UNIVERSITY OF TEHRAN
Electrical and Computer Engineering Department
Digital Logic Design, ECE 367, Fall 1397
Computer Assignment 5
Basic RTL Design, Using Counters, Shifters, Implementing with Quartus
Week 12

Name:

Date:

In this assignment, you will design a frequency multiplier and implement it in Quartus II.

A frequency multiplier takes an input incoming signal frequency (*inSignal* with frequency *inFreq*) and multiplies it by a value determined by multiplication factor (*multFactor*). The output is a signal (*outSignal*) with a frequency equal to *outFreq* = *inFreq*multFactor*. Multiplication factor is a power of two and could be shown as 2^n, where n is between 1 and 5. Therefore, the circuit takes $log_2^{MultFactor}$ as the input value. For example, if *multFactor* is 8, the corresponding input value would be 3. *inFreq* frequency range is between 1 KHz and 1 MHz. A reference clock input is used for the reference working clock of this circuit and is provided via an input named *refClk*, with a high frequency of 50 MHZ as provided on an FPGA development board. The output of the circuit that carries the faster signal is *outSignal* with a frequency of *outFreg*. The frequency of this signal is $inFreq\times 2^n$. The circuit has a *valid* output that is asserted when frequency multiplication is taking place and is appearing on *outSignal*. When a positive pulse appears on *adjust*, the circuit is informed of a new multiplication factor, and begins preparation for the multiplication process with this new factor. While this preparation is taking place, *valid* becomes 0 and remains 0 until preparation is complete, at which time the generated *outSignal* signal frequency represents the new multiplication factor.

Hint: $outFreq=inFreq\times 2^n = 50\div k$; $k=50\div(inFreq\times 2^n)$; $m=50\div inFreq$; $k=m\div 2^n$

Design Phase:

A) Show the design of a circuit that continuously divides the *refClk* by *inSignal* and calculates the numeric *m* value. This circuit can then be put in a mode to calculate *k* from *m*. This requires shifting *m* by 2^n, (i.e., shift *m* *n* places). Provide control signals for calculation mode and divide mode. We refer to this circuit as *kCalculator*.

B) The next part of the datapath requires a circuit that divides the reference clock (*refClk*) by *k*. We refer to this part as the *continuous50Divider*.

C) Show the complete datapath, including the *kCalculator* and *continuous50Divider*.

D) Generate a controller to start the *kCalculator*, and when it is done, perform the $50\div k$ operation in a continuous way using *continuous50Divider*. This will result in the continuous *outSignal* output signal.

E) Draw a state diagram that shows the behavior of your controller and issuance of control signals.
F) Show wiring between the datapath and the controller.

Implementation Phase:

A) Build the datapath in Quartus II using predefined Altera components, Verilog modules, or discrete parts.
B) Describe the circuit controller in Verilog.
C) Generate a symbol for the controller from its Verilog description.
D) Generate the complete frequency multiplier design by instantiating its datapath and controller.
E) Synthesize the frequency multiplier circuit and generate its .vo and .sdo files.
F) In a testbench, test your complete circuit.