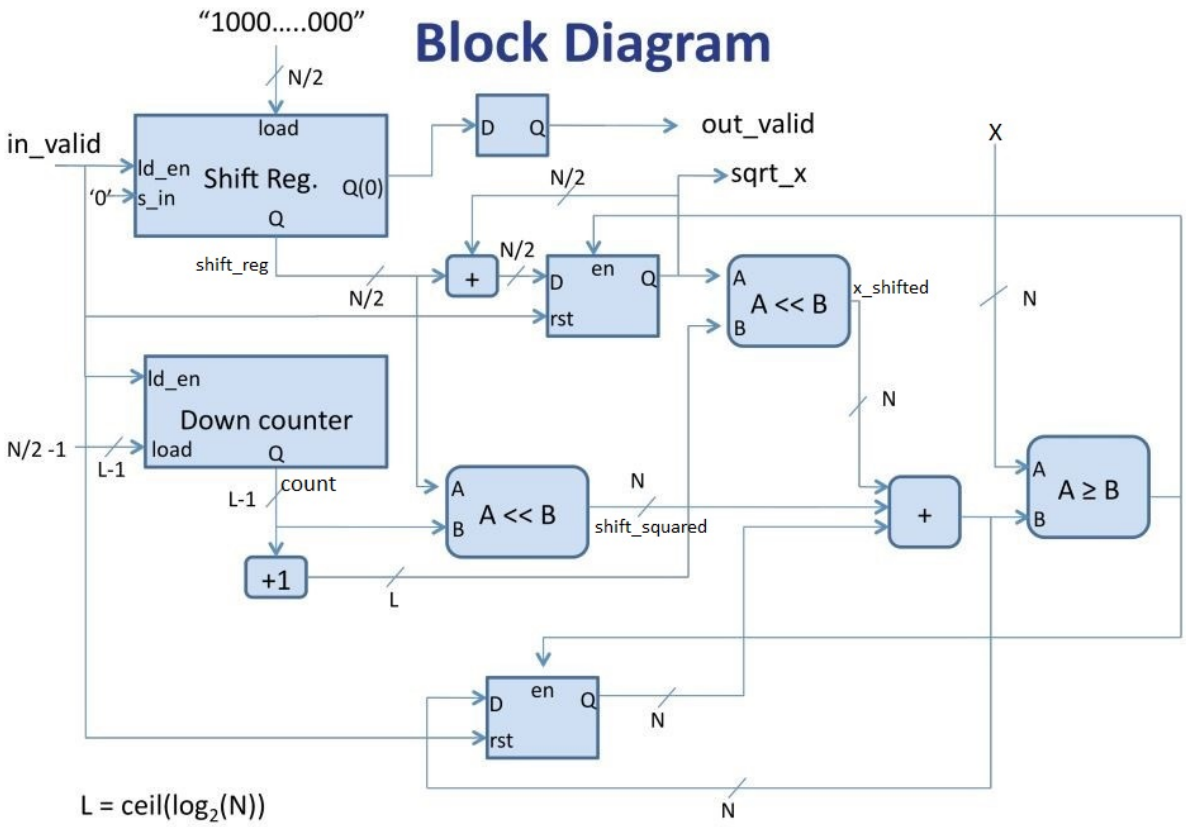
The interface for the square root circuit is shown below:




| Signal Name | Description |
|---|---|
| X | Input value of generic width N. |
| in_valid | Indicates a new input value is available to begin calculation. Held high for one clock cycle. |
| clk | Clock signal |
| sqrt_x | Output signal, valid only when out_valid is high. |
| out_valid | Held high for one clock cycle to indicate the calculation has completed. |

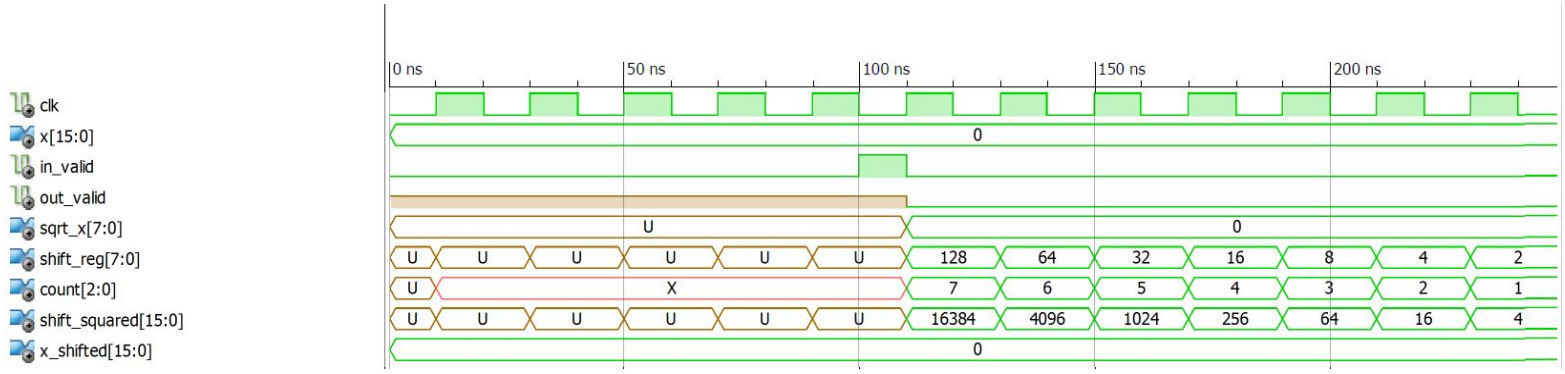
N is a generic, specifying the bus width of the input X. The output width is half of the input width. The input in_valid is held high for one clock cycle to indicate that a new input value is available for calculation. Similarly, out_valid is held high for on clock cycle when a calculation completes. Note that this operation takes multiple clock cycles to complete, and a new calculation cannot begin until the previous one completes.

The block diagram for the design is shown below.

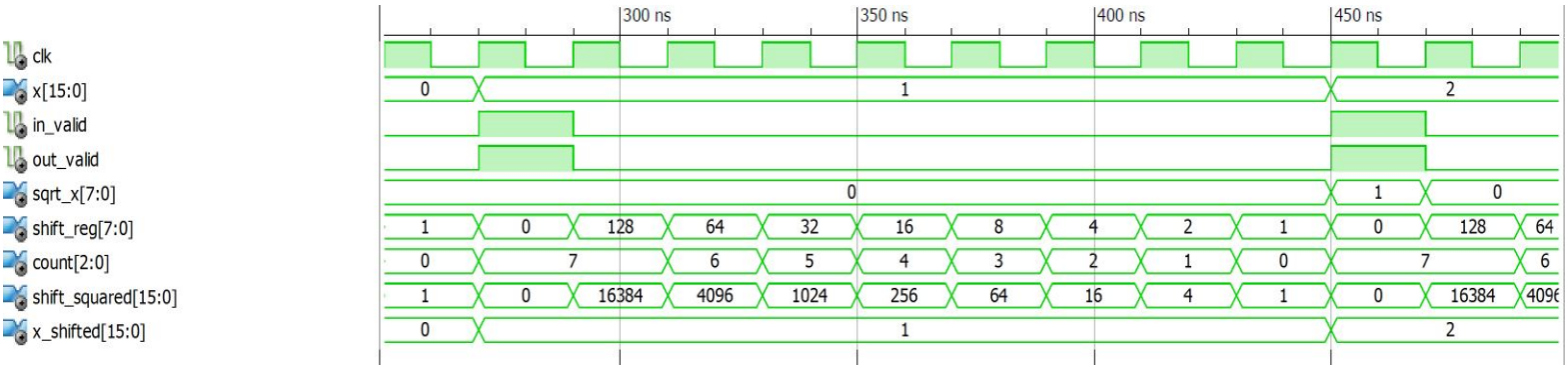



- All synchronous components (with an implicit clk input) are represented by rectangular boxes. All combinational components have rounded edges.
- "A ≥ B" has a value of '1' if A ≥ B, and '0' otherwise.
- All values are unsigned.
- The load value for the shift register, denoted "1000…000," has a '1' in the MSB, and a '0' in all other bits. It is loaded into the register when ld_en = '1'. When ld_en = '0', the shift register shifts right one bit, shifting a '0' into the MSB.
- The down counter loads the value (N/2-1) when ld_en = '1'. When ld_en = '0', the value of the counter decrements by 1.
- The shift register and counter loads and all register resets are synchronous.

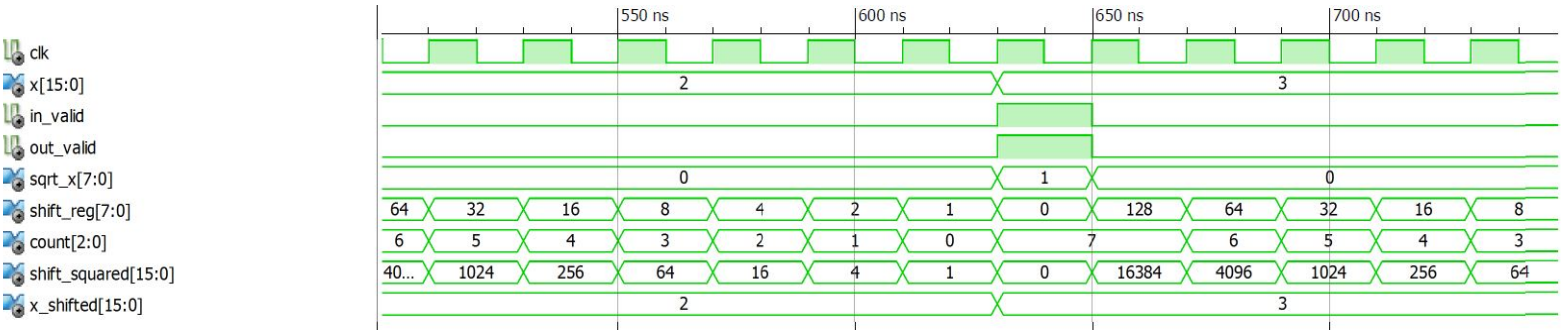
## Timing Waveforms:



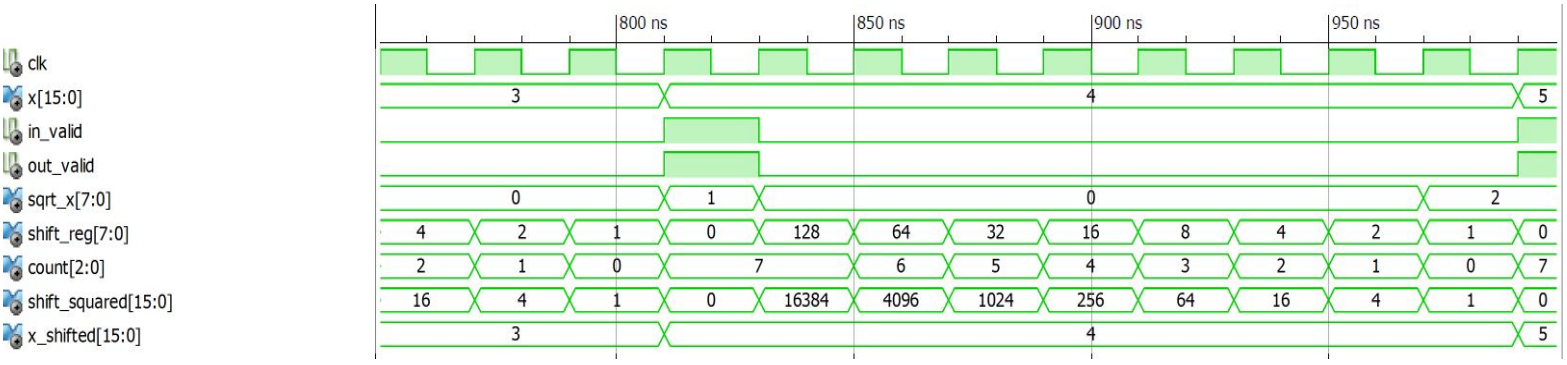

---




---




---



**Tasks:**
Perform the following tasks:

1. Write a synthesizable VHDL code representing the square root circuit
2. Write a separate testbench to test the variable rotator (shown below) and print timing waveform for this part separately.


3. Write a testbench verifying the operation of your entire circuit.
4. Perform functional simulation of your circuit and use it to debug your VHDL code. Take a print out of the waveform showing the entire operation using default PDF conversion tool installed in the lab (Use multiple page option in order to display necessary information on multiple pages, if required).
5. Synthesize your circuit.
6. Implement your circuit using
7. FPGA family: Artix 7
8. Device: x c 7 A l 0 0 t -1 csg324AB
9. Speed Grade: -1
10. Run the static timing analysis of your circuit.
11. Based on the circuit block diagram and the report from the static timing analysis, determine the most critical path in your circuit and the circuit maximum clock frequency.
12. Based on the implementation reports, determine the number of CLB slices, LUTs, flip-flops, and pins used by the circuit.
13. Perform the timing simulation of your circuit at the maximum clock frequency returned by the static timing analysis. Take a printout of the waveform showing the entire operation using default PDF conversion tool installed in the lab (Use multiple page option in order to display necessary information on multiple pages, if required).

**Deliverables:**

1. VHDL code of your entire circuit fulfilling the requirements specified in the *Design Requirements* section above.
2. VHDL code of your testbenches for the variable rotator and the entire circuit.
3. Timing waveforms from the functional and timing simulations demonstrating the correct operation of the variable rotator and the entire circuit.
4. Description of the critical path in the entire circuit.
5. FPGA resource utilization for the entire circuit (as defined in Task 8 above).
6. Minimum clock period and maximum clock frequency of the entire circuit after placing and routing.