UNIVERSITY OF TEHRAN

Electrical and Computer Engineering Department

Digital Logic Design ECE 367 / Digital System I ECE 894

Fall 1399

Computer Assignment 6

RTL Complete Component Design - Week 16

Name:	**Date:**
Username:	

You are to design an accelerator for calculation of *cos(x)*, where *x* is between 0 and π/2. Calculation is to be done by Taylor series expansion of *cos(x)*. The input *x* is a 10-bit fixed point number with 8 fractional and 2 integer bits. The output is also a 10-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 *cos(x)*. The iteration of Taylor series calculations stops if a term being added is less that *y*. Complete the RTL design of this accelerator. Perform simulation and synthesis of this circuit. This circuit has the standard *start* and *ready* handshaking signals.

$$\cos x = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n)!} x^{2n} = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \cdots \quad \text{for all } x$$

Design and Verification 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. Use a lookup-table ROM for coefficients.
4. Write Controller SystemVerilog description and show its Hoffman model. Simulate and verify the stand-alone operation of the controller.
5. Show how the controller Hoffman model connects to the datapath and outside busses.

Implementation Phase:

1. Build the *cos(x)* computation unit (datapath) in Quartus II using predefined Altera components, Verilog modules, and/or discrete parts. The datapath must consists of separate datapath components, i.e., do not use a single SystemVerilog for the entire datapath. Use a memory hex file for initializing your ROM.
2. Enter the SystemVerilog description of the controller of *cos(x)* in Quartus II.
3. Generate the complete design of *cos(x)*.
4. Synthesize the complete circuit and generate its .vo and .sdo files.
5. In a testbench, test your complete circuit.