BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI

I Semester 2015-2016

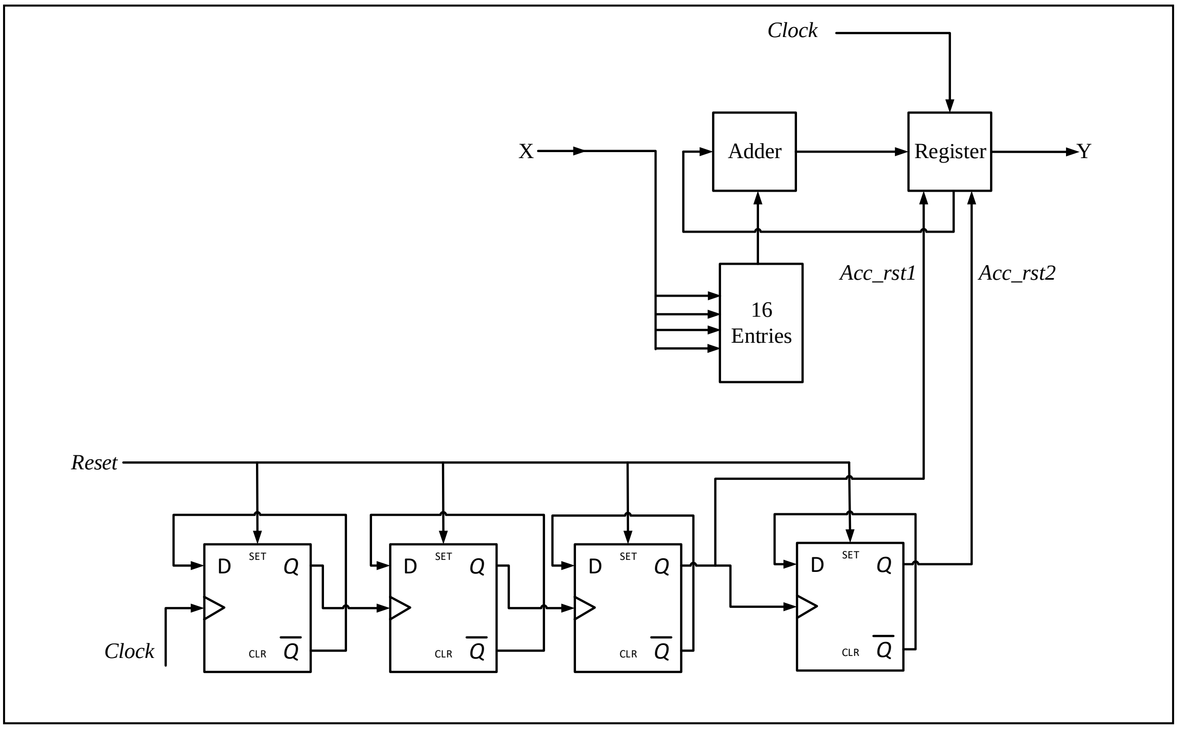
CS F342 Computer Architecture

Lab Exam

Date: 16/11/2015 TIME: 90 Minutes MM: 33

INSTRUCTIONS: (i) Create a folder, rename it to your IDNO. (ii) Rename the project with your IDNO (iii) Write your IDNO and name in .v file. *Write all modules in a single .v file*

Implement the following arithmetic circuit as shown in the figure and described below.



During a clock cycle the circuit takes as input a four bit value X and produces the product X*25. This product is added to a value already stored in a register (initially zero is stored in the register) and the result is put back in the register. This process is repeated for 4 clock cycles with four possible new values of X in each cycle. The value at the end of four cycles is retained in the register for the next four cycles. At the beginning of next cycle (ninth) the register value is reset to zero and the whole process repeats. Besides X, Clock and Reset are also input to the circuit. The output of the circuit is Y and is of 13 bits.

Implement the above circuit in following modules:

1. **16_Entries**: This is 16:1 Multiplexer having the select lines as the 4 bit number X. The 16 inputs are the pre-computed product values of 25 with 0 to 15. The multiplexer has to select the pre-computed product value according to the 4 bit input X. For example, if X = 0010, then the output of 16:1 Multiplexer is 25 multiplied by 2. The output of the multiplexer is of 9 bits. Implement the 16:1 multiplexer using behavioral or dataflow modelling.
2. **Adder_Register:** The adder and register block together is called Accumulator. The purpose of this block is to add the output of the 16:1 multiplexer in each clock cycle with previously accumulated value in register. The register has two resets *Acc_rst1* and *Acc_rst2.* Both the *signals* are generated from the circuit shown in figure. *Acc_rst1* is a clock divide by 8 and *Acc_rst2* is clock divide by 16 signal. At the Positive edge and Negative edge of the clock divide by 16 signal *(Acc_rst2)*, the Accumulator is reset to value ‘0’. When the clock divide by 8 signal *(Acc_rst1)* is high (it will be high for 4 clock cycles), the accumulator adds the multiplexer output with previously stored value from the register and stores it back in the

register for 4 *clock* cycles. After 4 *clock* cycles, the value in register remains same for the next four clock cycles. Implement this Adder_Register block using Verilog behavioral description. The size of the Accumulator is of 13 bits.

3. **ACC_RST:** This module takes as input the clock and Reset and produces the Acc_rst1 and Acc_rst2 signals as output. Describe the D-FFs used in this module using behavioral modeling.
4. **INTG:** This is the integrator module which integrates the above three modules into a working circuit.
5. Write the test bench for the following inputs
   1) Show the output for the first four clock cycles input as 10, 5, 12, 1
   2) Show the output for the next four clock cycles input as 13, 7, 9, 2
   3) Show the output for the next four clock cycles input as 11, 5, 4, 2