ECE 526L Lab6

In this Lab you will create a reloadable 8-bit up counter. Also, you will create a reset synchronizer (AASD type) for the reset signal.
1. For the counter use the following ports:

- Single bit inputs: Clock, Reset, Enable, Load
- Eight bit Data input
- Eight bit Count output

2. Model the reset as an asynchronous, active low input.

3. Model the enable as a synchronous, active high input. When asserted, the count is incremented or loaded with new data (if LOAD is high). If ENABLE is not asserted, the counter will hold its value. The counter does not load new data if it is not enabled.

4. Model LOAD as a synchronous, active high input. When asserted, the value on the data pins is loaded into the counter after the positive edge of the clock if the counter is enabled.

5. When LOAD is low, ENABLE is high and RST is high, the counter advances on the positive edge of the clock.

6. Create a module for the asynchronous assert, synchronous de-assert function of reset synchronizer as described in the figure below.

7. Instantiate the two design modules in a top-level design unit. The AASD circuit should be at the same level as the counter, not embedded in it.

8. Create a test bench that does the following:
a. Instantiates your top level design.

b. Provides a clock generator with a 20 ns period.
c. Demonstrates an asynchronous reset.
d. Show the counter initiates incrementing after reset is released. After it reaches a count of 8, parallel load 240 decimal
e. Demonstrate that it will count from 240 until the counter rolls over (i.e. returns to zero) and then starts counting back up.
f. Demonstrates that reset overrides both load and increment.
g. Demonstrates the correct functioning of enable.

9. Extra Credit**:**
a) If the reset was synchronous, how would the circuit behave differently?
b) How would you change the code to have a defined max counter 245 and then counter rolls over (i.e. returns to zero) and then starts counting back up.
Answer these questions in your lab report.