UNIVERSITY OF TEHRAN
Electrical and Computer Engineering Department
Digital Logic Design, ECE 367, Fall 1398
Computer Assignment 6
Accelerator RTL design with a simple wrapper
Week 15

Name:
Date:

You are to design an accelerator for calculation of *sin(x)*, where *x* is between 0 and π. Calculation is to be done by Taylor series expansion of *sin(x)*. The input *x* is a 16 bit fixed point number with 8 fractional and 8 integer bits. The output is also a 16 bit fixed point number, the integer part of which is always 0. In addition to the *x* input, the circuit has an 8-bit fixed point *y* input that defines the precision of the calculation of *sin(x)*. The iteration of Taylor series calculations stops if a term being added is less that *y*. A) Complete the RTL design of this accelerator. B) Develop a wrapper that connects this circuit to an 8-bit shared bus for getting the inputs *x* and *y*, and the output, *z*.

Design Phase:

1. Show the schematic diagram of the datapath of this circuit
2. Show the controller state diagram
3. Implement the datapath (on paper) using components discussed in class.
4. Write Controller SystemVerilog description and show its Hoffman model.
5. Show how the controller Hoffman model connects to the datapath and outside busses.
6. Show the schematic diagram of the datapath of the wrapper circuit
7. Show the wrapper controller state diagram
8. Implement the wrapper datapath (on paper) using components discussed in class
9. Write the wrapper controller Verilog description and show its Hoffman model
10. Show how the wrapper controller and datapath are connected
11. Show the wiring between the wrapper and *sin(x)* circuit

Implementation Phase:

1. Build the *sin(x)* computation unit in Quartus II using predefined Altera components, Verilog modules, and discrete parts
2. Enter the Verilog description of the controller of *sin(x)* is Quartus II
3. Generate the complete design of *sin(x)*
4. Generate the complete design of the wrapper you designed for *sin(x)*
5. Wire *sin(x)* computation with its wrapper in Quartus II
6. Synthesize the complete circuit and generate its .vo and .sdo files
7. In a testbench, test your complete circuit