Paper Number(s): **E1.2**

IMPERIAL COLLEGE LONDON

DEPARTMENT OF ELECTRICAL AND ELECTRONIC ENGINEERING
EXAMINATIONS 2008

EEE/ISE Part I: MEng, BEng and ACGI

Corrected Copy

DIGITAL ELECTRONICS I

Monday, 2 June: 10:00 am

Time allowed: 2.00 hours

There are FOUR questions on this paper.

Question 1 is compulsory.
Answer Question 1 and any two of Questions 2-4.
Q1 carries 40% of the marks. Questions 2 to 4 carry equal marks (30% each).

Any special instructions for invigilators and information for candidates are on page 1.

Examiners responsible:

Paper Setter: Cheung, P.
First Marker(s): Chaudhuri, B.
Second Marker(s): Durrani, Z.

This page is intentionally left blank.

Special instructions for invigilators: None

Information for candidates:

In the figures showing digital circuits, all components have, unless explicitly indicated otherwise, been drawn with their inputs on the left and their outputs on the right. All signals labelled with the same name are connected together. All circuits use positive logic. The least significant bit of a bus signal is labelled as bit 0, and the most significant bit with the highest integer number. Therefore the signal X[7:0] is an eight bit bus with X7 being the MSB and X0 the LSB.

In questions involving circuit design, you may use any standard digital circuits that are not explicitly forbidden by the question provided that you fully specify their operation.

Marks may be deducted for unnecessarily complex designs unless you are explicitly instructed not to simplify your solution.

[Question 1 is compulsory]

1. a) Simplify the following Boolean expressions using De Morgan's theorem and/or Boolean algebra.

i) $ABC + \overline{A}CD + \overline{B}CD$

[4]

ii) $AB + (C + \overline{B})(AB + \overline{C})$

[4]

b) Simplify the following Boolean equation using a Karnaugh map.

$$F(X,Y,Z) = \sum(1,3,4,5,6)$$

[4]

c) Express the Boolean function as depicted by the Karnaugh map shown in Figure 1.1 as a Boolean equation in Product-of-Sum form.

[4]

AB \ CD	00	01	11	10
00	1	1	0	1
01	0	1	0	0
11	0	0	0	0
10	1	1	0	1

Figure 1.1

d) The timing waveforms for signals A, B and C shown in Figure 1.2 are applied to the circuit shown in Figure 1.3. Draw the timing waveform for the output signal X. (It may help if you first derive the timing waveforms for the intermediate signals.)

[4]

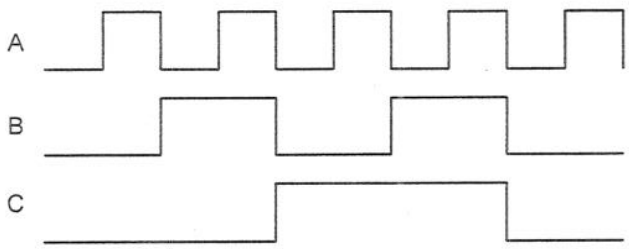


Figure 1.2

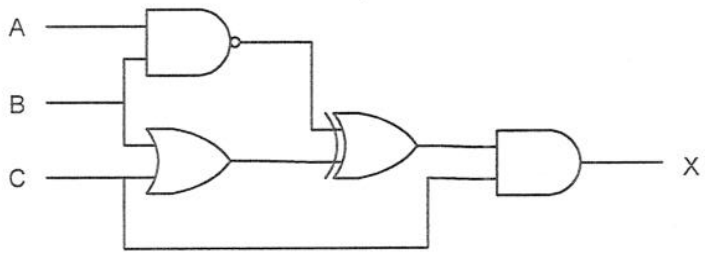


Figure 1.3

e) Given that the decimal code for the ASCII character 'A' is 65, and assuming that all numbers can be represented using 8 bit binary, complete the missing entries which are not shaded in the following table. (No marks will be awarded for this question unless you show how the solutions are derived.)

[8]

Binary	Hexadecimal	Unsigned Decimal	Signed Decimal	ASCII
?		70		?
	D4	?	?	

f) The timing waveforms for signals A and B shown in Figure 1.4 are applied to the circuit shown in Figure 1.5. Copy the timing diagram and add the timing waveforms for P, Q, R and S. Assume that initially R = 0 and S = 1.

[6]

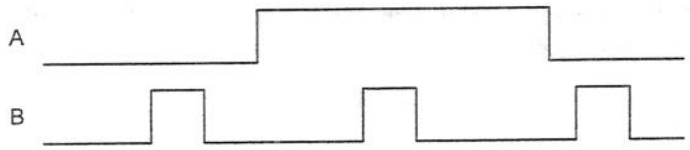


Figure 1.4

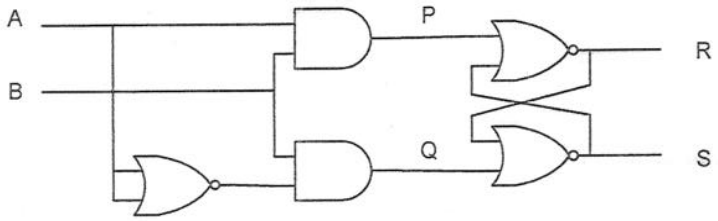


Figure 1.5

g) Figure 1.6 shows the circuit of a finite-state machine (FSM) with four states It consists of two D-type flip-flops and two gates. Assume that the initial values of S0 and S1 are zero, draw the state transition diagram for this FSM.

[6]

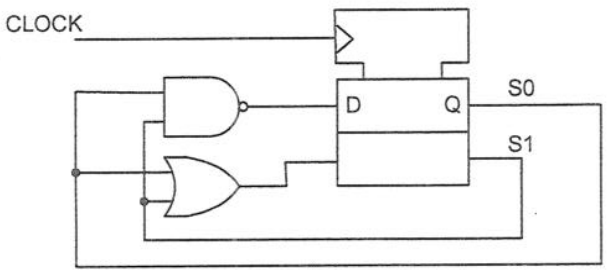


Figure 1.6

2. The 1-bit full subtractor shown in Figure 2.1 a) performs the binary subtraction $(P-Q-B_{in})$ where P and Q are 1-bit variables and B_{in} is borrow input from the previous stage. It produces two outputs: the difference D and the borrow output B_{out}. The truth table describing the function of this 1-bit full subtractor is shown in Figure 2.1 b).

   a) Derive the Boolean equations for D and B_{out}.

   [8]

   b) Implement these equations using only NOR gates.

   [8]

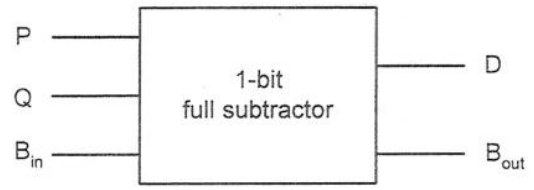
   c) Design a 4-bit unsigned binary subtractor using four 1-bit full subtractor circuits. What is the meaning of borrow output from this 4-bit subtractor?

   [8]

   d) Hence or otherwise, design a circuit that provides the maximum value of two 4-bit unsigned binary numbers M and N i.e.

$$X = \max(M, N).$$

   [6]



a)

B_{in}	P	Q	D	B_{out}
0	0	0	0	0
0	0	1	1	1
0	1	0	1	0
0	1	1	0	0
1	0	0	1	1
1	0	1	0	1
1	1	0	0	0
1	1	1	1	1

b)

Figure 2.1

3. Two alternative schematic representations of a 2-to-1 multiplexer (MUX) are shown in Figure 3.1 with its function described as a Boolean equation.

   a) Figure 3.2 a) shows a logic cell made up of three 2-to-1 MUXs and has 7 inputs and 1 output. Given that such a logic cell is connected as shown in Figure 3.2 b), using Boolean algebra, Karnaugh map or any other method, derive the Boolean equation for Z in terms of input variables A, B, C and D.

   [10]

   b) The logic cell shown in Figure 3.2 a) is used to implement a new Boolean equation

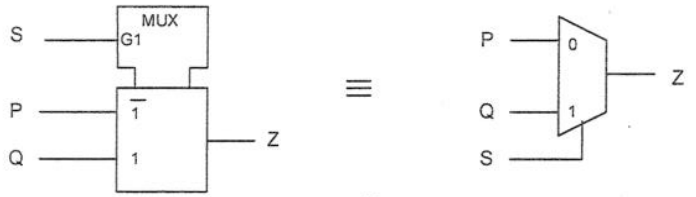
   $$Z = A\overline{D} + A\overline{B}D + BCD\,.$$

   How would you connect the inputs of the logic cell?

   [10]

   c) Figure 3.3 a) shows a sequential circuit constructed with gates and MUXs. The inputs W, X and Y are driven by signals as shown in Figure 3.3 b). Assuming that the value of the output Z is initial '0', sketch the waveforms showing all the input and output signals.

   [10]



$$Z = P\overline{S} + QS$$

Figure 3.1

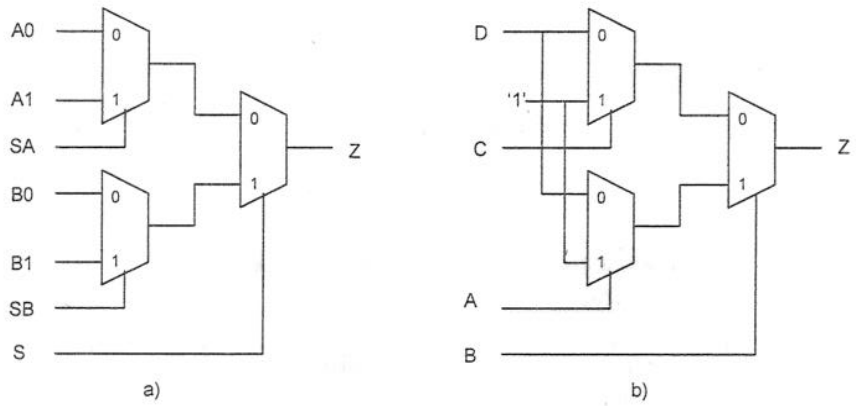


Figure 3.2

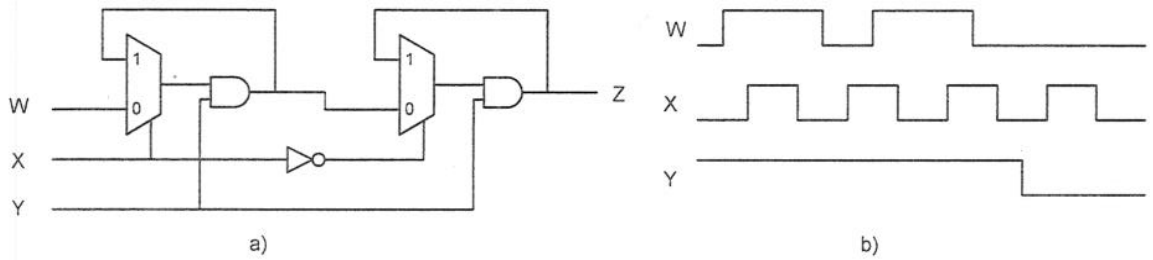


Figure 3.3

4. Figure 4.1 shows the state diagram of a Mealy finite state machine (FSM) with one input X and output signal Y.

   a) Draw the state transition table for the FSM.

   [6]

   b) Hence, or otherwise, design a circuit to implement this FSM using two D flip-flops and logic gates. Your design should be in the form of Boolean equations. There is no need to draw the detail gate level circuit diagram.

   [14]

   c) Instead of using two D flip-flops and binary code to encode the four states, it is decided to use four D flip-flops and a new state encoding method (known as 'one-hot' encoding) as shown in figure 4.2.

   Re-design the FSM using this new state encoding in the form of Boolean equations.

   [10]

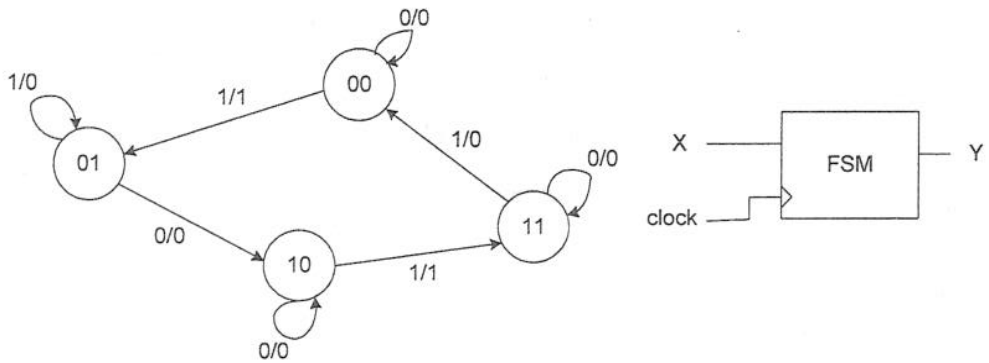


Figure 4.1

Original state encoding	New state encoding
00	0001
01	0010
10	0100
11	1000

Figure 4.2

[THE END]