UNIVERSITY OF TEHRAN

Electrical and Computer Engineering Department

Digital Logic Design, ECE 367, Digital System I, 894, Fall 1403

Computer Assignment 5&6

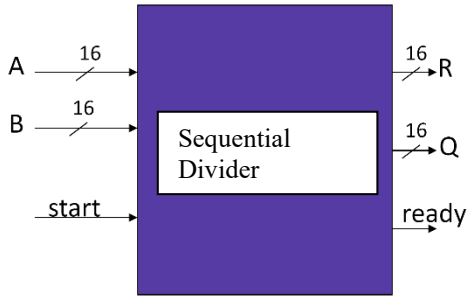
RTL Accelerator Design

Name:	Username:
Date:	Student Number:

In this assignment, you are to design a Restoring divider accelerator and interface it with an 8-bit bus for burst mode operation.

Part 1: Divider Module Design

You are to design a Sequential Divider accelerator that uses the Restoring divide algorithm. The divider takes two 16-bit inputs, Dividend (*A*) and Divisor (*B*). After receiving the *start* positive pulse, the module captures the inputs at the rising edge of the clock. This input capturing should occur in a single clock cycle. After performing the division algorithm on the received inputs, the module gives two 16-bit outputs that are Remainder (*R*) and Quotient (*Q*) and issues a *ready* signal. This signal remains active until the next division operation begins. Note that the inputs are being captured in burst mode.

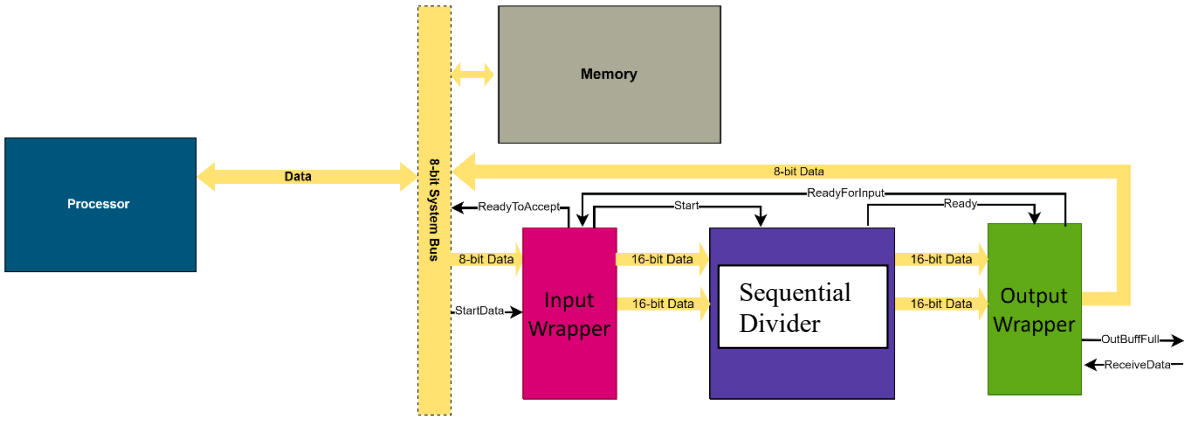


a) Show the complete Data path and controller of your design.
b) Write the Verilog description of the module.

c) Simulate the module with different test cases such as Positive dividend and divisor, Negative dividend and divisor, Zero divisor (ensure proper error handling),…

Part 2: 8-Bit Bus Interfacing

Your Divider will function as an accelerator in a system with an 8-bit data bus as shown below:



a) Design an input wrapper to accept burst inputs from an 8-bit bus for the dividend and divisor. The wrapper issues a *ReadyToAccept* signal to indicate that it is ready to receive data. After receiving a *StartData* positive pulse, the inputs are buffered internally. Once the *ReadyForInput* signal is asserted, the 16-bit data is provided to the Divider module as its operands, and the *start* signal is asserted for the Divider.

b) Design an output wrapper to collect 16-bit results (Quotient and Remainder) from the Divider module and send them in burst mode over an 8-bit bus. The output wrapper begins data transfer after receiving the *Ready* signal from the Divider. It then asserts the *OutBuffFull* signal to indicate that data is ready for transfer. Upon receiving the *ReceivedData* signal from the bus controller, it starts the data transfer process. Once the data transfer is completed, the wrapper issues the *ReadyForInput* signal and prepares to receive the next set of data.

c) Write the Verilog description of the input, output wrappers and connect them to your divider and test the whole design. Show your results.

Deliverables:

Generate a report that includes all the items below:

a) For Part 1 show your Data path and Controller diagram on paper.
b) For other parts, show simulation results. Your simulation run and the project built for this purpose must be demonstrated to the TA.
c) For all parts, project files, and results must be demonstrated to the TA. Using waveforms, circuit diagrams, and other circuit representations justify your answers.

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.