Universiti Tun Hussein Onn Malaysia

UNIVERSITI TUN HUSSEIN ONN MALAYSIA

FINAL EXAMINATION
SEMESTER II
SESSION 2023/2024

COURSE NAME	:	DIGITAL DESIGN
COURSE CODE	:	BEJ 30503
PROGRAMME CODE	:	BEJ
EXAMINATION DATE	:	JULY 2024
DURATION	:	3 HOURS
INSTRUCTIONS	:	1. ANSWER ALL QUESTIONS FROM PART A AND ONE QUESTION FROM PART B 2. THIS FINAL EXAMINATION IS CONDUCTED VIA ☐ Open book ☒ Closed book 3. STUDENTS ARE **PROHIBITED** TO CONSULT THEIR OWN MATERIAL OR ANY EXTERNAL RESOURCES

THIS QUESTION PAPER CONSISTS OF **EIGHT (8)** PAGES

PART A

Q1 Digital design is a process of designing a digital system from specifications rather than by using a general-purpose processor.

(a) Discuss the advantage of digital design as compared to general-purpose processors in specific application such as video processing.

(4 marks)

(b) Computer-Aided Design (CAD) tools not only simplifies the design process but also allows for designing complex digital systems. However, explain why the designer must have an in-depth understanding of the relevant theories and design methodologies to design a circuit effectively rather than only relying on CAD tools.

(4 marks)

(c) Field Programmable Gate-Array (FPGA) is very popular for fast prototyping of digital circuits as it supports the implementation of relatively large logic circuits. In FPGA, the logics are implemented using the Look-up Table (LUT). Explain the advantage of using LUT instead of using the AND or the OR planes.

(2 marks)

Q2 Verilog HDL is a hardware description language used to model and design digital systems. It's widely used in digital design, particularly for describing and simulating electronic systems at the register-transfer level (RTL) of abstraction.

(a) With a suitable example, explain Verilog modelling styles in describing digital circuits.

(9 marks)

(b) **Figure Q2.1** shows the Verilog program describing a digital circuit. Rewrite the program by using behavioural modelling.

```
module Q2b(En, w, y);
        input En;
        input [1:0]w;
        output [0:3]y;

        and(y[0], En, ~w[1], ~w[0]);
        and(y[1], En, ~w[1],  w[0]);
        and(y[2], En,  w[1], ~w[0]);
        and(y[3], En,  w[1],  w[0]);

endmodule
```

Figure Q2.1

(8 marks)

(c) A combinational logic module, **Q2c**, is described by the Verilog code in **Figure Q2.2**. It comprises a logic component, **Q2slice**.

```
module Q2c(in3, in2, in1, in0, flag, out3, out2, out1, out0);
    input in3, in2, in1, in0;
    output flag, out3, out2, out1, out0;
    wire co0, co1;

    Q2slice M0(in3, in0, in3, out0, co0);
    Q2slice M1(in3, in1, co0, out1, co1);
    Q2slice M2(in3, in2, co1, out2, flag);

    assign out3 = in3;

endmodule

module Q2slice(in3, ini, bi, s, co);

    input in3, ini, bi;
    output s, co;
    wire ai;
    assign ai = in3 ^ ini;
    assign s = ai ^ bi;
    assign co = ai & bi;

endmodule
```

Figure Q2.2

(i) Sketch the logic circuit of **Q2slice** component.

(4 marks)

(ii) Draw the Functional Block Diagram (FBD) of module **Q2c**, showing how the **Q2slice** components are interconnected.

(7 marks)

(iii) If the input vector, (*in*3, *in*2, *in*1, *in*0) is 0101, what is the value of the output vector, (*flag*, *out*3, *out*2, *out*1, *out*0)?

(2 marks)

Q3 **Figure Q3.1** shows an adder-subtractor circuit, where FA is a full adder module. From the figure, answer the following questions:

(a) Explain how the addition or subtraction is implemented.

(4 marks)

(b) Given a = 1011, b = 0001 and c_{in} = 1, determine the value of s and c_{out}. Furthermore, verify your answer by providing the calculation in decimal.

(3 marks)

(c) From the circuit, explain how to generate the overflow signal.

(3 marks)

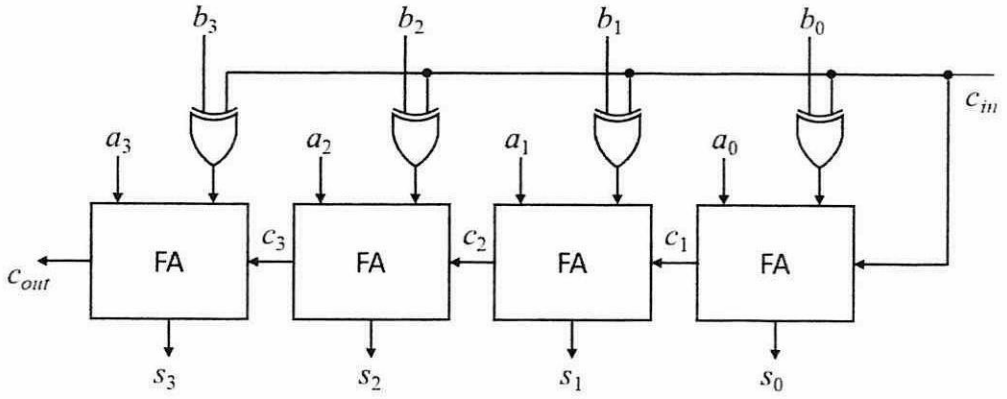

Figure Q3.1

Q4 **Figure Q4.1** shows a Verilog program describing a sequential circuit. By referring to the program, answer questions **Q4(a)** to **Q4(c)**:

(a) Name the assignment type (*blocking* or *non-blocking*) that is used in the code, i.e., in the "$Q1$ <= D" statement.

(2 marks)

(b) Draw the circuit that is synthesized from the codes.

(6 marks)

(c) Draw the output signal at $Q1$, $Q2$ and $Q3$ based on the timing diagram given in **Figure Q4.2**.

(6 marks)

```
module Q4c(clk, D, Q1, Q2, Q3);
    input clk, D;
    output reg Q1, Q2, Q3;
    always@(posedge clk)
        begin
            Q1 <= D;
            Q2 <= Q1;
            Q3 <= Q2;
        end
endmodule
```

Figure Q4.1

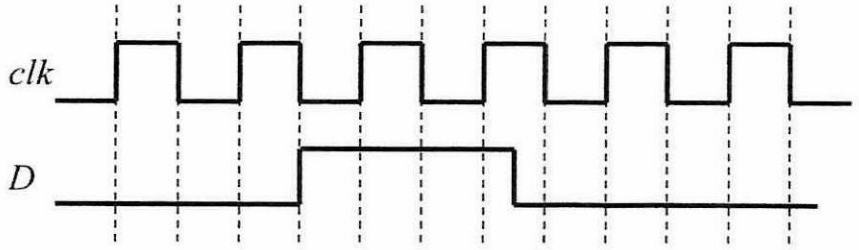

Figure Q4.2

(d) **Figure Q4.3** shows a Verilog program describing an 8-bit register with synchronous reset signal. A timing diagram in **Figure Q4.4** showing the input data at *A*. Based on the timing diagram, draw the output signal *Q*.

```
module Q4d(clk, rst, A, Q);
        input clk, rst;
        input [7:0] A;
        output reg [7:0] Q;

        always@(posedge clk)
            if (rst)
                Q <= 0;
            else
                Q <= A;
endmodule
```

Figure Q4.3

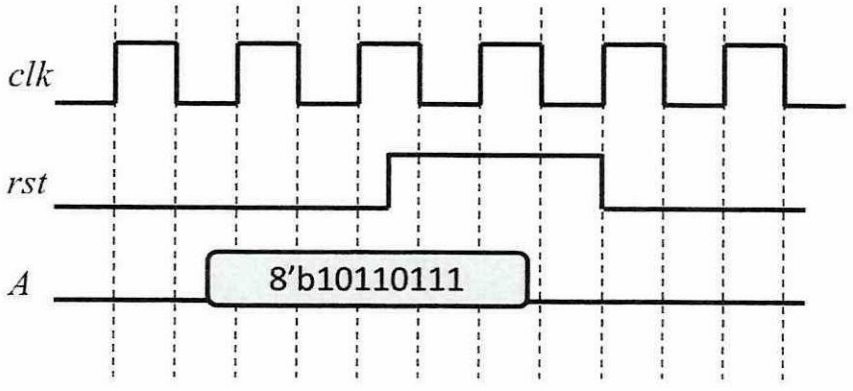

Figure Q4.4

(4 marks)

(e) For the circuit in **Figure Q4.5**, determine the output Q at the next positive clock edge if $L = 1$, $w = 1$, and $R = 1011$.

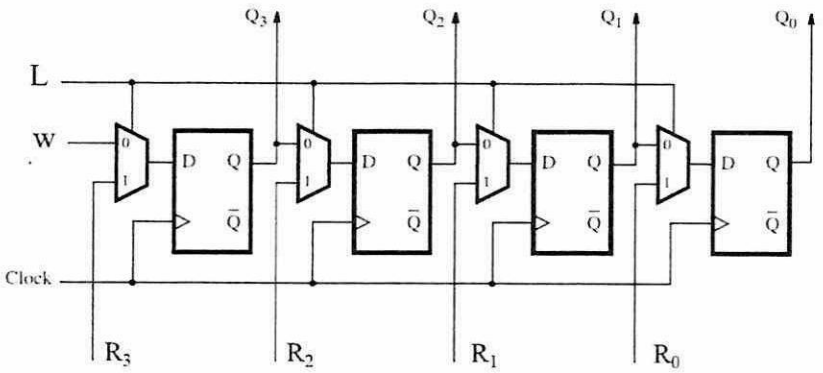

Figure Q4.5

(2 marks)

PART B

Q5 **Figure Q5.1** shows a datapath and a controller of a digital system, containing two 8-bit registers *R*0 and *R*1, an Adder-Subtractor circuit, and a 1KB memory. The controller is used to control the operation of the datapath. The *start* signal is asserted to start the operation, while the *done* signal is used to indicate the completion of the operation.

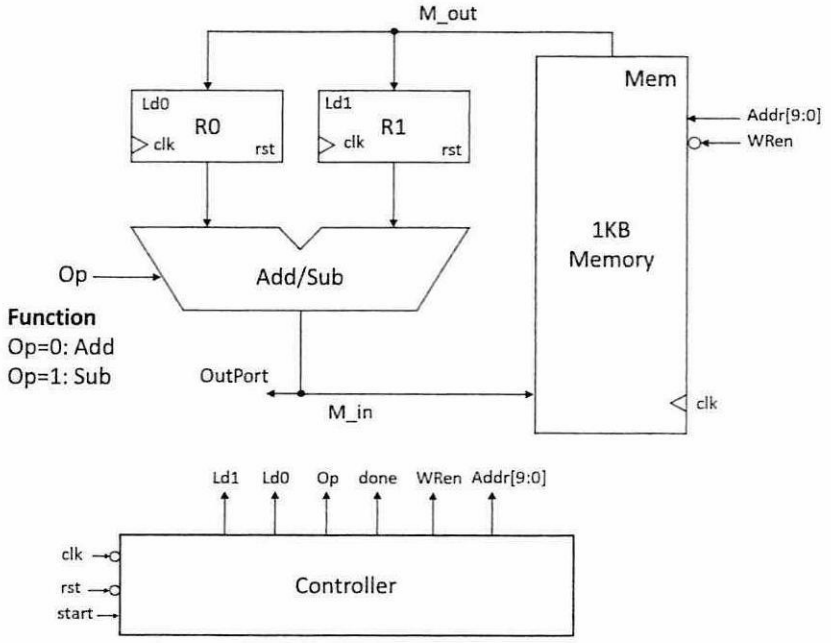

Figure Q5.1

(a) Write a Verilog program to describe the datapath circuit.

(10 marks)

(b) Derive the RTL codes to represent the operations below:

1. Load the data from memory location 0x100 into *R*0.
2. Load the data from memory location 0x200 into *R*1.
3. Add the content of *R*1 and *R*2 and the result is then stored in memory location 0x100.

(5 marks)

(c) Derive the Algorithmic State Diagram (ASM) chart to implement the operation in **Q5(b)**.

(8 marks)

(d) Draw the functional block diagram (FBD) of the controller by showing the state register, input, and output signals.

(7 marks)

Q6 A fully dedicated architecture (FDA) that implement the following computation is to be designed.

$$y = \sum_{i=1}^{4} c_i x_i$$

(a) Draw the data flow graph for the FDA.

(5 marks)

(b) Derive the as-late-as-possible (ALAP) schedule, and resource allocation for a minimum hardware implementation.

(10 marks)

(c) Derive the RTL codes for your design.

(7 marks)

(d) From the RTL codes in **Q6(c)**, draw and label the FBD of the datapath of the design.

(8 marks)

- END OF QUESTIONS -