IMPERIAL COLLEGE LONDON

DEPARTMENT OF ELECTRICAL AND ELECTRONIC ENGINEERING
EXAMINATIONS 2012

EEE/ISE PART I: MEng, BEng and ACGI

DIGITAL ELECTRONICS 1

Wednesday, 30 May 10:00 am

Time allowed: 2:00 hours

There are THREE questions on this paper.

Answer ALL questions.
Q1 carries 40% of the marks. Questions 2 and 3 each carry 30%.

Any special instructions for invigilators and information for candidates are on page 2.

Examiners responsible First Marker(s) : Z. Durrani
Second Marker(s) : J.V. Pitt

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

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

i) $\overline{A}C + \overline{A}\overline{C}D + ABC + AB\overline{C}D$

ii) $\overline{(A \oplus B) + \overline{(A\overline{B})}\overline{C}}$ [4]

[4]

b) Simplify the following Boolean equation, in product-of-sums form, using a Karnaugh map.

$f(A,B,C,D) = \Sigma(4,5,6,12,14,15)$, with a 'don't care' at 10

[4]

c) Simplify the following Boolean equation, in sum-of-products form, using a Karnaugh map.

$$f = \overline{A}\overline{B}C + ABC + AB\overline{C}\overline{D} + A\overline{B}C$$

[4]

d) Assuming that all numbers are 16 bits wide, 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 solution is derived.)

[8]

Decimal	Hexadecimal	Binary (signed)	Octal
7955	?		
		?	1336
	0D71		?
?		1000 0001 0000 0001	

e) An 8 × 2 ROM is to be used to implement simultaneously the following Boolean functions:

$$f(A,B,C) = A\overline{C} + \overline{A}\,\overline{B}C$$
$$g(A,B,C) = B\overline{C} + A\overline{B}\,\overline{C}$$

Complete the missing stored data values (in hexadecimal) in the table shown in Figure 1.1, to implement functions f and g. Also state how you would connect the logic variables, and functions f and g, to the ROM.

[6]

Address X[2:0]	Data Y[1:0]
0	0
1	2
2	1
3	-
4	3
5	-
6	-
7	-

Figure 1.1

f) Sketch the Moore state diagram for the circuit shown in Figure 1.2, where A is the input variable. You may assume that initially, Q[2:0] = 000.

[6]

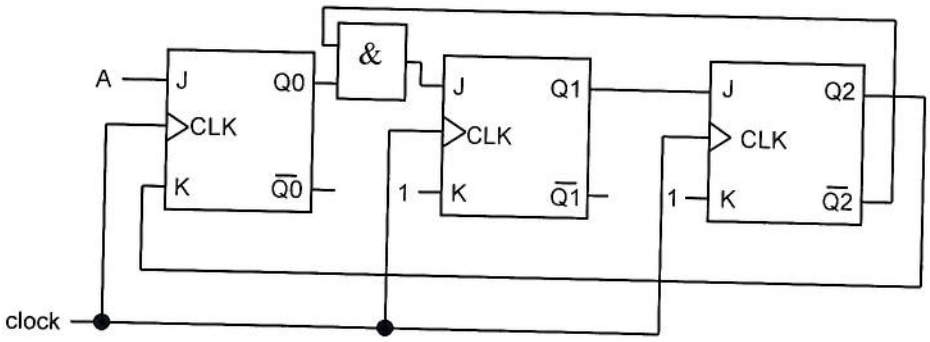


Figure 1.2

g) Determine the number of bits needed to represent the range of results obtained when two N bit signed numbers are multiplied together.

[4]

Question 2

2. a) Given that the variables A[2:0] form a three bit grey code to represent numerical values:

i) Write down all binary combinations for A[2:0], in ascending order of numerical value.

[4]

ii) A three bit grey code-to-decimal decoder is to be designed, where the outputs X[7:0] are active low. Write down the Boolean functions for each output terminal.

[6]

b) Consider the Boolean function:

$$f = \overline{A}\,\overline{B}\,\overline{C} + AB + AC$$

i) Implement this function using one 4×1 multiplexer, with the restriction that the variable C cannot be connected to a multiplexer select line.

[4]

ii) Implement this function using two 2×1 multiplexers.

[8]

c) Figure 2.1 shows the IEEE/ANSI symbol for a 74x163 counter integrated circuit. The input signals CLOCK, MR and ML vary as shown in Figure 2.2.

i) By inspecting symbols and dependencies within Figure 2.1, state the action of the terminals MR and ML on the counter.

[4]

ii) Hence complete the timing diagram of Figure 2.2, for the output signals Z[3:0]. Here, the initial value of Z3 = Z2 = Z1 = Z0 = 1.

[4]

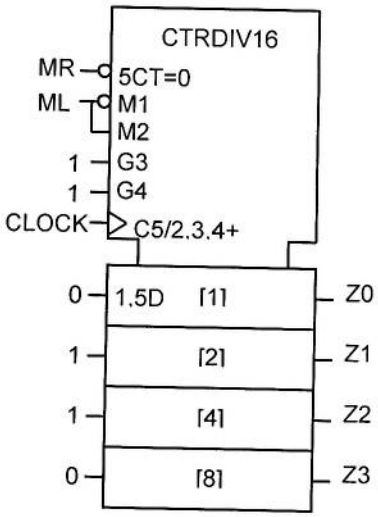


Figure 2.1

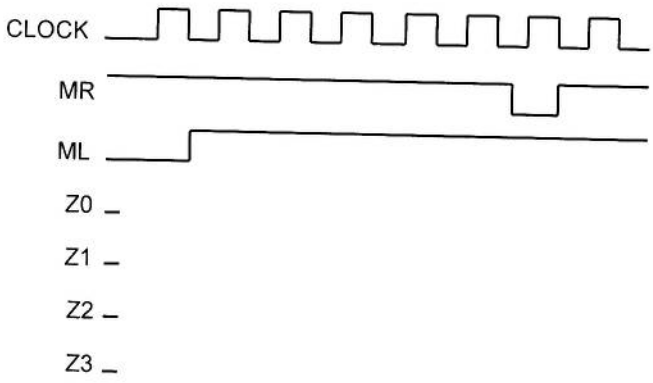


Figure 2.2

Question 3

3. Figure 3.1 shows a Moore finite state machine (FSM) with two inputs X0, X1 and one output Z. The FSM detects the following input sequences:

   i) Input sequence X0 X1 = 01, followed by 10, causes the output to become 1.
   ii) Input sequence X0 X1 = 11, followed by 10, causes the output to become 1.

   After detecting one of these input sequences, the FSM resets the output to 0 after one clock cycle. For all other input transitions, the FSM remains reset.

   a) Given that the FSM is implemented with three states (S0, S1, S2), draw the state diagram for the FSM.

   [10]

   b) Given that the three states are encoded using the following state assignment, derive the state transition table for the FSM.

State	Q2	Q1	Q0
S0	0	0	1
S1	0	1	0
S2	1	0	0

   [8]

   c) The FSM is to be implemented using D-type flip-flops and a combinational logic circuit. Determine the Boolean expressions necessary for this implementation in sum-of-products form.

   [8]

   d) Sketch the circuit diagram for your design. This should show the D-type flip-flops, the gate-level circuit diagram for the combinational logic circuit, the input/output variables for the combinational logic circuit, and the output variables of the D-type flip-flops.

   [4]

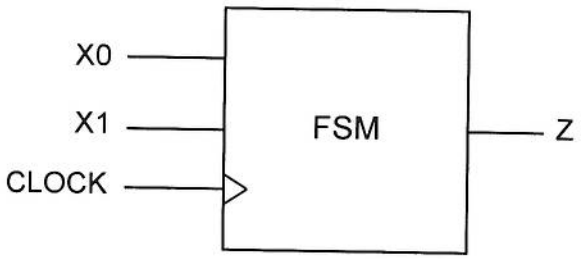


Figure 3.1

[THE END]

E1.2 Digital Electronics 1
Solutions 2012

Answer to Question 1:

a) (i)

$$\begin{aligned}
&\overline{A}C + \overline{A}\,\overline{C}D + ABC + AB\overline{C}D \\
&= \left(\overline{A} + AB\right)C + \left(\overline{A} + AB\right)\overline{C}D \\
&= \left(\overline{A} + B\right)\left(C + \overline{C}D\right) \\
&= \left(\overline{A} + B\right)\left(C + D\right) \\
&= \overline{A}C + BC + \overline{A}D + BD
\end{aligned}$$

Straightforward application of boolean algebra, note use of rule 11, lecture 4

[4]

(ii)

$$\begin{aligned}
&\overline{(A \oplus B) + \overline{\left(A\overline{B}\right)}\,\overline{C}} \\
&= \overline{\left(A \oplus B\right)}\,\overline{\overline{\left(A\overline{B}\right)}\,\overline{C}} \\
&= \left(\overline{A}\,\overline{B} + AB\right)\left(A\overline{B} + C\right) \\
&= \left(\overline{A}\,\overline{B} + AB\right)C \\
&= \overline{A}\,\overline{B}C + ABC
\end{aligned}$$

Successive use of De Morgan's theorem

[4]

b)

$f(A,B,C,D) = \Sigma(4,5,6,12,14,15)$, with a 'don't care' at 10

The k-map is as follows, where we use X = 0 in the simplification.

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

Group for zeros. When writing POS expression, note inversion of the 'common' variables

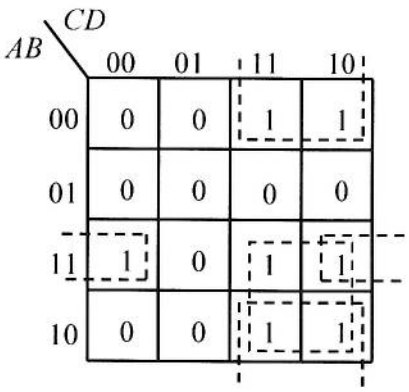
[3]

$$\Rightarrow f = B\left(A + \overline{C} + \overline{D}\right)\left(\overline{A} + C + \overline{D}\right)$$

[1]

c)

$$f = \overline{A}\,\overline{B}C + ABC + AB\overline{C}\,\overline{D} + A\overline{B}C$$



Note that the 'quad's' overlap

[3]

$$\Rightarrow f = \overline{B}C + AC + AB\overline{D}$$

[1]

d)

Decimal	Hexadecimal	Binary (signed)	Octal
7955	1F13		
		0000 0010 1101 1110	1336
	0D71		6561
-32511		1000 0001 0000 0001	

[2+2+2+2]

For the hex to octal conversion, write as binary, then regroup in three bit groups and write the corresponding octal

e)

The given functions f and g are:

Quite a few candidates had difficulty in spotting the pattern in the truth table

$$f(A,B,C) = A\overline{C} + \overline{A}\overline{B}C$$
$$g(A,B,C) = B\overline{C} + A\overline{B}\overline{C}$$

The truth table corresponding to these functions is then as follows:

A	B	C	f	g
0	0	0	0	0
0	0	1	1	0
0	1	0	0	1
0	1	1	0	0
1	0	0	1	1
1	0	1	0	0
1	1	0	1	1
1	1	1	0	0

[2]

Comparison of the f and g columns with the table in Figure 1.1 demonstrates that if f and g correspond to the stored data Y[1:0] in the 8 × 2 ROM, with f corresponding to Y1 and g corresponding to Y0, then the truth table matches the given values in the table in Figure 1.1.

This requires that variable A, B, C are connected to the three ROM address lines X[2:0] such that A = X2, B = X1 and C = X0. Other combinations of A, B, C on different address lines do not lead to an f, g pattern matching Figure 1.1

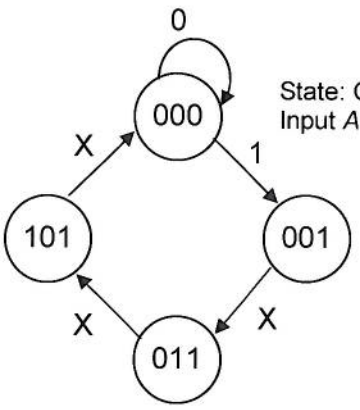
We then have the following complete data table:

Address X[2:0]	Data Y[1:0]
0	0
1	2
2	1
3	**0**
4	3
5	**0**
6	**3**
7	**0**

[4]

f)

Moore state diagram:



State: Q2Q1Q0
Input *A* values on arrows

Note the timing of the change in the states. Quite a few candidates went to the wrong state after 001. The rest of the diagram then comes out wrong

Give 2 marks for four states, 2 marks for the correct state values, and 2 marks for the correct arrows.

[6]

g) With two N bit numbers multiplied together, the result has the range:

$$-(2^{2N-2} - 2^{N-1}) \text{ to } 2^{2N-2}$$

[2]

This needs 2N bits. Note that it nearly fits into 2N-1 bits, whose range extends to $2^{2N-2} - 1$.

[2]

Study group question

Answer to Question 2:

2. a) (i) The three bit grey code is:

A2	*A1*	*A0*	*Numerical value*
0	0	0	0
0	0	1	1
0	1	1	2
0	1	0	3
1	1	0	4
1	1	1	5
1	0	1	6
1	0	0	7

Give 1 mark for the correct number of combinations (i.e. 8 combinations), and 3 marks for the correct grey code.

[4]

(ii) The truth table, including active low output Boolean functions, for a three bit grey code-to-decimal decoder is as follows:

Numerical value	*A2*	*A1*	*A0*	*Boolean output function*
0	0	0	0	$A2 + A1 + A0$
1	0	0	1	$A2 + A1 + \overline{A0}$
2	0	1	1	$A2 + \overline{A1} + \overline{A0}$
3	0	1	0	$A2 + \overline{A1} + A0$
4	1	1	0	$\overline{A2} + \overline{A1} + A0$
5	1	1	1	$\overline{A2} + \overline{A1} + \overline{A0}$
6	1	0	1	$\overline{A2} + A1 + \overline{A0}$
7	1	0	0	$\overline{A2} + A1 + A0$

Give 2 marks for demonstrating that an active low decoder requires Boolean functions which output zero when active, 4 for the correct Boolean functions.

[6]

In (i), the most common error is at numerical value 4, where the wrong bit is changed. In (ii) a typical error is not to be able to form an active-low output.

b) (i) The given Boolean function is:

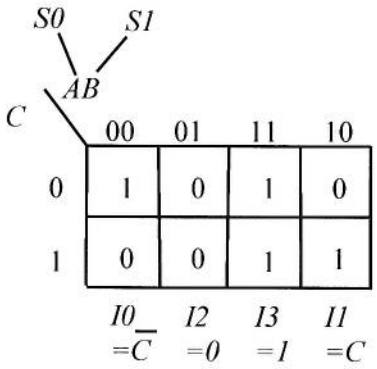
$$f = \overline{A}\,\overline{B}\,\overline{C} + AB + AC$$

This corresponds to the truth table:

A	B	C	f
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

[1]

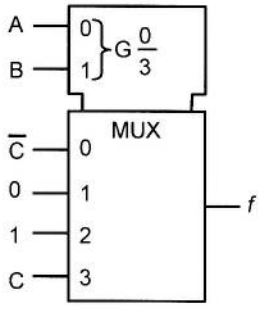
As AB must be used on the select lines of the 4 × 1 MUX, we draw a k-map table, with AB arranged along the rows and C along the columns. Furthermore, we choose to connect A = S0 and B = S1.



One can always use a MUX with one less select line than the number of boolean variables. The method is also discussed in study group questions. Alternative, correct solutions are also possible.

[2]

This gives the following implementation:



Note: The alternative solution has B = S0 and A = S1. We would then have I0 = C(bar), I1 = 0, I3 = 1, and I2 = C.

[1]

(ii) Using two 2 × 1 MUXs, we first need to rearrange the given function in the form of the Boolean function for the output Z for a 2 × 1 MUX, with select input S:

$$Z = \overline{S}I0 + SI1$$

This one is trickier. It requires you to rearrange the boolean function in the form of the 2x1 MUX output equation. Again, alternative solutions are possible.

For the given function, we then have:

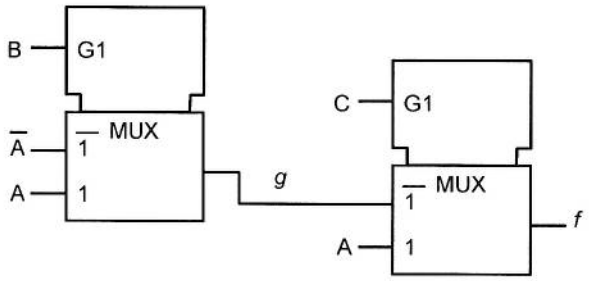
$$\begin{aligned} f &= \overline{A}\,\overline{B}\,\overline{C} + AB + AC \\ &= \left(\overline{A}\,\overline{B}\right)\overline{C} + AB\left(\overline{C} + C\right) + AC \\ &= \left(\overline{A}\,\overline{B} + AB\right)\overline{C} + ABC + AC \\ &= \left(\overline{A}\,\overline{B} + AB\right)\overline{C} + AC \end{aligned}$$

[4]

Comparison with the expression for Z implies that:

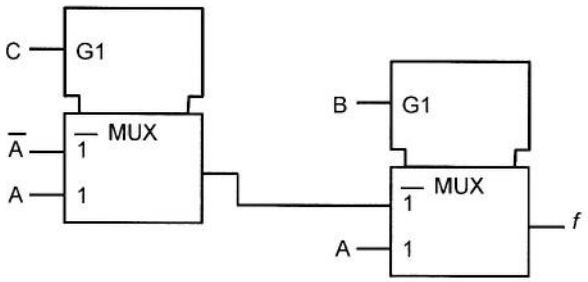
$$I0 = \left(\overline{A}\,\overline{B} + AB\right) = g$$
$$I1 = A$$
$$S = C$$

We then need to implement $\overline{A}\,\overline{B} + AB$ using a second 2 × 1 MUX. Here, either A or B could be connected to the select line. We then have the following implementation:



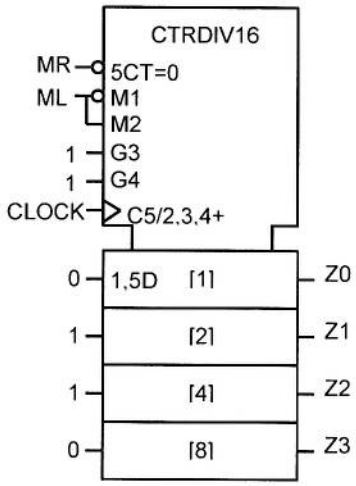
An alternative implementation uses C on the select line of the first MUX and B on the select line of the second MUX, as follows from the expression:

$$\begin{aligned} f &= \overline{A}\,\overline{B}\,\overline{C} + AB + AC \\ &= \left(\overline{A}\,\overline{C} + AC\right)\overline{B} + AB \end{aligned}$$



[4]

c)



This is a quite hard unravelling of IEEE symbols. Furthermore, the timing diagram is tricky to get if one is not completely clear about the operation of the chip.
This proved to be the most difficult question in the exam.

Figure 2.1

(i) Terminal MR: The label '5CT=0' indicates that the function 'CT=0' is performed when the control dependency associated with the label '5' (i.e. the clock on C5) is asserted. Note that this happens synchronously, on a rising edge. CT = 0 will then clear the counter to 0000. Therefore, as MR is active low:

MR = 0 ⇒ Clear the counter on the next clock rising edge.

[2]

Terminal ML: This connects directly to 'M2', and with inversion to 'M1'. The label '2' can be seen, along with G3 and G4, to control the clock terminal, and the increment (+) operation on the counter. Therefore, M2 must be asserted to allow the counter to increment with the clock edge. Alternatively, if M1 is asserted, then the input values on the D flip-flops are loaded into the counter (in the question, 0110 is loaded). We therefore have the operation:

ML = 1 ⇒ Counter increments with the clock rising edge.
ML = 0 ⇒ Counter loads input value on D flip-flops with the clock rising edge.

[2]

(ii) The timing diagram is as follows:

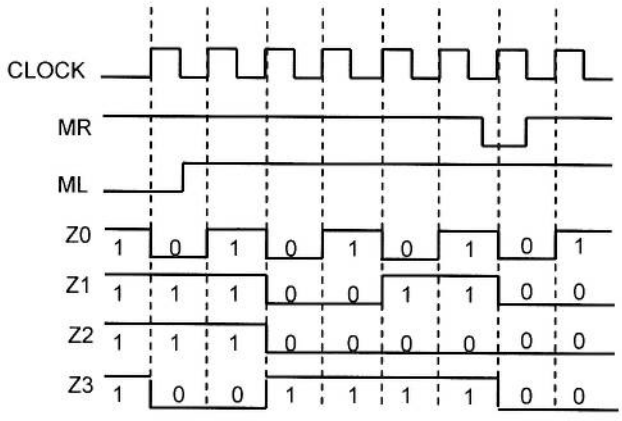


Figure 2.2

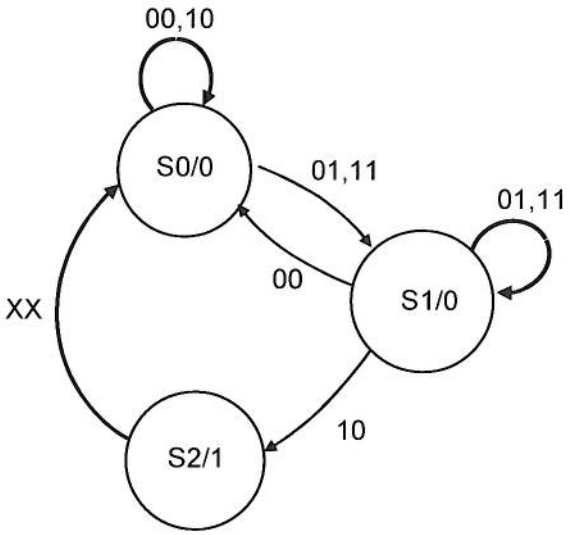
Give 1 mark per output signal Z[3:0].

[4]

Answer to Question 3

a) Moore state diagram:

This is somewhat harder than the typical FSM question. (i) There are two bits for the input, and four combinations possible for these. (ii) The 1-hot decoding means that one only needs 12 lines in the state table, not 2^5. Furthermore, (iii) a k-map is unnecessary.



Give 2 marks for correctly labelled states (state name plus output), 4 for the correct arrows, 4 for the correct input values on the arrows.

[10]

b) Here, the three states are defined using 1-hot encoding. We then have the following state transition table for the FSM:

Current state Q2Q1Q0	Input X1X0	Next state $Q2^+Q1^+Q0^+$	Output Z
001	00	001	0
001	01	010	0
001	10	001	0
001	11	010	0
010	00	001	0
010	01	010	0
010	10	100	0
010	11	010	0
100	00	001	1
100	01	001	1
100	10	001	1
100	11	001	1

Give 2 marks for the correct number of rows and columns in the table, 3 for matching the current state and input values correctly, and 3 for matching the next state and output values correctly.

[8]

c) The SOP Boolean expressions necessary for a D-type flip-flop implementation can be written directly from the '1's in the truth table. K-map simplification would be complex and is unnecessary due to the relatively simple expressions directly possible from the 1-hot encoding:

$$Q2^{+} = Q1X1\overline{X0}$$

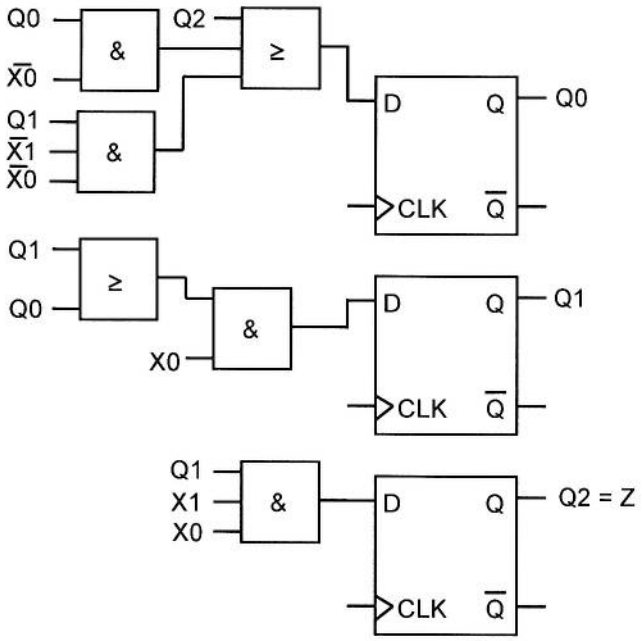
$$Q1^{+} = Q0\left(\overline{X1}X0 + X1X0\right) + Q1\left(\overline{X1}X0 + X1X0\right)$$
$$= Q0X0 + Q1X0 = \left(Q0 + Q1\right)X0$$

$$Q0^{+} = Q0\left(\overline{X1}\,\overline{X0} + X1\overline{X0}\right) + Q1\overline{X1}\,\overline{X0} + Q2$$
$$= Q0\overline{X0} + Q1\overline{X1}\,\overline{X0} + Q2$$

$$Z = Q2$$

[2+2+2+2]

d) Circuit implementation with D-type flip-flops is shown below, with input variables to the combination logic part, and output variables for the D-flip-flops shown. The interconnections between the output and input terminals, and the clock line, are not explicitly shown for clarity.



[4]