# Digital System Design

**Submission Date: 23rd October, 2017**

Q: 1 Write RTL Verilog code of the module device under test given in $Figure\ 2.22$. Four 8 bit inputs, in1, in2, in3 and in4, are input to four 8 bit registers, R0, R1, R2 and R3, respectively, at every positive edge of the clock. Four values in these registers are added, and bitwise AND operation is performed on the values stored in these registers to produce out 1 and out 2.

Q: 2 Design a datapath with three 8 bit accumulators. The first accumulator, acc1, adds a 4 bit input data in acc1 in every clock cycle. The second accumulator, acc2, adds the first accumulator in itself, and the third accumulator, acc3, adds the first and second accumulators in itself in every clock cycle. Each accumulator has an asynchronous reset. Draw the RTL level diagram and code the design in RTL Verilog.

Q 3: Write RTL Verilog Code to implement Multiply Accumulator (MAC) architecture. The design implements the following:

$$Acc = A * B + C * D + Acc$$

The signals A, B, C, D and Acc are 8, 8, 8, 8 and 32-bit wide unsigned numbers respectively. Verify your logic using a stimulus.

Q 4: Draw RTL level diagram and write ***parameterized*** RTL Verilog code and stimulus for the following digital design:

a. At every positive edge of clock two N-bit signed numbers $a$ and $b$ are input to the system and these are stored in two registers $a\_reg$ and $b\_reg$ respectively.
   Combinational logic in the design computes maximum of these two numbers and the value is stored in a register $max\_reg$.
   A 5*N-bit accumulator register $acc\_reg$ adds the value stored in $max\_reg$ in itself at every clock cycle. Output the value in $acc\_reg$ from the system.
b. Write stimulus to give the following set of inputs to the design in every clock cycle $-5, 10, 2, 55, 102, -111, -90, 10$ print the value in the $acc\_reg$ in every clock cycle.

Q: 5 Sign Magnitude Adder:

An integer can be represented in sign magnitude format, in which the MSB is the sign and the remaining bits form the magnitude. For example, 3 and 3 become "0011" and “1011" in 4-bit sign magnitude format.

A sign magnitude adder performs an addition operation in this format. The operation can be summarized as follows:

- If the two operands have the same sign, add the magnitudes and keep the sign.
- If the two operands have different signs, subtract the smaller magnitude from the larger one and keep the sign of the number that has the larger magnitude.

1. You are required to present a design of the circuit.
2. Write its RTL VERILOG code for a 4-bit Sign Magnitude Adder. (For add operation simply use "+" sign)

# NOTE:

- Code should be properly ***indented*** and ***commented***.
- Write your ***code in MODELSIM***, verify it using ***stimulus*** and ***attach output screen shots of your simulation*** in a separate docx file. Code/Screen shots should not be copied, ***in case of copying assignment will not be marked.***
- MODELSIM Projects (VERILOG files and all) along with screen shots of your simulation results should be ***emailed as a single ".rar" file*** on assignments.sajid@gmail.com.
- ***RAR*** file should be ***named*** as ***"YOURNAME_ASSIGN#_DSD"***.
- Subject of your ***email MUST*** be as ***"YOURNAME_ASSIGN#_DSD"***.
- ***VERY IMPORTANT:*** RTL diagrams should either be drawn in VISIO for software submission, for submission in hard form draw the RTL diagrams with your ***HANDS instead of FEET.***