IMPERIAL COLLEGE LONDON

DEPARTMENT OF ELECTRICAL AND ELECTRONIC ENGINEERING
EXAMINATIONS 2009

EEE/ISE PART I: MEng, BEng and ACGI

Corrected Copy Q3

DIGITAL ELECTRONICS 1

Thursday, 28 May 10:00 am

Time allowed: 2:00 hours

There are FOUR questions on this paper.

Q1 is compulsory.
Answer Q1 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 First Marker(s) : N.P. Sedcole, N.P. Sedcole
Second Marker(s) : Z. Durrani, Z. Durrani

Special instructions for students

In the figures showing digital circuits, all components have, unless explicitly indicated otherwise, been drawn with their inputs on the left or top and their outputs on the right or bottom. 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.

The Questions

[Question 1 is compulsory]

1. a) Simplify the following two Boolean expressions using Boolean algebra:

   i) $\overline{(A+B)}(\overline{A}+C)$

   ii) $A\overline{B}\overline{C}+\overline{B}C+\overline{A}BC$

   [6]

   b) Convert:

   i) the decimal number 70 into unsigned 8-bit binary and BCD forms

   ii) the 2's complement binary number 1010000 into decimal and 12-bit binary forms.

   [6]

   c) For the circuit shown in Figure 1.1, write out the truth table. Hence, or otherwise, write the canonical sum-of-products expression for Z.

   [6]

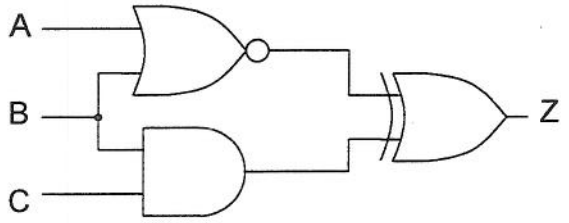


Figure 1.1

   d) For the circuit shown in Figure 1.2(a), complete the waveforms for Y and Z given the waveforms of the inputs A and B in Figure 1.2(b).

   [6]

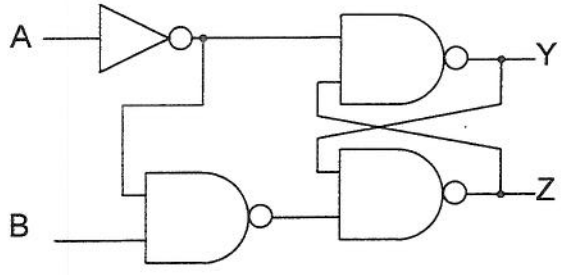


Figure 1.2(a)

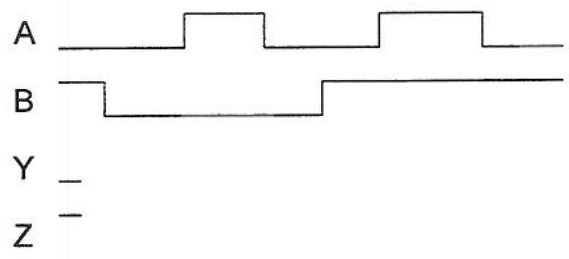


Figure 1.2(b)

e) The truth table below defines the functions F and G in terms of Boolean variables A, B and C. Using Karnaugh maps, derive minimal sum-of-products expressions for F and G

A	B	C	F	G
0	0	0	0	1
0	0	1	0	1
0	1	0	1	0
0	1	1	1	0
1	0	0	0	X
1	0	1	0	0
1	1	0	0	1
1	1	1	1	0

[6]

f) For the sequential circuit shown in Figure 1.3, draw the state diagram.

[6]

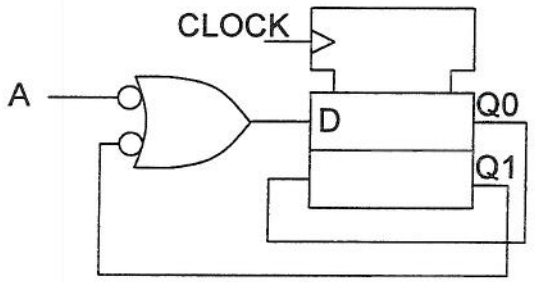


Figure 1.3

g) A ROM has 5 address inputs and 4 data outputs. What is the capacity of the ROM in bits?

[4]

2. A binary number can be multiplied by a constant amount by shifts and additions. For example, multiplying a number A by 5 can be achieved by adding A and 4A. The 9x circuit in Figure 2.1 multiplies a 4-bit signed (two's complement) number A[3:0] by 9 to produce an N-bit number P.

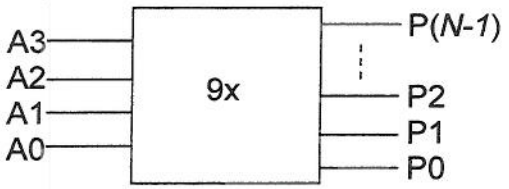


Figure 2.1

a) Show that the minimum value of N, the number of bits required to represent P, must be 8.
[5]

b) Figure 2.2 shows a full adder which has been connected to signals D and C_i and produces output signals S and C_{i+1}. Using a truth table or otherwise, derive expressions for S and C_{i+1} in terms of D and C_i.
[4]

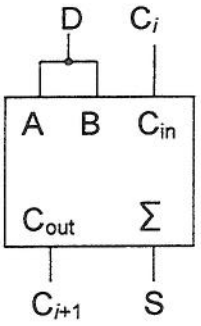
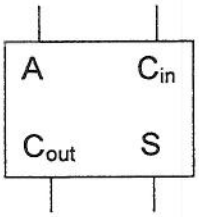


Figure 2.2

c) Draw a circuit diagram to show how the 9x circuit can be implemented using 2 full-adders and 1 half-adder by first sign-extending A to 8 bits. You will need to use the result of part b) for this. Ensure all signals are labelled. (Marks will be deducted for a design using more than 2 full-adders.)
[13]

d) The 9x circuit can also be implemented using half-decrementor circuits. A chain of half-decrementors subtracts 1 from a number. Figure 2.3 shows a half-decrementor circuit and its truth table. Draw circuit diagrams showing how the half-decrementor can be implemented using only NOR gates and inverters. Do not use more gates than necessary.
[8]



A	C_{in}	S	C_{out}
0	0	1	0
0	1	0	1
1	0	0	1
1	1	1	1

Figure 2.3

3. The Moore Finite State Machine (FSM) shown in Figure 3.1 has the following state transition table:

Current state	Input A	Next state	Output Z
S0	0	S0	0
S0	1	S1	0
S1	0	S2	0
S1	1	S1	0
S2	X	S2	1
S3	X	S3	X

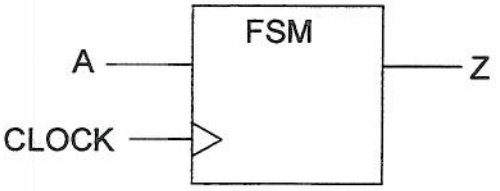


Figure 3.1

The FSM is to be implemented using D flip-flops to encode the state. The state variables are Q1 and Q0, for which the values 00, 01, 10, 11 are assigned for states S0, S1, S2, S3 respectively.

a) Draw the state diagram of the FSM.

[9]

b) Determine the logic required to implement the next state and the output. You do **not** need to draw the circuit diagram: Boolean expressions for the logic will be sufficient.

[9]

c) Determine how many AND gates, OR gates and inverters are required to implement the FSM logic.

[6]

d) Copy and complete the waveforms in Figure 3.2, assuming the FSM is initially in state S0.

[6]

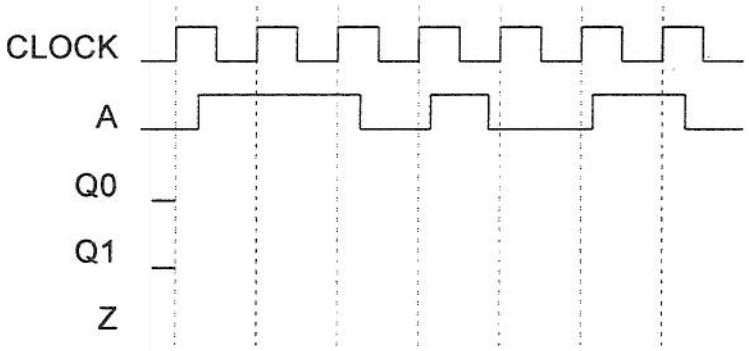


Figure 3.2

4. Figure 4.1 shows a 4-input multiplexer connected to implement a function F of the three Boolean variables: A, B, C.

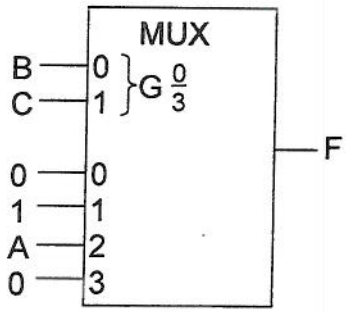


Figure 4.1

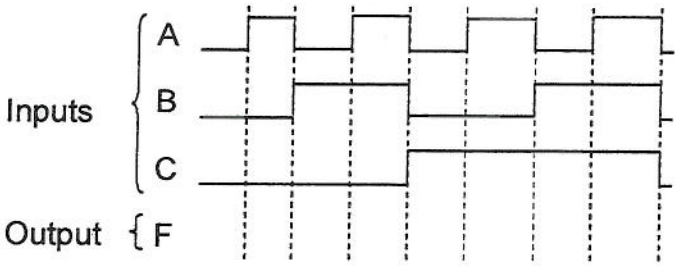


Figure 4.2

a) Copy and complete the waveform of Figure 4.2, showing the value for F given the values of A, B, and C.

[4]

b) The symbol for a 2-input multiplexer is shown in Figure 4.3. The operation of a 2-input multiplexer is defined by $Z = P\overline{S} + QS$.

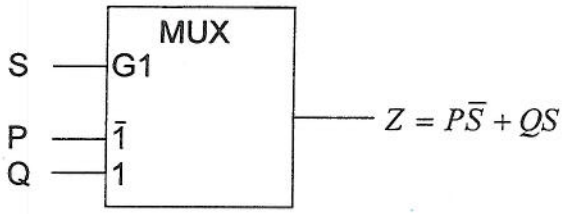


Figure 4.3

Show how the 2-input multiplexer and an inverter can be used to implement an XOR function on two Boolean variables A, B.

[4]

c) The Hamming code was invented by Richard Hamming in 1950 for error checking and correction in computer systems. For each four bits of data, A, B, C, D, three parity bits P1, P2, P3, are generated which are defined by:

$$P1 = A \oplus B \oplus D$$
$$P2 = A \oplus C \oplus D$$
$$P3 = B \oplus C \oplus D$$

Using the result from part b), draw a circuit diagram to show how P1, P2 and P3 can be implemented using only five 2-input multiplexers and inverters. (Marks will be deducted for a design that uses more than five multiplexers.)

[10]

Continued on the following page

Figure 4.4 gives the symbol for a 4-bit comparator circuit, which compares two 4-bit unsigned numbers and has three outputs. L is only high when A[3:0] is less than B[3:0], E is high when A[3:0] and B[3:0] are equal, and G is high when A[3:0] is greater than B[3:0]. Show how two of these comparator circuits can be connected with additional logic to perform the same three comparisons on two 8-bit unsigned numbers. Make sure all inputs and outputs are labelled.

[12]

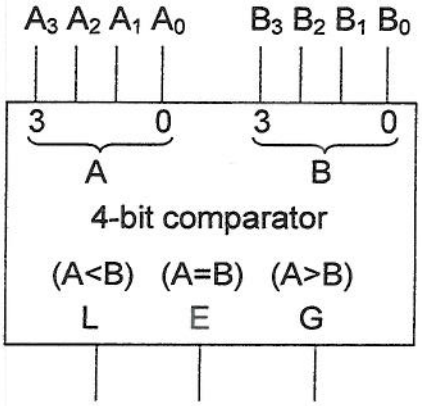


Figure 4.4

[THE END]

master - May 09

Digital Electronics I

Solutions 2009

[Question 1 is compulsory]

Question 1 tests the student's basic knowledge of the whole course. It requires understanding and the ability to analyse, but no design. All parts to this question are essentially straight from course notes and coursework.

1. a) Simplify the following two Boolean expressions using Boolean algebra:

i) $\overline{(A+B)}(\overline{A}+C)$

$$\begin{aligned}\overline{(A+B)}(\overline{A}+C) &= \overline{A}\,\overline{B}(\overline{A}+C)\\ &= \overline{A}\,\overline{B}\overline{A} + \overline{A}\,\overline{B}C\\ &= \overline{A}\,\overline{B} + \overline{A}\,\overline{B}C\\ &= \overline{A}\,\overline{B}(1+C)\\ &= \overline{A}\,\overline{B}\end{aligned}$$

1 mark for using DeMorgan's theorem correctly.
1 mark for a correct expansion or simplification.
1 mark for correct answer.

ii) $A\overline{B}\,\overline{C} + \overline{B}C + \overline{A}BC$

$$\begin{aligned}A\overline{B}\,\overline{C} + \overline{B}C + \overline{A}BC &= A\overline{B}\,\overline{C} + (A+\overline{A})\overline{B}C + \overline{A}BC\\ &= A\overline{B}\,\overline{C} + A\overline{B}C + \overline{A}\,\overline{B}C + \overline{A}BC\\ &= A\overline{B}(\overline{C}+C) + \overline{A}C(\overline{B}+B)\\ &= A\overline{B} + \overline{A}C\end{aligned}$$

1 mark for the expansion
1 mark for a correct simplification step.
1 mark for the right answer.

[6 marks]

b) Convert:

i) the decimal number 70 into unsigned 8-bit binary and BCD forms

70 decimal = 64+4+2 = $2^6+2^2+2^1$ = $(01000110)_2$
Can also use repeated division by 2:
70/2 = 35 rem 0
35/2 = 17 rem 1
17/2 = 8 rem 1
…
(2 marks)

70 decimal = 0111 0000 in BCD
(1 mark – note that 111 0000 is not correct and should get ½ mark)

ii) the 2's complement binary number 1010000 into decimal and 12-bit binary forms.

$1010000 = -2^6 + 2^4 = -64 + 16 = -48$ in decimal
(2 marks)

$1010000 = 111111010000$
(1 mark)

Marks can be given for an incorrect answer where working is shown and the working demonstrates the student has knowledge of what to do.

[6 marks]

c) For the circuit shown in Figure 1.1, write out the truth table. Hence, or otherwise, write the canonical sum-of-products expression for Z.

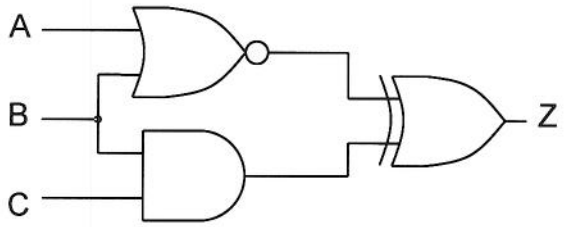


Figure 1.1

A	B	C	A NOR B	B AND C	Z
0	0	0	1	0	1
0	0	1	1	0	1
0	1	0	0	0	0
0	1	1	0	1	1
1	0	0	0	0	0
1	0	1	0	0	0
1	1	0	0	0	0
1	1	1	0	1	1

$$Z = \overline{A}\,\overline{B}\,\overline{C} + \overline{A}\,\overline{B}C + \overline{A}BC + ABC$$

2 marks for a truth table with A, B, C as inputs, Z as an output and 8 rows.
2 marks for getting the values for the Z column correct, otherwise 1 mark if an intermediate column (A NOR B, B AND C) is shown and is correct.
2 marks for writing down the canonical form from the truth table, otherwise 1 mark if the expression is canonical SOP form but wrong.

[6 marks]

d) For the circuit shown in Figure 1.2(a), complete the waveforms for Y and Z given the waveforms of the inputs A and B in Figure 1.2(b).

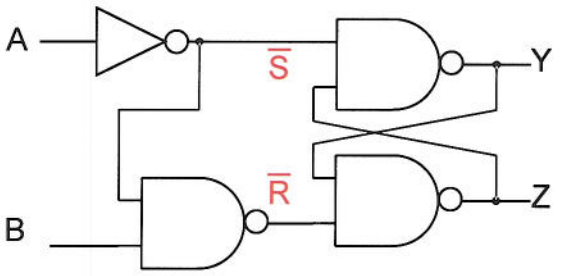


Figure 1.2(a)

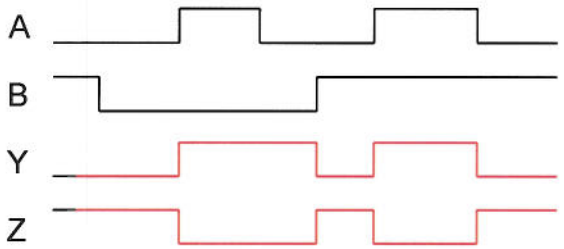


Figure 1.2(b)

The idea here is to understand that the circuit contains a SR latch. We can label the latch inputs with SET and RESET as shown. Then the latch will be SET when A is HIGH, and RESET when A is LOW and B is HIGH.

Full marks (6) if the waveforms are correct. Otherwise award partial marks at your discretion. Look for:
Z = NOT Y
Realising it is a latch
Realising A can set the latch when HIGH.
Realising B can reset the latch when HIGH.
Realising A overrides B.

[6 marks]

e) The truth table below defines the functions F and G in terms of Boolean variables A, B and C. Using Karnaugh maps, derive minimal sum-of-products expressions for F and G

A	B	C	F	G
0	0	0	0	1
0	0	1	0	1
0	1	0	1	0
0	1	1	1	0
1	0	0	0	X
1	0	1	0	0
1	1	0	0	1
1	1	1	1	0

For F:

A\BC	00	01	11	10
0	0	0	1	1
1	0	0	1	0

$F = \overline{A}B + BC$

For G:

A\BC	00	01	11	10
0	1	1	0	0
1	X	0	0	1

$G = \overline{A}\overline{B} + A\overline{C}$

1 mark for having at least one K-map table drawn out with 3 variables and columns and rows labelled correctly.

2 marks for filling in both the K-maps correctly.
2 marks for grouping terms.
1 mark for getting the expressions for F and G both correct.

[6 marks]

f) For the sequential circuit shown in Figure 1.3, draw the state diagram.

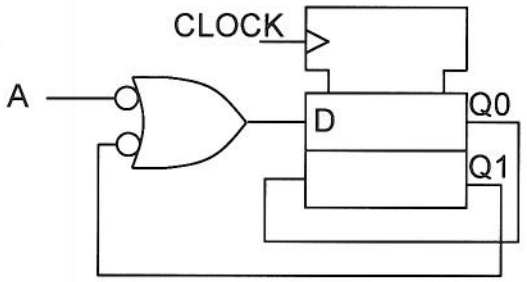
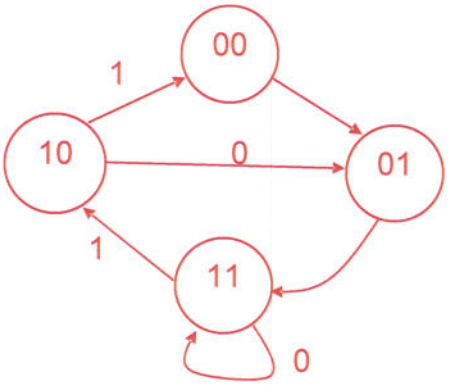


Figure 1.3

Note that Q1+ = Q0. Also, if Q1 = 0 then Q0+ = 1. If Q1 = 1, then Q0+ = NOT A.



1 mark for having 4 states.
1 mark for labelling the states with 00, 01, 10, 11.
1 mark for showing arcs between the states, labelled with 0, 1 (or A, NOT A)
3 marks if the arcs and labels are correct.

[6 marks]

g) A ROM has 5 address inputs and 4 data outputs. What is the capacity of the ROM in bits?

Capacity = 2^5 x 4 = 128 bits

1 mark for the correct form of the equation 2^a x b
1 mark for putting 5 and 4 in the right places
2 marks for the correct value of 128

[4 marks]

Question 2 examines binary arithmetic circuits and knowledge of binary numbers, in particular two's complement, full-adders and half-adders. Some circuit design is required.

2. A binary number can be multiplied by a constant amount by shifts and additions. For example, multiplying a number A by 5 can be achieved by adding A and 4A. The 9x circuit in Figure 2.1 multiplies a 4-bit signed (two's complement) number A[3:0] by 9 to produce an N-bit number P.

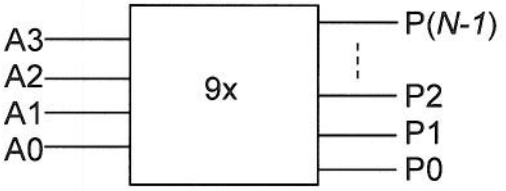


Figure 2.1

a) Show that the minimum value of N, the number of bits required to represent P, must be 8.

The range of A is -8 to +7. Multiplying by 9 gives a range of -72 to 63.
63 can be represented in 2's complement binary using 7 bits (0111111).
However, -72 requires at least 8 bits (10111000).
Therefore N = 8.

1 mark for determining the range of A.
2 marks for the range of P.
2 marks for determining how many bits are required for the range of P.

[5 marks]

b) Figure 2.2 shows a full adder which has been connected to signals D and C_i and produces output signals S and C_{i+1}. Using a truth table or otherwise, derive expressions for S and C_{i+1} in terms of D and C_i.

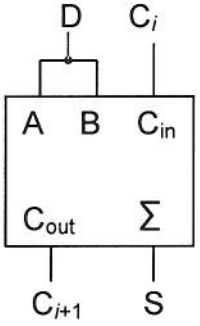


Figure 2.2

The truth table:

D	C_i	S	C_{i+1}
0	0	0	0
0	1	1	0
1	0	0	1
1	1	1	1

$S = C_i$
$C_{i+1} = D$

1 mark for drawing a truth table with the right number of rows and columns correctly labelled.
2 marks for the right values in the table
1 mark for the Boolean expressions

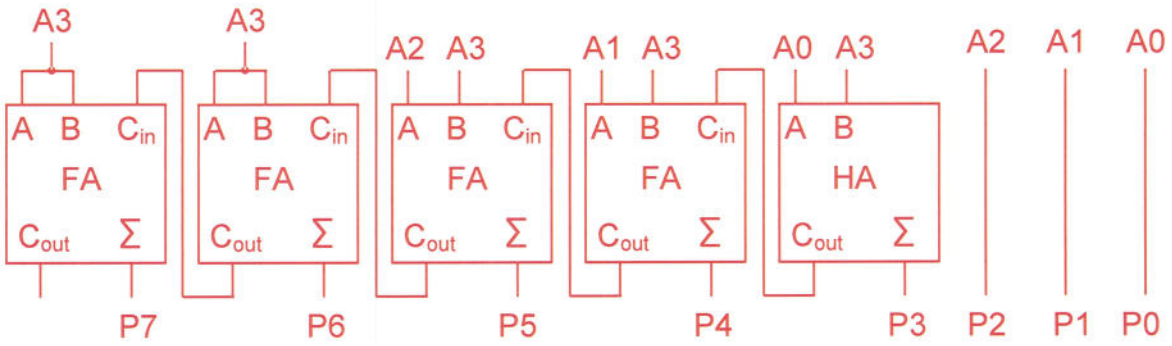
[4 marks]

c) Draw a circuit diagram to show how the 9x circuit can be implemented using 2 full-adders and 1 half-adder by first sign-extending A to 8 bits. You will need to use the result of part b) for this. Ensure all signals are labelled.

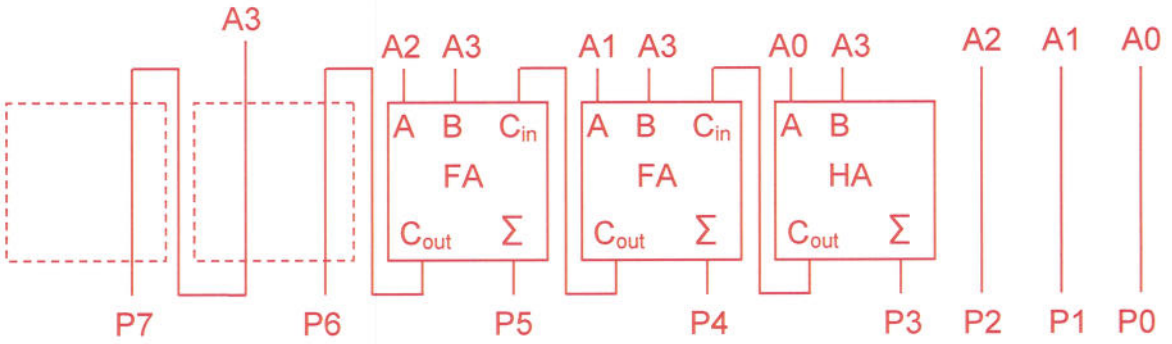
One has to realise that A x9 is the same as adding A x8 and A (1 mark). Also, A x8 is just A shifted to the left by 3 places (1 mark). The student is told to sign extend to 8 bits, so the two numbers to add are:
A3 A3 A3 A3 A3 A2 A1 A0
A3 A3 A2 A1 A0 0 0 0

2 marks for figuring this out.

The lowest three bits of P, P[2:0], are just A[2:0]. (1 mark)
P3 is A3 + A0 (addition), which requires a half-adder. (1 mark)
P[7:4] appears to require full adders. (1 mark), to give:



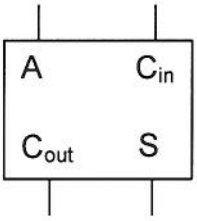
However, if we notice that the two left full adders have their inputs connected together, we can use the result from part b to simplify these.



1 mark for connecting the adders correctly, 1 mark for labelling signals, 2 marks for realising that the left two adders can be removed, 2 marks for a fully correct solution.

[13 marks]

d) The 9x circuit can also be implemented using half-decrementor circuits. A chain of half-decrementors subtracts 1 from a number. Figure 2.3 shows a half-decrementor circuit and its truth table. Draw circuit diagrams showing how the half-decrementor can be implemented using only NOR gates and inverters. Do not use more gates than necessary.



A	C_{in}	S	C_{out}
0	0	1	0
0	1	0	1
1	0	0	1
1	1	1	1

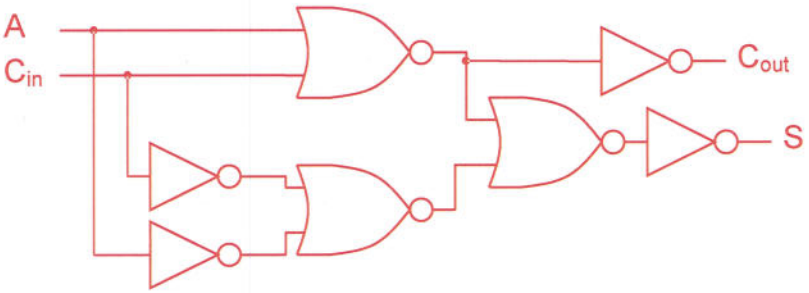
Figure 2.3

The first thing is to write equations for S and C_{out} – these can then be converted into NOR form in equations, or from the circuit diagram.

$$S = \overline{A}\,\overline{C_{in}} + AC_{in} = \overline{\overline{A + C_{in}} + \overline{\overline{A} + \overline{C_{in}}}}$$

$$C_{out} = A + C_{in} = \overline{\overline{A + C_{in}}}$$

2 marks for the original equations, 2 marks for writing them in NOR form.



2 marks for having circuit diagrams correct, 2 marks for reusing the NOR gate.

[8 marks]

Question 3 examines sequential logic and Finite State Machine designs. Circuit design is required, but the design does not require creativity, just "cranking the handle" to get the solution.

3. The Moore Finite State Machine (FSM) shown in Figure 3.1 has the following state transition table:

Current state	Input A	Next state	Output Z
S0	0	S0	0
S0	1	S1	0
S1	0	S2	0
S1	1	S1	0
S2	X	S2	1
S3	X	S3	X

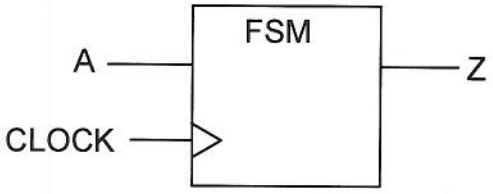
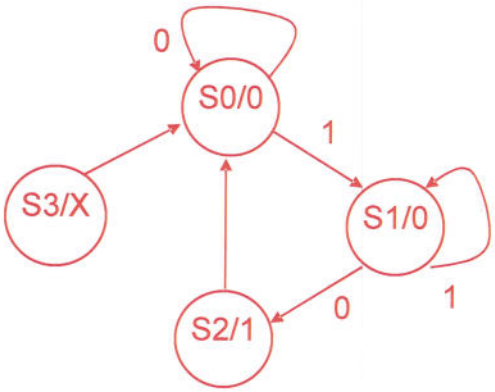


Figure 3.1

The FSM is to be implemented using D flip-flops to encode the state. The state variables are Q1 and Q0, for which the values 00, 01, 10, 11 are assigned for states S0, S1, S2, S3 respectively.

a) Draw the state diagram of the FSM.



2 marks for drawing four states labelled S0 to S3 (or 00, 01, etc).
2 marks for drawing the arcs correctly.
2 marks for labelling the arcs with the correct input value required (0 or 1) – these can be labelled with A and NOT A instead of 1 and 0.
1 mark for showing output labels (on arcs or in states).
2 marks for having the output values correct, whether in the states or on the arcs.

[9 marks]

b) Determine the logic required to implement the next state and the output. You do **not** need to draw the circuit diagram: Boolean expressions for the logic will be sufficient.

Firstly, the assigned-state table is:

Q1 Q0	Input A	Q1+ Q0+	Output Z
0 0	0	0 0	0
0 0	1	0 1	0
0 1	0	1 0	0
0 1	1	0 1	0
1 0	X	0 0	1
1 1	X	0 0	X

For Q0+

A\Q1Q0	00	01	11	10
0	0	0	0	0
1	1	1	0	0

$$Q0+ = A\overline{Q1}$$

3 marks for correct equation

For Q1+

A\Q1Q0	00	01	11	10
0	0	1	0	0
1	0	0	0	0

$$Q1+ = \overline{A}\,\overline{Q1}Q0$$

3 marks for correct equation

For Z, this will be 1 when Q1 is 1, so no additional logic is required.

3 marks for Z=Q1.

[9 marks]

c) Determine how many AND gates, OR gates and inverters are required to implement the FSM logic.

For Q0+ we need 1 ~~OR~~ AND gate and 1 inverter.
(2 marks)

For Q1+ we need 1 ~~OR~~ AND gate and 2 inverters.
(2 marks)

However, we can reuse one of the inverters. Therefore we need:
2 ~~OR~~ AND gates and 2 inverters in total.
(2 marks)

[6 marks]

d) Copy and complete the waveforms in Figure 3.2, assuming the FSM is initially in state S0.

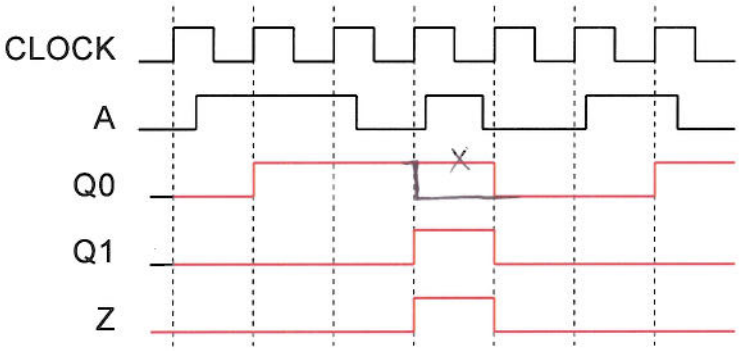


Figure 3.2

1 mark for showing all signals changing only on positive clock edges.
4 marks for the correct waveforms for Q0 and Q1
1 mark if Z=Q1.

[6 marks]

Question 4 is about using more complex blocks of circuits to build logic functions. It requires some inventiveness. Muxes and comparators are covered in the course notes, but the application of them in new circuits is tested in this question.

4. Figure 4.1 shows a 4-input multiplexer connected to implement a function F of the three Boolean variables: A, B, C.

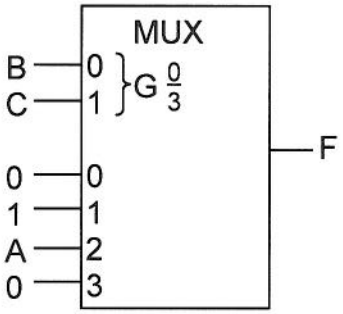


Figure 4.1

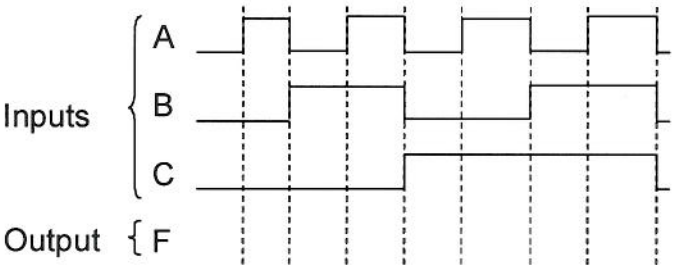
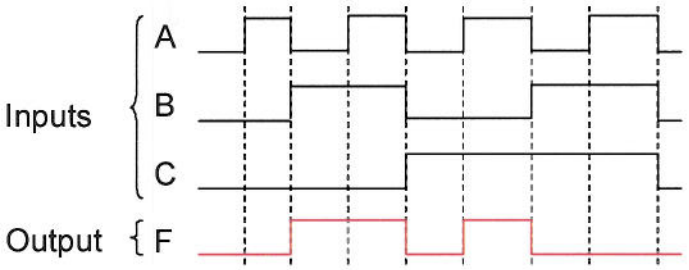


Figure 4.2

a) Copy and complete the waveform of Figure 4.2, showing the value for F given the values of A, B, and C.



½ mark for each division of the waveform correct

[4 marks]

b) The symbol for a 2-input multiplexer is shown in Figure 4.3. The operation of a 2-input multiplexer is defined by $Z = P\overline{S} + QS$.

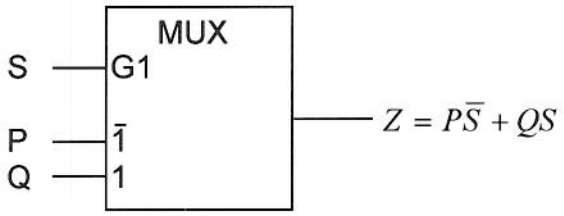
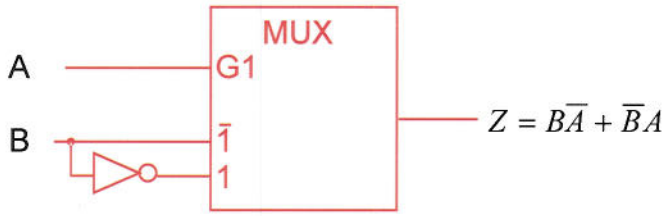


Figure 4.3

Show how the 2-input multiplexer and an inverter can be used to implement an XOR function on two Boolean variables A, B.

A mux and inverter can make an XOR function if Q = NOT P.



[4 marks.]

c) The Hamming code was invented by Richard Hamming in 1950 for error checking and correction in computer systems. For each four bits of data, A, B, C, D, three parity bits P1, P2, P3, are generated which are defined by:

$$P1 = A \oplus B \oplus D$$
$$P2 = A \oplus C \oplus D$$
$$P3 = B \oplus C \oplus D$$

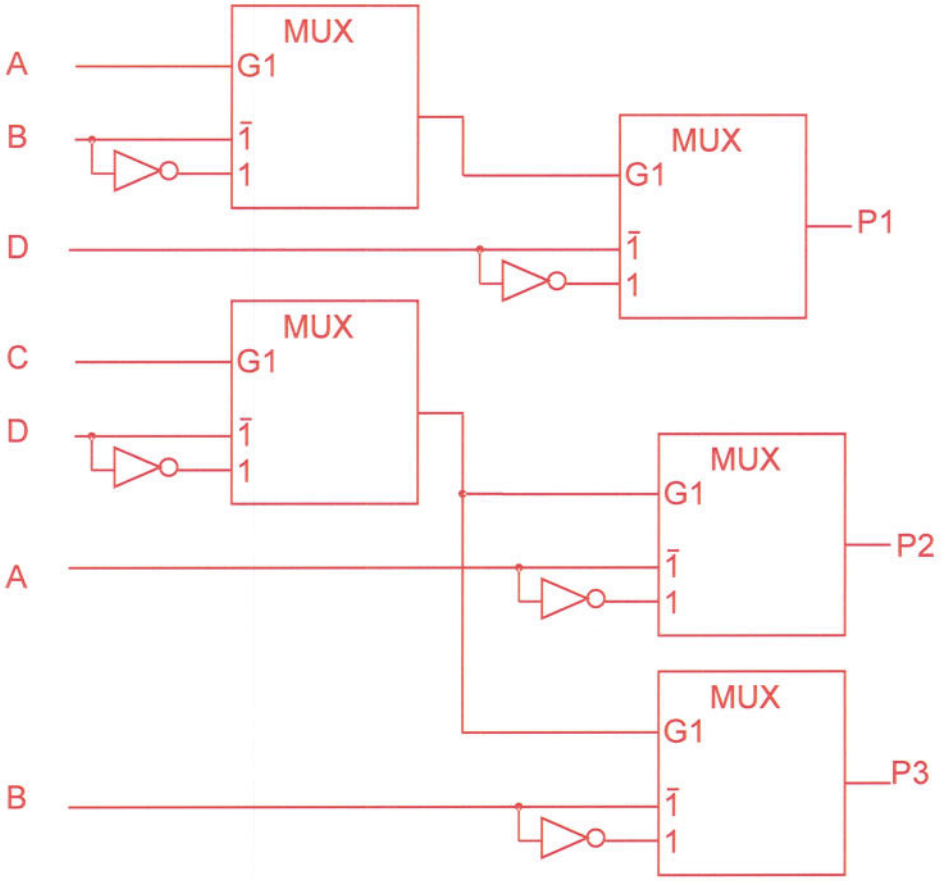
Using the result from part b), draw a circuit diagram to show how P1, P2 and P3 can be implemented using only five 2-input multiplexers and inverters. (Marks will be deducted for a design that uses more than five multiplexers.)

The Hamming code is not taught, so this is a new application of theory.

To implement P1 P2 and P3 directly needs 6 XOR gates. However, both P2 and P3 have a the term C XOR D. Therefore, let's use an intermediate variable $Q = C \oplus D$ so that:

$$P1 = A \oplus B \oplus D$$
$$P2 = A \oplus Q$$
$$P3 = B \oplus Q$$

The circuit is therefore:



10 marks for drawing this correctly, deduct 3 marks if an extra MUX is used.

c) Figure 4.4 gives the symbol for a 4-bit comparator circuit, which compares two 4-bit unsigned numbers and has three outputs. L is only high when A[3:0] is less than B[3:0], E is high when A[3:0] and B[3:0] are equal, and G is high when A[3:0] is greater than B[3:0]. Show how two of these comparator circuits can be connected with additional logic to perform the same three comparisons on two 8-bit unsigned numbers. Make sure all inputs and outputs are labelled.

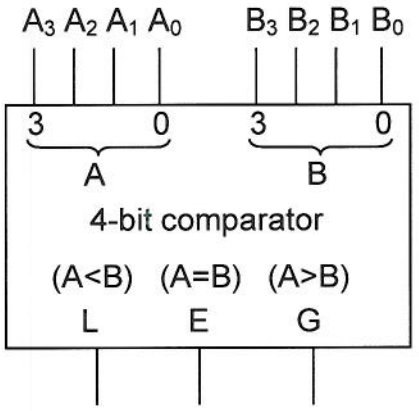


Figure 4.4

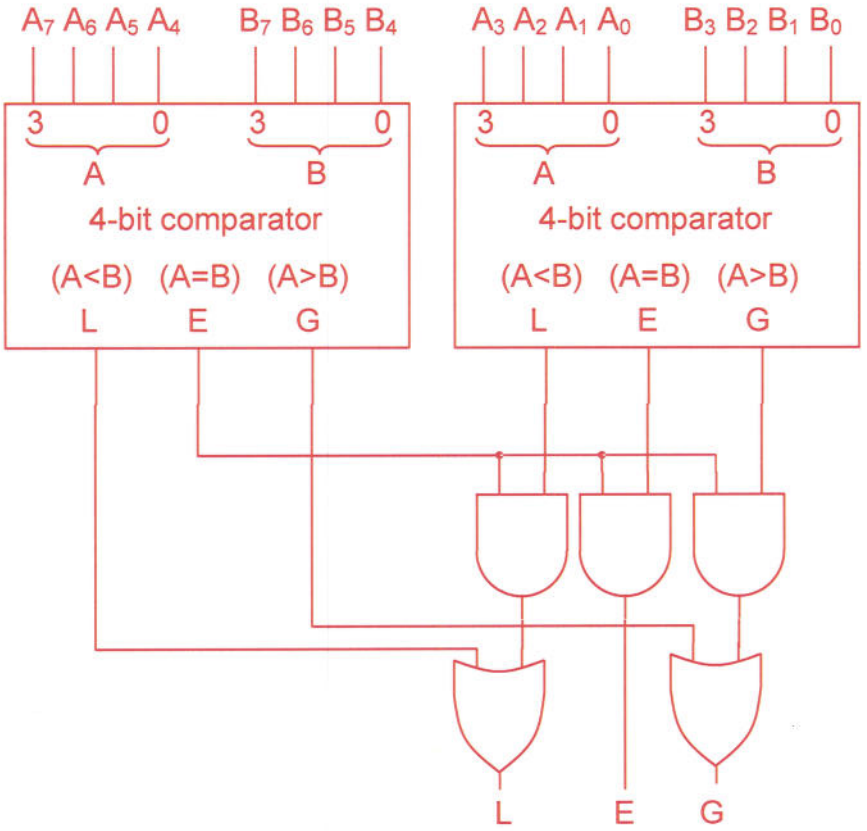
We can compare two 8 bit numbers by comparing the upper 4 bits and the lower 4 bits separately.

2 marks for figuring this out.

The following logic will produce L, E and G:

L is 1 if A[7:4] < B[7:4] OR (A[7:4] = B[7:4] AND A[3:0] < B[3:0])
G is 1 if A[7:4] > B[7:4] OR (A[7:4] = B[7:4] AND A[3:0] > B[3:0])
E is 1 if A[7:4] = B[7:4] and A[3:0] = B[3:0]

5 marks for determining these functions.



5 marks for the correct circuit diagram

[THE END]