UNIVERSITI
TEKNOLOGI
PETRONAS

FINAL EXAMINATION
JANUARY 2025 SEMESTER

COURSE : **EFB4133 - DIGITAL SYSTEM DESIGN**

DATE : **12 APRIL 2025 (SATURDAY)**

TIME : **9.00 AM - 12.00 NOON (3 HOURS)**

INSTRUCTIONS TO CANDIDATES

1. Answer **ALL** questions in the Answer Booklet.
2. Begin **EACH** answer on a new page in the Answer Booklet.
3. Indicate clearly answers that are cancelled, if any.
4. Where applicable, show clearly steps taken in arriving at the solutions and indicate **ALL** assumptions, if any.
5. **DO NOT** open this Question Booklet until instructed.

Note :

i. There are **SEVEN (7)** pages in this Question Booklet including the cover page .

ii. **DOUBLE-SIDED** Question Booklet.

iii. **Engineering Data & Formulae Booklet will be provided.**

1. a. The Verilog code in **FIGURE Q1** is written using the dataflow modelling style. Rewrite the full Verilog code to describe the same circuit by using:

   i. Structural modelling style

[5 marks]

   ii. Behavioural modelling style

[5 marks]

```
module complex_logic_circuit(a, b, c, d, e, f);
   input a, b, c, d, e;
   output f;

   assign f = ((a & b) | (~c & d)) ^ (e & (b | d));
endmodule
```

FIGURE Q1

b. Identify and discuss one advantage and one disadvantage of using behavioural modelling compared to structural modelling in Verilog.

[5 marks]

2. A digital system consists of a datapath and a control unit that work together to execute operations. Consider a simple 4-bit processor that performs arithmetic and logic operations using a single-cycle control unit.

a. Design a datapath for a simple 4-bit processor that operates on 4-bit data and supports the following instructions. Your design should include a datapath diagram showing key components such as registers, ALU, multiplexers, and control signals. Assume the processor has four general-purpose registers (R0-R3), and immediate values are 4 bits wide.

Supported Instructions:

- ADD R1, R2 → R3 (4-bit addition: R3 = R1 + R2)
- SUB R1, R2 → R3 (4-bit subtraction: R3 = R1 - R2)
- AND R1, R2 → R3 (4-bit bitwise AND: R3 = R1 & R2)
- OR R1, R2 → R3 (4-bit bitwise OR: R3 = R1 | R2)
- LOAD IMM → R1 (Load a 4-bit immediate value into register R1)

[15 marks]

b. The control unit generates control signals to coordinate the datapath operations. Identify and describe **FOUR (4)** key control signals required to execute the given instructions. Clearly explain how each signal influences the datapath components (e.g., ALU, registers, multiplexers, or memory access).

[5 marks]

c. Design a state transition diagram for a simple single-cycle control unit that fetches, decodes, and executes only the given instructions. Clearly show the states and their transitions, and label key control signals in each state.

[10 marks]

3. **FIGURE Q3** shows a sequential circuit that consists of edge-triggered D flip-flops.

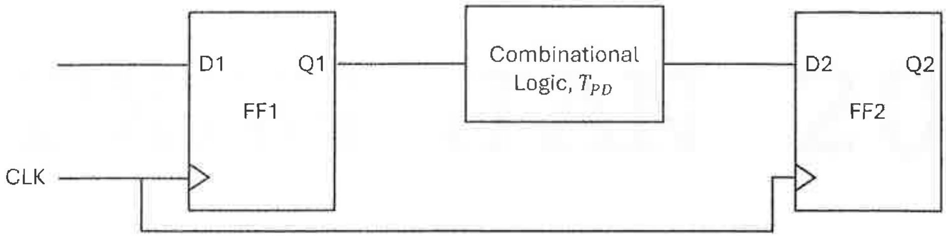

FIGURE Q3

Timing Parameters:

Setup time, T_s = 150 ps

Hold time, T_h = 100 ps

Clock-to-Q delay, T_{CQ} = 200 ps

Combinational circuit propagation delay, T_{pd} = 500 ps

Clock period, T_{clk} = 1.2 ns (1200 ps)

Clock skew, T_{skew} = 50 ps

a. Determine whether the circuit satisfies the setup time requirement. If not, find the minimum clock period needed to avoid setup time violations.

[5 marks]

b. Check whether the circuit meets the hold time requirement. If not, determine the additional delay required to prevent a hold time violation.

[5 marks]

c. Suppose the clock period is reduced to 1.0 ns to improve performance. Analyze whether the system will still function correctly. If not, suggest a suitable clock period.

[5 marks]

d. Analyze the impact of clock skew on setup and hold time constraints. Determine the maximum clock skew the circuit can tolerate before violating either requirement. Justify your answer with calculations.

[8 marks]

4. A 4-bit ALU has been designed to perform arithmetic and logical operations based on a 2-bit opcode. The ALU has the functionality listed in **TABLE Q4**.

TABLE Q4

Opcode	Operation	Description
00	A + B	Addition
01	A – B	Subtraction
10	A & B	Bitwise AND
11	A	B

You are given the ALU design code, as written in **FIGURE Q4a**:

```
module ALU (
   input [3:0] A, B,
   input [1:0] opcode,
   output reg [3:0] result
);
   always @(*) begin
      case (opcode)
         2'b00: result = A + B;
         2'b01: result = A - B;
         2'b10: result = A & B;
         2'b11: result = A | B;
         default: result = 4'b0000;
      endcase
   end
endmodule
```

FIGURE Q4a

a. Explain why a testbench is necessary for verifying the ALU design. Identify two potential issues that could arise if the ALU design is not properly verified.

[5 marks]

b. Write a Verilog testbench to verify that the ALU performs correctly for all 4 operations. The testbench should include test cases for different values of A and B and a self-checking mechanism that compares expected vs actual results. Your testbench code must follow the structure written in **FIGURE Q4b**.

```
`timescale 1ns/1ps
module ALU_tb;
        // Declare testbench variables
        // Instantiate the ALU module
        // Task to verify ALU operation
        task check_result;
          begin
          end
        endtask
        // Apply stimulus
        initial begin
          $display("Starting ALU Testbench...");
          // Test ADDITION (A + B)
          // Test SUBTRACTION (A - B)
          // Test AND (A & B)
          // Test OR (A | B)
          $display("ALU Testbench Completed.");
          $stop; // Stop simulation
        end
endmodule
```

FIGURE Q4b

[15 marks]

c. If the simulation shows that A - B does not work correctly, suggest one possible design issue and how to fix it.

[5 marks]

d. Modify the ALU design so that it outputs 0000 for an invalid opcode (>= 4).

[7 marks]

-- END OF PAPER --