UNIVERSITY OF RUHUNA

Faculty of Engineering

End-Semester 6, Examination in Engineering, July 2025

Module No: EE6205 Module Name: Hardware Description Language (C-18)

[3 hours]

[Answer all questions. Marks allocated are indicated after each question.]

Q1 Consider the following Verilog module, modeled at RTL.

```
module unknown1( input wire a, input wire b, output wire c);
    assign c = (a & b) | (~a & ~b);
endmodule
```

a) Write down the corresponding truth table and draw the circuit diagram. [2 Marks]

b) Modify the module code by using

i) behavioral modeling, and

ii) gate-level modeling.
Note: Primitive for gate instant is `GateName GateInstant(Out, In1, In2, ..);`
Eg. **nand** mynand(y, a, b); where, y is the output, a and b are inputs of the NAND.

[4 Marks]

c) The testbench for the module is listed below.

i) What does the testbench display on the console?

ii) Accurately draw the waveform output (timing diagram for a, b and c) of the testbench.

[4 Marks]

```
'timescale 1ns/1ps
module tb();
    reg a; reg b; wire c;
    unknown1 uut(.a(a), .b(b), .c(c));
    initial begin
        $dumpfile("dump.vcd"); $dumpvars(0, tb);
        a = 1'b0; b = 1'b0;
        #10; a = 1'b0; b = 1'b1;     #10; a = 1'b1; b = 1'b1;
        #10; a = 1'b1; b = 1'b0;     #10; a = 1'b0; b = 1'b0;
    end

    initial begin
        $monitor("a=%1b, b=%1b, c=%1b", a, b, c);
    end
endmodule
```

Q2 a) Explain the difference between these constructs with examples.

i) **initial** block vs **always** block

ii) **wire** vs **reg**

[4 Marks]

b) Explain why the following code may cause synthesis issues. Rewrite the code snippet that resolves identified issues.

```
always @ (posedge clk or in1 or in2) begin
    if (in1) q<= 1'b1;
    else if (in2) q<= 1'b0;
end
```

[2 Marks]

c) A 1-bit Full Adder module is given below.

```
module fulladder_1b (
    input wire a,
    input wire b,
    input wire cin , // input carry bit
    output wire s,   // results
    output wire cout // resulting carry bit
    );
    assign {cout , s}= cin + a + b;
endmodule
```

Write the code for 8-bit Full Adder modules named

i) fulladder_8b(), by modifying fulladder_1b().

ii) fulladder_8b_m(), by using 8 instances of fulladder_1b().

[4 Marks]

Q3 a) Following module implements a D-Flipflop. Assume that the propagation delay of the D-Flipflop is much less than the clock (clk) period.

```
module dflipflop(
    input wire d,       // D data input
    input wire clk,     // CLK clock input
    input wire en,      // Enable (the IC)
    input wire reset,   // RESET (active-high)
    output reg q        // Q output
    );

    always @ (posedge clk or posedge reset) begin
        if (reset)   q <= 0;
        else if(en)  q <= d;
    end
endmodule
```

i) How does the D-Flipflop given in this code function? Describe by giving sketches of all corresponding I/O wave forms (timing diagrams).

[2 Marks]

ii) Write a testbench, named as tb_dff(), to test the D-Flipflop for all possible combinations of I/O signals.

[2 Marks]

iii) Modify the **always** block to make the resetting asynchronous. You may use the following code snippet to generate clock signal
always #(CLK_PERIOD/2) clk = ~clk;.

[2 Marks]

b) Write a module code to make a simple circuit that *on* and *off* an LED every second. Assume that the frequency of the input clock is 50MHz and the LED is connected to the output of the circuit.

[2 Marks]

c) Write a module named bus() to implement the circuit given in Figure Q3(c).

[2 Marks]

Q4 a) i) Illustrate propagation delay t_{pd} and contamination delay t_{cd} in a timing diagram, drawn for a NOT gate.

ii) Find the propagation delay and contamination delay of the circuit shown in Figure Q4(a), if each gate has a propagation delay of 60 ps and a contamination delay of 100 ps. Redraw the circuit while indicating a critical path and a shortest path.

iii) Describe what are the symbols X - the illegal value, and Z - the floating value, mean at the level of digital electronics circuit implementation.

[6 Marks]

b) i) Draw a block diagram of a typical FPGA Configurable Logic Block (CLB) that consists of lookup table (LUT), D-Flipflop and a Multiplexer. Describe its functionality.

ii) What are the other essential and fundamental building blocks of an FPGA?

[4 Marks]

Q5 Design a Finite State Machine to detect the sequence 1101 with; Mealy machine (output depends on both the current state and input) implementation, overlap detection and error state for invalid sequence.

a) Provide state transition diagram. [5 Marks]

b) Write the corresponding Verilog code [5 Marks]

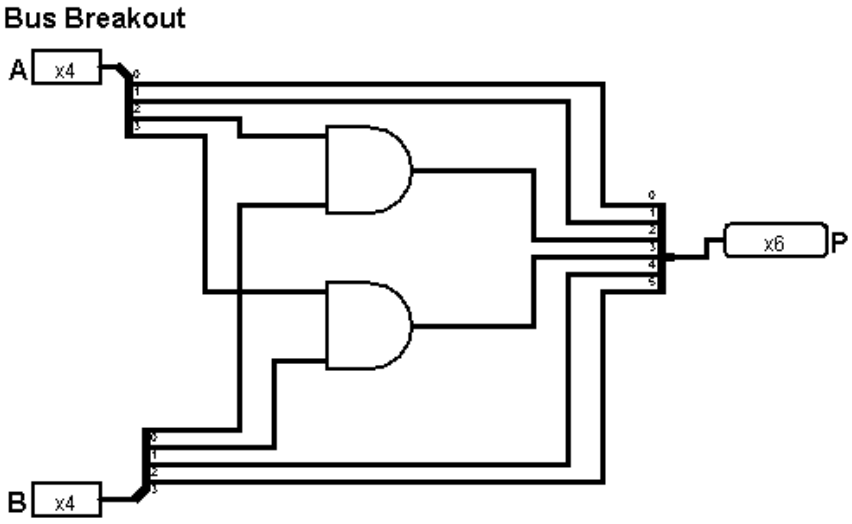

Figure Q3(c): Bus breakout. A and B are 4-bit inputs while P is a 6-bit output.

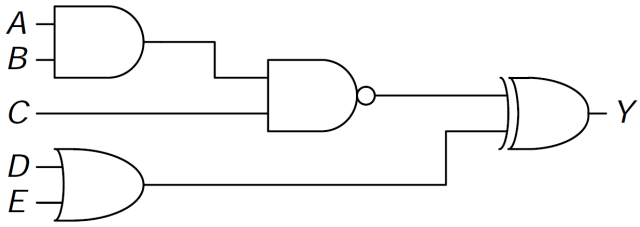

Figure Q4(a): Calculate the total propagation and contamination delays for the circuit. For each of the gates, $t_{cd} = 100ps$ and $t_{pd} = 60ps$