UNIVERSITY OF TEHRAN
Electrical and Computer Engineering Department
Digital Logic Design, ECE 367 / Digital Systems I, ECE 894
Fall 1400 - Computer Assignment 6
RTL Accelerator Design
Week 24

Name:
Date:

Taylor series is one of well-known methods to compute mathematical functions such as *sin(x)*, *cos(x)*, *exp*, etc. In this problem you are to design an accelerator for calculation of a trigonometric function. This is a sequential circuit that computes an approximation of *tan x* using the first 8 terms of its Taylor expansion. The numbers B_{2n} are Bernoulli numbers. The algorithm you choose to implement should use an array multiplier and a ROM for series coefficients.

$$\tan x = \sum_{n=1}^{\infty} \frac{B_{2n}(-4)^n\,(1-4^n)}{(2n)!}\,x^{2n-1} \qquad = x + \frac{x^3}{3} + \frac{2x^5}{15} + \cdots \qquad \text{for } |x| < \frac{\pi}{2}$$

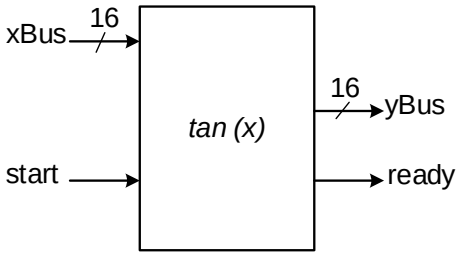
$$\tan x = x + \tfrac{1}{3}x^3 + \tfrac{2}{15}x^5 + \tfrac{17}{315}x^7 + \tfrac{62}{2835}x^9 + \ldots$$

Shown below is the block diagram of the module that approximates *tan()*. The module accepts a 16-bit fixed point value on *xBus* between 0 and π/2 after *start* is asserted. After that, computation is started. During the computation, the *busy* signal is asserted. After the completion of the computation, the result becomes available on output *yBus*, *busy* is deactivated, and *ready* is issued.

As mentioned, the *x* input is between 0 and π/2, and all numbers are represented in 16-bit fixed point format. In addition, a 16-bit fixed point adder and a 16-bit fixed point array multiplier are available as datapath components. The series coefficients must be pre-calculated and stored in a combinational lookup table. The table has four address lines and a 16-bit data output.

a. Generate a ROM in Quartus for the required memory of factorial calculations.
b. Design the datapath of module *tan(x)* and implement it in Quartus. For the Rom, you need to use the ROM you created in Quartus in Part a, for the multiplier you can use the multiplication operator and leave it for Quartus to generate the multiplier for you, for rest of the datapath you can use SystemVerilog descriptions and/or *Lpm* packages from the Quartus library.
c. Write SystemVerilog description for the controller of module *tan(x)*. Make this a separate module and after verifying its operation, make a symbol for it to be used in the top-level design of *tan(x)* accelerator.

d. Put the DP and CU of this trigonometric accelerator together to build a Quartus model of the complete system. Synthesize this circuit and generate post-synthesis timing and netlist files.
e. Generate a testbench for testing your accelerator of Part d.
f. Report your findings, check the timing, and see the floorplan on the FPGA.



Deliverables:

Generate a report that includes all the items below:

A. Prior coming to the lab, for all the above problems hand-drawn schematic diagrams and partial timing diagrams are required. Your SystemVerilog descriptions must correspond to the circuit diagrams. Your simulation run and the project built for this purpose must be demonstrated to the TA.
B. Document your Quartus projects of Parts b (includes a), c, and d. Make sure you understand the synthesis outputs and their corresponding timings. Document images and reports of synthesis in your report.
C. For all three parts, you should look at the FPGA layouts, device view and RTL view. Be able to explain the details of various views. In the layout be able to identify FPGA cells that use a memory element versus those that are purely combinational.
D. For Part e, your testbench must test the accelerator for timing and for functionality of the corner cases. Include simulation run images in your report. Annotate the images with timing information.

Make a PDF file of your report and name it with the format shown below:

FirstinitialLastnameStudentnumber-CAnn-ECEmmm

Where *nn* is a two-digit number for the Computer Assignment, *mmm* is the three-digit course number under which you are registered, and hopefully you know the rest. For the *Firstinitial* use only one character. For *Lastname* and for the multi-part last names use the part you are most identified with. Use the last five digits of your student id (exclude 8101) for the *Studentnumber* field of the report file name.