Final

E1.2

IMPERIAL COLLEGE LONDON

DEPARTMENT OF ELECTRICAL AND ELECTRONIC ENGINEERING
EXAMINATIONS 2010

EEE/ISE PART I: MEng, BEng and ACGI

**DIGITAL ELECTRONICS 1**

Thursday, 27 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) : | Z. Durrani, Z. Durrani |
| | Second Marker(s) : | J.V. Pitt, 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 is compulsory]**

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

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

[4]

ii) $\overline{(A+B)(\overline{A}+\overline{B})}$

[4]

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

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

[4]

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

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

[4]

d) The timing waveforms for signals A, B and C shown in Figure 1.1 are applied to the circuit shown in Figure 1.2. Draw the timing waveform for the output signals X and Y, given that their initial values are X = 0 and Y = 1.

[6]

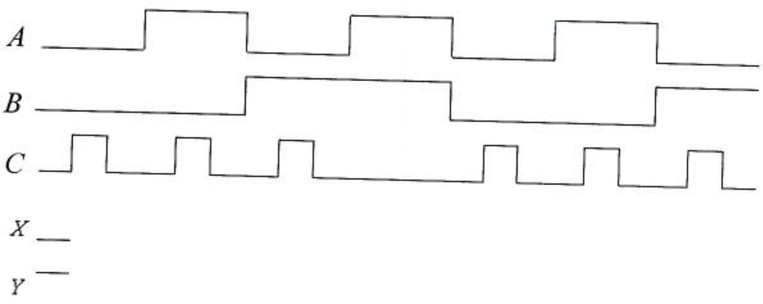


*Figure 1.1*

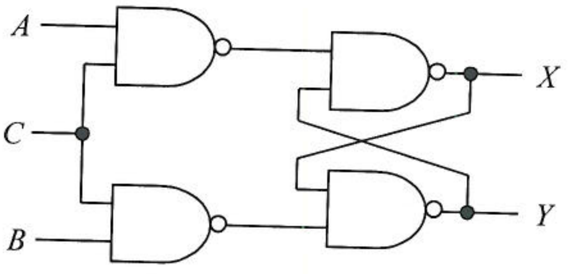


*Figure 1.2*

e) 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 | BCD |
|---|---|---|---|
| 7199 | ? | ▨ | ▨ |
| ▨ | ▨ | 1000011110010010 | ? |
| ▨ | 27ED | ? | ▨ |
| -2317 | ? | ▨ | ▨ |

f) For the circuit shown in Figure 1.3, draw a truth table showing the output Q for all combinations of inputs A, B and C.

[6]

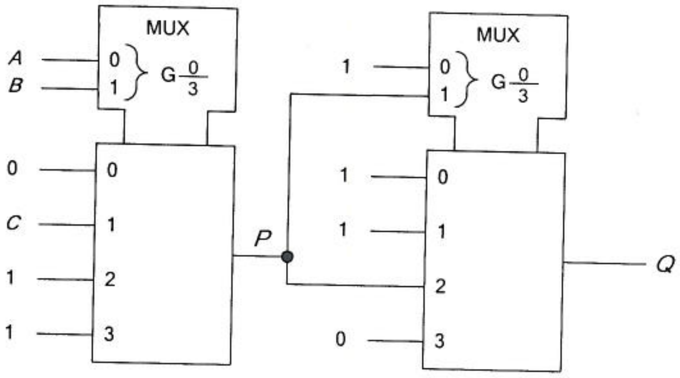


*Figure 1.3*

g) Figure 1.4 shows the Moore state diagram for a finite-state machine (FSM). Draw the state transition table corresponding to this diagram.

[4]

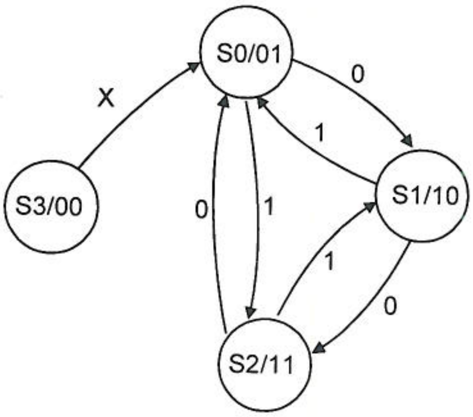


*Figure 1.4*

2. a) A 2-bit full adder circuit is to be implemented using the read-only memory (ROM) shown in Figure 2.1. The circuit should add the numbers X[1:0] and Y[1:0], and there is no input carry. The ROM address lines are A[3:0] and the ROM data lines are D[3:0]. The contents of the ROM (in hexadecimal), stored in the first four addresses, are shown in Figure 2.2. Draw the truth table showing the full contents of the ROM, expressed in binary and in hexadecimal.

[10]

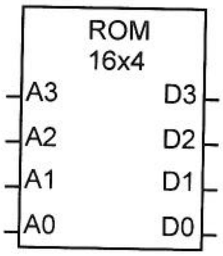


*Figure 2.1*

| Address | Data |
|---|---|
| 0 | 0 |
| 1 | 2 |
| 2 | 4 |
| 3 | 6 |
| - | - |
| - | - |

*Figure 2.2*

b) Figure 2.3 shows a finite state machine (FSM) implemented with a ROM that contains four 4-bit numbers. The ROM address lines are A[1:0] and the ROM data lines are D[3:0]. D[1:0] are connected to the inputs of two D-type flip-flops. The upper two data bits from the ROM, D[3:2], provide output signals X and Y. The outputs of the flip-flops, Q1 and Q0, are connected to the address lines A1 and A0 of the ROM respectively. The contents of the ROM are shown in Figure 2.4. The flip-flops are initially in the reset state (i.e. Q0 = Q1 = '0').

(i) Draw a Moore state diagram, and the state transition table, for the FSM.

[6]

(ii) Sketch the waveforms for the output signals X and Y for 5 cycles of the clock.

[4]

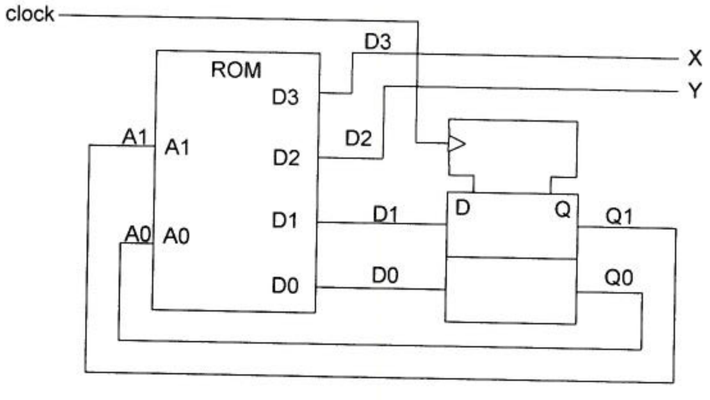


*Figure 2.3*

| Address | Data |
|---|---|
| 0 | 1101 |
| 1 | 1010 |
| 2 | 0111 |
| 3 | 0000 |

*Figure 2.4*

c) Figure 2.5 shows an arithmetic module COMP that compares the magnitude of two 4-bit unsigned numbers A[3:0] and B[3:0]. It produces a logic '1' on outputs H, E and L if A>B, A=B and A<B respectively. Derive the Boolean expressions for H, E and L. You do not need to simplify your Boolean equations.

[10]

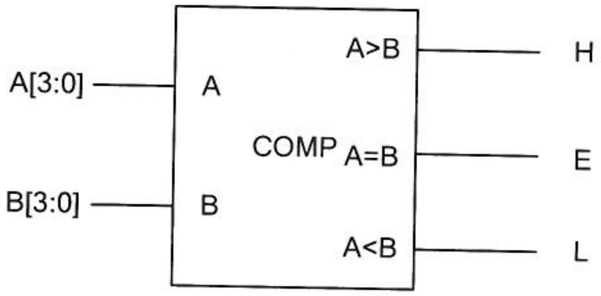


*Figure 2.5*

3. a) With the aid of a diagram, show how the following Boolean function can be implemented using a programmable array logic (PAL) device.

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

[5]

b) Sketch the gate-level circuit diagram for a 4-bit 'even-parity' generator.

[5]

c) An 8 × 1 multiplexer, shown in Figure 3.1, is to be used to implement the Boolean function Z:

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

Show how the 8 × 1 multiplexer can be used for this implementation, with the restriction that Boolean variable D cannot be used on the select lines of the multiplexer.

[10]

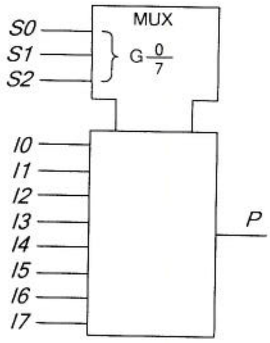


*Figure 3.1*

d) An 8 line × 3 line, octal-binary encoder is shown in Figure 3.2. The inputs A[7:0] may be activated only one at a time, with the encoder generating the corresponding binary output X[2:0].

(i) Draw the truth-table for this encoder. You should show only those input combinations that occur.

[5]

(ii) Hence, determine the Boolean expressions for X[2:0].

[5]

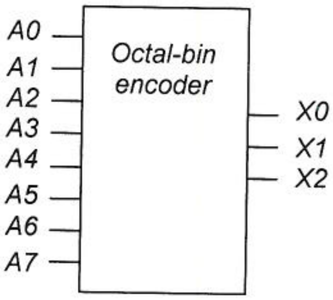


*Figure 3.2*

4. Figure 4.1 shows a sequence detector using a finite state machine (FSM), with input signal X and output signal Y. The FSM is required to detect the serial bit sequence 1101, where the MSB is to be detected first.

   a) Draw a Mealy state diagram for the FSM.

   [8]

   b) Draw the state transition table for the FSM.

   [8]

   c) The FSM is to be implemented using two J-K flip-flops and a combinational logic circuit. Derive the Boolean expressions necessary for this implementation.

   [10]

   d) Sketch the circuit diagram for your design. This should show the J-K flip-flops, the gate-level circuit diagram for the combinational logic circuit, and any interconnections.

   [4]

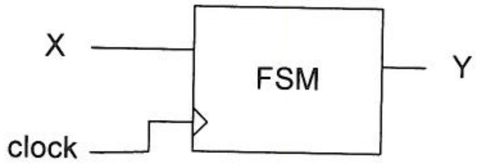


*Figure 4.1*

[THE END]

# E1.2 Digital Electronics 1
# Solutions 2010

## Answer to Question 1 (Compulsory):

a) i)

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

[4]

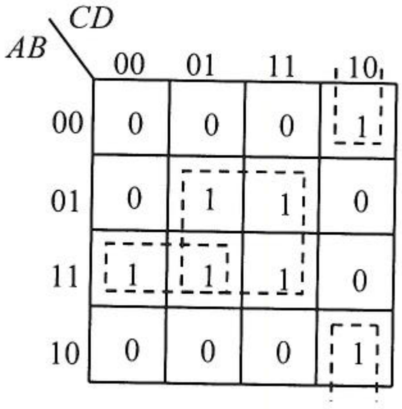
ii)

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

[4]

b)

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



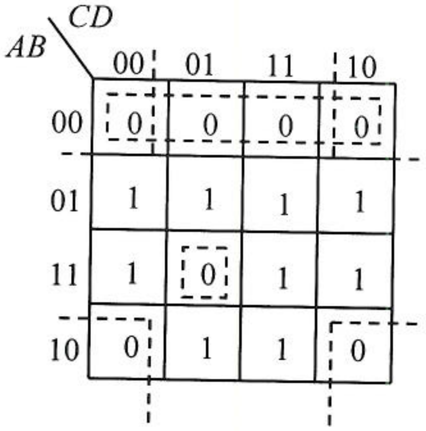
| AB \ CD | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 0 | 0 | 0 | 1 |
| 01 | 0 | 1 | 1 | 0 |
| 11 | 1 | 1 | 1 | 0 |
| 10 | 0 | 0 | 0 | 1 |

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

[4]

c)

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

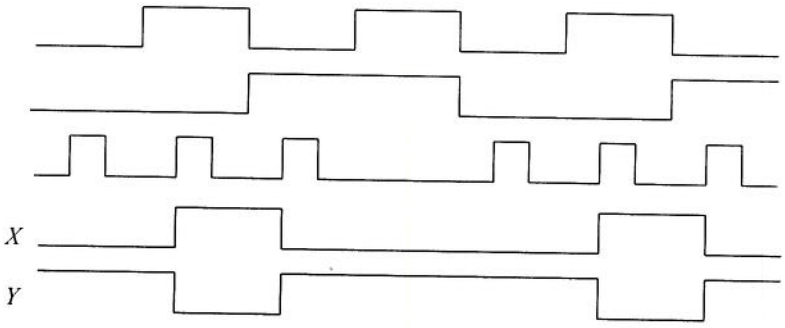


| AB \ CD | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 0 | 0 | 0 | 0 |
| 01 | 1 | 1 | 1 | 1 |
| 11 | 1 | 0 | 1 | 1 |
| 10 | 0 | 1 | 1 | 0 |

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

[4]

d)



[6]

e)

| Decimal | Hexadecimal | Binary | BCD |
|---|---|---|---|
| 7199 | 1C1F | ▨ | ▨ |
| ▨ | ▨ | 1000011110010010 | 8792 |
| ▨ | 27ED | 0010 0111 1110 1101 | ▨ |
| -2317 | F7F3 | ▨ | ▨ |

[8]

f)

| A | B | C | P | Q |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 | 0 |

[6]

g)

| Current state | Input | Next state | Output | |
|---|---|---|---|---|
| | | | P | Q |
| S0 | 0 | S1 | 1 | 0 |
| S0 | 1 | S2 | 1 | 1 |
| S1 | 0 | S2 | 1 | 1 |
| S1 | 1 | S0 | 0 | 1 |
| S2 | 0 | S0 | 0 | 1 |
| S2 | 1 | S0 | 1 | 0 |
| S3 | X | S0 | 0 | 1 |

[4]

## Answer to Question 2

a)

X[1:0] and Y[1:0] are set to correspond to ROM address lines A[3:2] and A[1:0] respectively (other combinations may also be used). For the ROM data, all D3 are set to zero, and D2, D1 and D0 represent the 'sum' bits S1 and S0, and the carry bit C0 respectively. This gives the solution below:

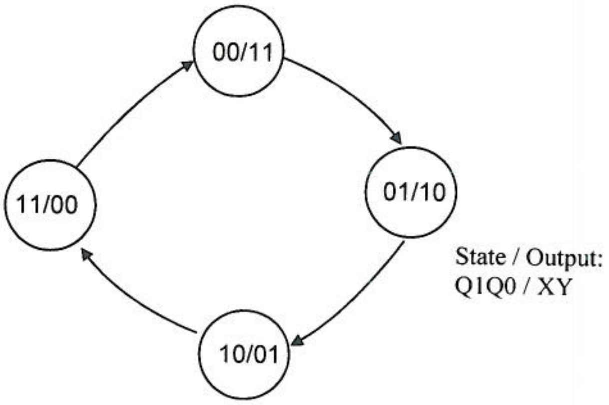
| Address | X1 | X0 | Y1 | Y0 | | S1 | S0 | C0 | Data |
|---|---|---|---|---|---|---|---|---|---|
| (Hex) | A3 | A2 | A1 | A0 | D3 | D2 | D1 | D0 | (Hex) |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 2 |
| 2 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 4 |
| 3 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 6 |
| 4 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 2 |
| 5 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 4 |
| 6 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 6 |
| 7 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 |
| 8 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 4 |
| 9 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 6 |
| A | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 |
| B | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 3 |
| C | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 6 |
| D | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 1 |
| E | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 3 |
| F | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 5 |

The answer can also show an alternative arrangement, with the position of X and Y on the address lines interchanged.

[10]

b) i)

Moore state diagram:



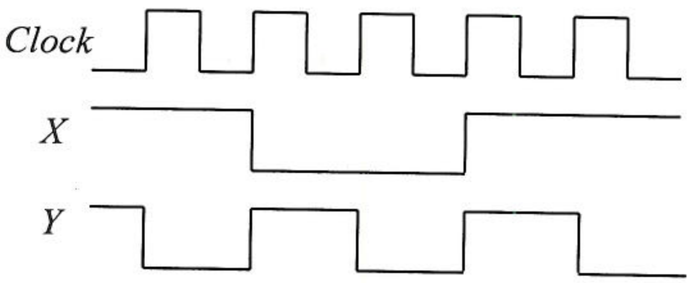
State transition table:

| Current state | Next state | Output | |
|---|---|---|---|
| | | X | Y |
| 00 | 01 | 1 | 1 |
| 01 | 10 | 1 | 0 |
| 10 | 11 | 0 | 1 |
| 11 | 00 | 0 | 0 |

[6]

ii)

Output waveforms:



[4]

c) Boolean expressions for H, E and L are shown below:

$$H = A_3\overline{B}_3 + \left(\overline{A_3 \oplus B_3}\right)A_2\overline{B}_2 + \left(\overline{A_3 \oplus B_3}\right)\left(\overline{A_2 \oplus B_2}\right)A_1\overline{B}_1 + \left(\overline{A_3 \oplus B_3}\right)\left(\overline{A_2 \oplus B_2}\right)\left(\overline{A_1 \oplus B_1}\right)A_0\overline{B}_0$$

$$E = \left(\overline{A_3 \oplus B_3}\right)\left(\overline{A_2 \oplus B_2}\right)\left(\overline{A_1 \oplus B_1}\right)\left(\overline{A_0 \oplus B_0}\right)$$

$$L = \overline{A}_3B_3 + \left(\overline{A_3 \oplus B_3}\right)\overline{A}_2B_2 + \left(\overline{A_3 \oplus B_3}\right)\left(\overline{A_2 \oplus B_2}\right)\overline{A}_1B_1 + \left(\overline{A_3 \oplus B_3}\right)\left(\overline{A_2 \oplus B_2}\right)\left(\overline{A_1 \oplus B_1}\right)\overline{A}_0B_0$$

Alternatively, given H and E as above, we have: $L = \overline{H}.\overline{E}$

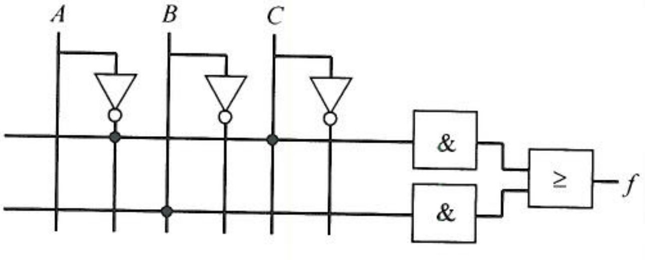
[10]

## Answer to Question 3

a)

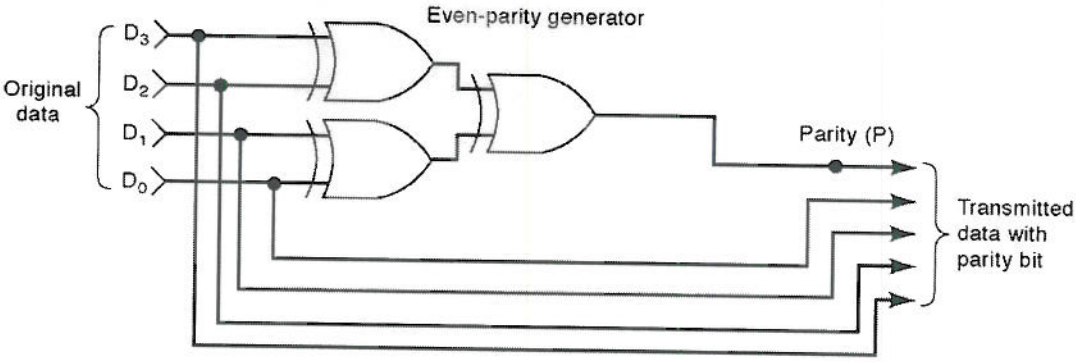
$$\begin{aligned} f &= (A+B+C)(\overline{A}+B) \\ &= \overline{A}B+\overline{A}C+AB+B+BC \\ &= \left(\overline{A}+A\right)B+\overline{A}C+B(C+1) \\ &= B+\overline{A}C \end{aligned}$$

Then, implement using a PAL device:



[5]

b)



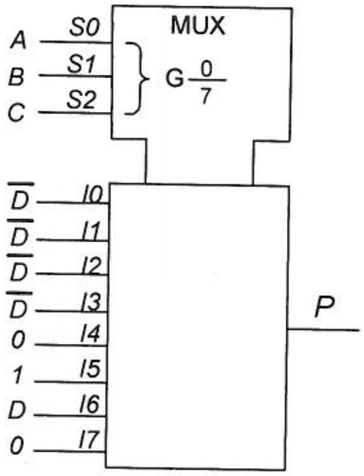
[5]

c)

As variable D cannot be used on the select lines S[2:0], we can organise *f* vs. A, B, C and D as follows:

| D \ ABC | 000 | 001 | 011 | 010 | 110 | 100 | 101 | 111 |
|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| | *I0* | *I1* | *I3* | *I2* | *I6* | *I4* | *I5* | *I7* |

Here, the columns correspond to the input lines of the multiplexer, with A, B and C on the select lines (alternative arrangements of A, B and C on the select lines are also possible). This gives the following implementation:



[10]

d) i)

The truth table for the encoder is as follows:

| A0 | A1 | A2 | A3 | A4 | A5 | A6 | A7 | X2 | X1 | X0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |

[5]

ii)

By inspection, this gives the following Boolean expressions:

$$X2 = A4 + A5 + A6 + A7$$

$$X1 = A2 + A3 + A6 + A7$$

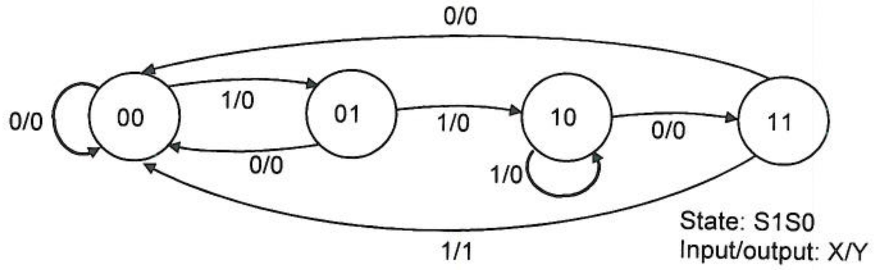
$$X0 = A1 + A3 + A5 + A7$$

[5]

## Answer to Question 4

a)

Mealy diagram for the FSM:



[8]

b)

State transition table for the FSM:

| Current state | | Input | Next state | | Output |
|---|---|---|---|---|---|
| S1 | S0 | X | S1+ | S0+ | Y |
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 1 |

[8]

c)

The transition table for the J-K flip-flop is the following:

| *Transition* | *J* | *K* |
|---|---|---|
| 0 → 0 | 0 | X |
| 0 → 1 | 1 | X |
| 1 → 0 | X | 1 |
| 1 → 1 | X | 0 |

Using this table, it is possible to draw Karnaugh maps for J1, K1 and J0, K0, as a function of S1, S0 and X, to give the next states S1+ and S0+. Y may be obtained directly from S1, S0 and X.

For S1+:

*K-map for J1:*

| $X$ \ $S1S0$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | 0 | 0 | X | X |
| 1 | 0 | 1 | X | X |

$$J1 = S0X$$

*K-map for K1:*

| $X$ \ $S1S0$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | X | X | 1 | 0 |
| 1 | X | X | 1 | 0 |

$$K1 = S0$$

For S0+:

*K-map for J0:*

| $X$ \ $S1S0$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | 0 | X | X | 1 |
| 1 | 1 | X | X | 0 |

$$J0 = \overline{S1}X + S1\overline{X}$$
$$\Rightarrow J0 = S1 \oplus X$$

*K-map for K0:*

| $X$ \ $S1S0$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | X | 1 | 1 | X |
| 1 | X | 1 | 1 | X |

$$K0 = 1$$

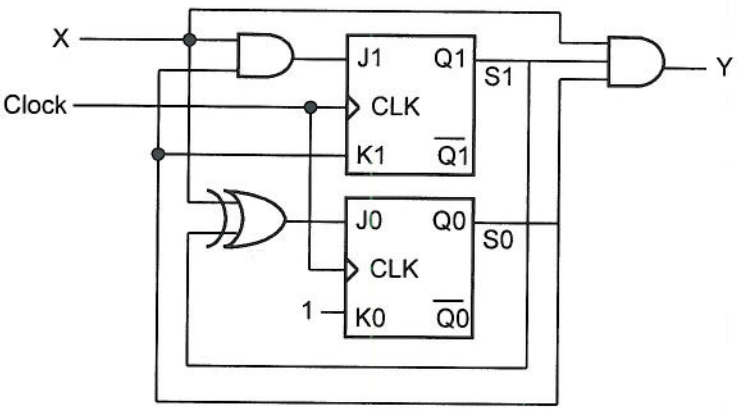
For Y:

Y can be found directly from the state transition table:

$$Y = S1.S0.X$$

[10]

d)



[4]