Homework 4B

Deliverables due by Saturday, October 26, 11:59 PM

Download the code for the Sorting example, available at
https://people-ece.vse.gmu.edu/coursewebpages/ECE/ECE545/F24/homework/Sorting.zip.
This example is described in detail in slides for
Lecture 4: RTL Design Methodology - Part A.

Perform the following tasks for this example:

1. Develop and debug a universal testbench that supports at least the following three simulation types available in Vivado: Behavioral, Post-Synthesis Timing, and Post-Implementation Timing, and is described by the following specification:
   a. The testbench should support the following three cases:
      i. N=16 and w=8
      ii. N=16 and w=16
      iii. N=32 and w=16
      The choice among these three cases can be made by commenting and uncommenting particular lines of code. Using any other mechanism supported by VHDL is allowed as well. Please note that the code of the testbench may also need to be slightly changed depending on the simulation type.
   b. Inputs and expected outputs for each case should be stored in separate text files. These files should have the format
      <address_in_hex> <data_in_hex>
      Each pair should be stored in a separate line. Lines starting with # should be treated as comments. All values should be expressed in hexadecimal notation.
   c. Inputs should be read from the respective file and applied to the ports RAdd and DataIn at the falling edges of the clock.
   d. Expected outputs should be read from the respective file and applied to the internal signal called DataOut_expected $1/8^{th}$ clock cycle after the rising edge of the clock.
   e. Actual outputs should be compared with expected outputs $1/4^{th}$ clock cycle before the next rising edge of the clock. Incorrect outputs and the simulation time when they appeared should be written to standard output in the format:
      Time: <value_in_decimal> ns, Actual output: <value_in_hex>, Expected output: <value_in_hex>
   f. Clock period should be set to 10 ns for behavioral simulation and at least 4/3 * minimum clock period for post-synthesis and post-implementation timing simulations.

2. Prepare a constraint file setting the target clock period to 10 ns and add it to your design. Modify this constraint in case of obtaining negative value of the worst negative slack (WNS) after synthesis or implementation.

For each of the three cases, i, ii, and iii:

3. Perform behavioral simulation. Include in the simulation waveforms all input/output ports and the most important internal signals named in the block diagram of the Datapath. Take, store, and name screenshots clearly demonstrating the correct operation of your testbench and design under test. Measure the execution time required for sorting in clock cycles and time units.

4. Synthesize your design and then analyze the outputs from the synthesis as follows:
   a. Modify the testbench if needed and perform post-synthesis timing simulation.

Revise the testbench until you obtain the correct results.
Take, store, and name screenshots clearly demonstrating the correct operation of your testbench.

b. Determine the worst negative slack (WNS) and the corresponding minimum clock period and maximum clock frequency for the requested clock period of 10 ns (or longer if needed).
c. Investigate and store the obtained schematic.

5. Implement your design and then analyze the outputs from the implementation as follows:
   a. Modify the testbench if needed and perform post-implementation timing simulation. Revise the testbench until you obtain the correct results.
   Take, store, and name screenshots clearly demonstrating the correct operation of your testbench.
   b. Determine the worst negative slack (WNS) and the corresponding minimum clock period and maximum clock frequency for the requested clock period of 10 ns (or longer if needed).
   c. Determine resource utilization separately for the Datapath and Controller. Copy to your report the following values:
      a) LUTs (a.k.a. Slice LUTs)
      b) FFs (flip-flops, a.k.a. Slice Registers)
      c) BRAMs (a.k.a. Block RAM Tiles)
      d) I/O pins (a.k.a. Bonded IOB).
   d. By comparing the block diagram of the Datapath from the slides and the schematic of the Datapath from Vivado, try to determine how many LUTs and FFs are necessary to implement each logic component that appears in the block diagram.

Deliverables:

1. Testbench and three testvector files.
2. Constraint file(s).
3. Screenshots clearly demonstrating the correct operation of:
   a. Behavioral simulation
   b. Post-synthesis timing simulation
   c. Post-implementation timing simulation

   for all three test cases.
   For at least two cases, demonstrate, by manipulating testvector files and using screenshots of the standard output window, that your testbench correctly reports discrepancies between expected outputs and actual outputs.
4. A report containing for each case, i, ii, and iii:
   a. post-synthesis worst negative slack (WNS) and the corresponding minimum clock period and maximum clock frequency for the requested clock period of 10 ns.
   b. post-implementation worst negative slack (WNS) and the corresponding minimum clock period and maximum clock frequency for the requested clock period of 10 ns.
   c. time required for sorting (in clock cycles and ns) and delay between the rising edge of the clock and DataOut (in ns).
   how do these times and delay depend on values of N and w?
   d. post-implementation resource utilization in terms of LUTs, FFs, BRAMs, and IOBs.

   Additionally, explain
   e. how the numbers of LUTs, FFs, BRAMs, and IOBs depend on values of N and w? why?
   f. how does the minimum clock period after synthesis and after implementation depend on parameters N and w? why?
   g. major differences between behavioral and post-implementation timing simulation.