

Two hours - online

This paper version is made available as a backup
In this event, only MCQ answers written in the boxes on the exam paper will be marked.

EXAM PAPER MUST NOT BE REMOVED FROM THE EXAM ROOM

**UNIVERSITY OF MANCHESTER
SCHOOL OF COMPUTER SCIENCE**

Fundamentals of Computer Engineering

Date: Friday 18th January 2019

Time: 14:00 - 16:00

This is an online examination. Please answer all Questions.

© The University of Manchester, 2019

This is a CLOSED book examination

The use of electronic calculators is permitted provided they are not programmable and do not store text

[PTO]

*Section A
contains Multiple
Choice Questions
and is restricted*

Section B**Answer both questions**

14. a) What is the difference between blocking and non-blocking assignments in Verilog? In what type of Verilog block will you find blocking or non-blocking assignments? For what type of circuit should you use non-blocking assignments? What data type is associated with variables assigned using blocking and non-blocking assignments? (5 marks)
- b) What is a multiplexer? Is a multiplexer a combinatorial or sequential logic device? Produce a Verilog statement, using a continuous assignment, that provides the functional operation of a 2:1 multiplexer. (4 marks)
- c) Discuss the operation of the D-type flip-flop, paying particular attention to the role of the inputs for everyday operation. (2 marks)
- d) The following Verilog module implements a simple D-type flip-flop design. Produce a modified module description that incorporates an additional input control signal which controls on which rising edges of the clock the flip-flop is updated. (3 marks)

```

module ff_1bit(input      D, clk,
              output reg  Q, Qbar);

    always @ (posedge clk)
    begin
        Q <= D;
        Qbar <= ~D;
    end

endmodule

```

- e) Produce a Verilog description of a 16-bit register. The register has a clock enable input that is active-low and an active-low reset signal that acts asynchronously. (6 marks)

15. a) Figure Q15.1 shows one possible implementation of the MU0 datapath. Briefly discuss the roles of the following components in the operation of MU0:
- i. the PC, (1 mark)
 - ii. the IR, (1 mark)
 - iii. the Control. (1 mark)

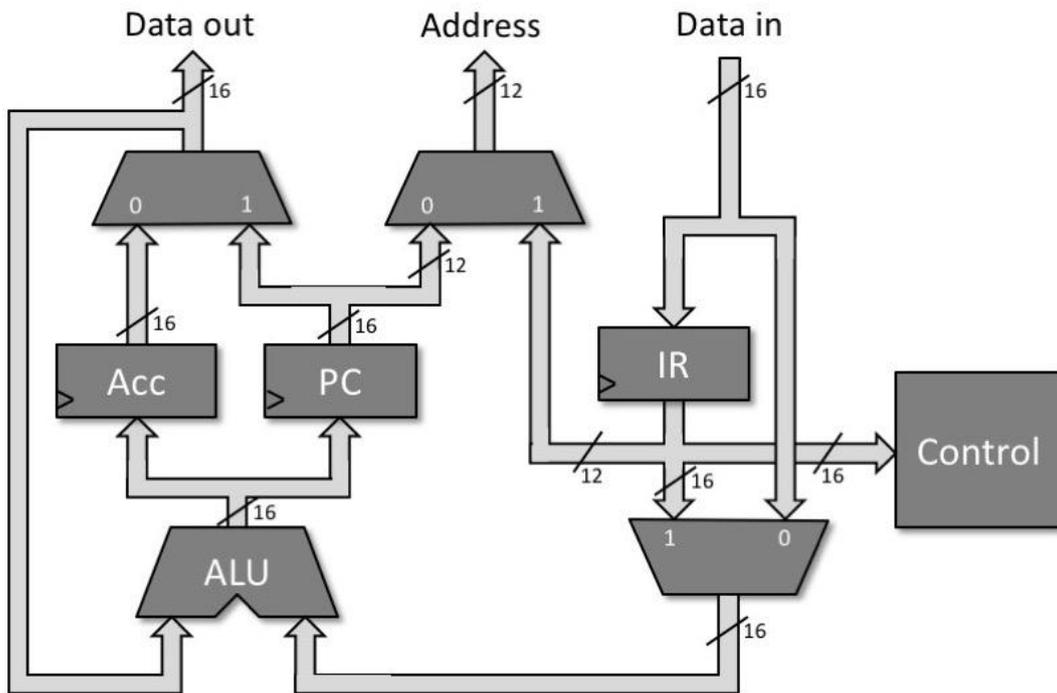


Figure Q15.1

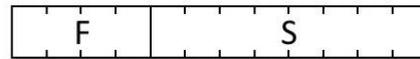
- b) In MU0 an instruction is executed over two stages: fetch and execute. Briefly describe what happens in each of these two stages. (3 marks)
- c) If the external clock to MU0 has a frequency of 20 MHz, how long will it take to execute a single instruction? (1 mark)
- d) Figure Q15.2 illustrates the MU0 instruction format and the MU0 instruction set.

Provide a sequence of MU0 instructions which can be used to add the data values stored at two memory locations, &0A0 and &0A1, and store the result at memory address &0A2. (3 marks)

If your first instruction is held in memory address &010, what will be the value held in the PC after your last instruction has been executed? (1 mark)

[PTO]

MU0 instruction format



MU0 instruction set

F	Mnemonic	Description
0	LDA S	Acc := [S]
1	STA S	[S] := Acc
2	ADD S	Acc := Acc + [S]
3	SUB S	Acc := Acc - [S]
4	JMP S	PC := S
5	JGE S	If Acc ≥ 0, PC := S
6	JNE S	If Acc ≠ 0, PC := S
7	STP	Halt

Figure Q15.2

- e) The MU0 arithmetic logic unit, ALU, contains a ripple carry adder. Briefly discuss the limitations of the ripple carry adder design. (2 marks)
- f) The MU0 ALU contains a preconditioner that configures the two 16-bit inputs, input_X and input_Y, to form the two 16-bit inputs to the ALU, output_X and output_Y, for the required ALU operation, as shown in figure Q15.3. The four operations are determined by the input control bus M as shown in the table in Figure Q15.4, which also shows the value of Cin for each operation.

As a designer you are required to complete the design of the preconditioner unit as a Verilog module. The module is defined for you as:

```

module precon(input      [15:0]  input_X, input_Y,
              input      [1:0]   M,
              input      Cin,
              output reg  [15:0]  output_X,
              output_Y);

// your behavioural code here

endmodule
  
```

Produce the Verilog behavioural code to describe the operation of the preconditioner unit in setting the outputs, output_X and output_Y, to perform the required operations. (7 marks)

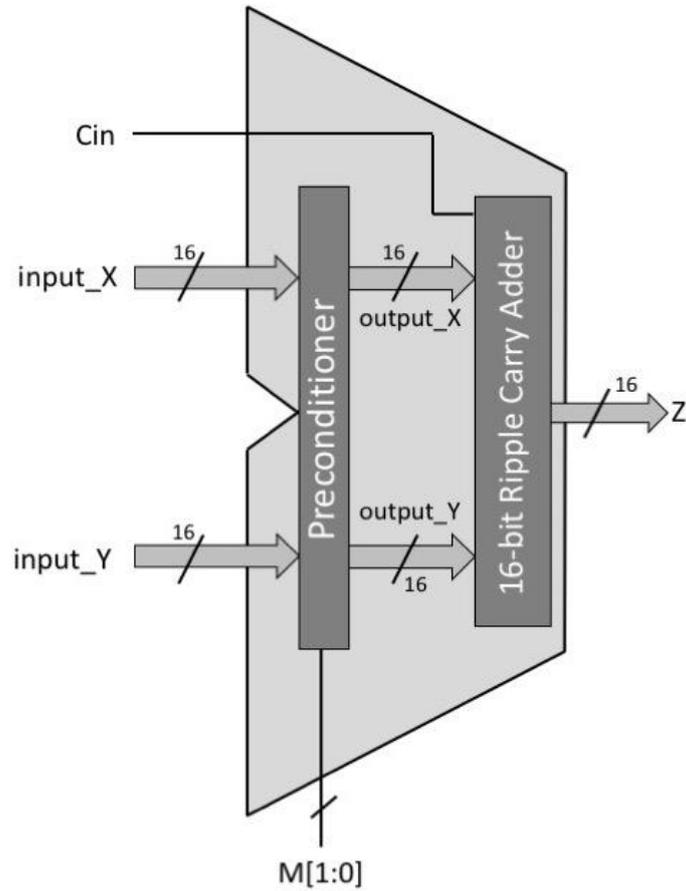


Figure Q15.3

$M[1:0]$	Action	Cin
00	Add	0
01	Sub	1
10	Inc ($X+1$)	1
11	Pass ($=Y$)	0

Figure Q15.4

END OF EXAMINATION