

IMPERIAL COLLEGE LONDON

DEPARTMENT OF ELECTRICAL AND ELECTRONIC ENGINEERING  
EXAMINATIONS 2013

EEE/ISE PART I MEng., BEng. and ACGI

**Digital Electronics I**

Time allowed: 2:00 hours

**There are THREE questions on this paper.**

**All questions are compulsory.**

**Q1 carries 40% of the marks. Questions 2 and 3 carry equal marks (30% each).**

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

Examiners responsible:

First Marker: *Zahid Durrani*

Second Marker: *Jeremy 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 Boolean algebra and/or De Morgan's theorem.

i)  $\overline{A}\overline{B}C + \overline{B}\overline{C} + AB\overline{C}$  [4]

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

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

$$f(A, B, C, D) = \Sigma(2, 3, 5, 8, 11), \text{ with a 'don't care' at } 12$$
 [4]

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

$$f = A\overline{B} + \overline{A}BC + AB\overline{C}$$
 [4]

- d) Assuming that all numbers are 8 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]

Hexadecimal	Octal	Unsigned decimal	Signed decimal	BCD
C5		?	?	
?	25			?

- e) The timing waveforms for signals  $A$  and  $B$  shown in Figure 1.1 are applied to the circuit shown in Figure 1.1. Draw timing waveforms for the signals  $C$ ,  $D$  and  $X$ , given that the initial value of  $X = 1$  and the initial value of  $Y = 0$ .

[6]

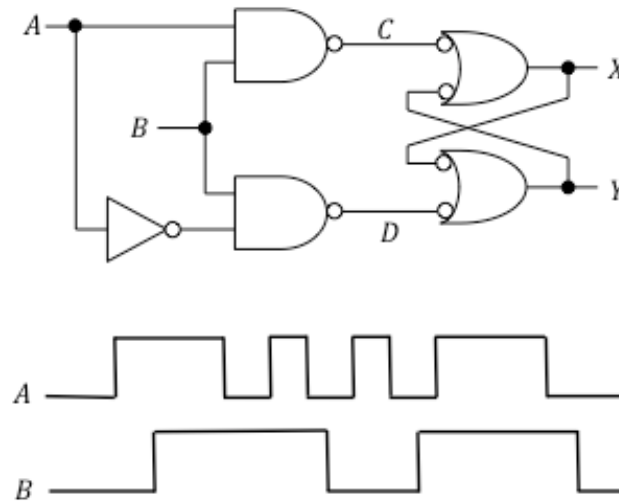


Figure 1.1

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

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

[4]

- g) Figure 1.2 shows a finite state machine (FSM) using two  $4 \times 1$  multiplexers, two D-type flip-flops, and an AND gate. The FSM has input  $A$ , output  $Z$ , and state  $Q1Q0$ . Assuming that initially  $Q1 = Q0 = 0$ , draw the Mealy state diagram for this FSM.

[6]

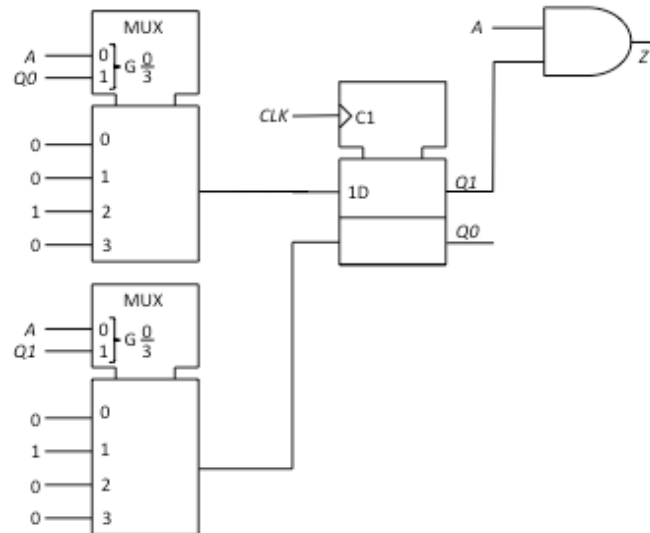


Figure 1.2

**Question 2**

2. a) An  $8 \times 2$  read-only memory (ROM), shown in Figure 2.1, is to be used to implement a 1-bit full-adder (FA) circuit. The ROM address lines A[2:0] are connected to signals  $X$ ,  $Y$ , and  $Z$ , with  $X$  connected to the least significant bit (LSB) and  $Z$  connected to the most significant bit (MSB) of the address. The LSB and MSB of the output lines provide the 'carry-out' ( $C$ ) and 'sum' ( $S$ ) signals of the FA, respectively.

- i) Draw a table showing the contents of the ROM, in binary and hexadecimal.

[4]

- ii) Hence, construct a 4-bit subtractor circuit using these ROMs, which subtracts  $P[3:0]$  from  $Q[3:0]$ . You may use any necessary gates with the ROMs.

[4]

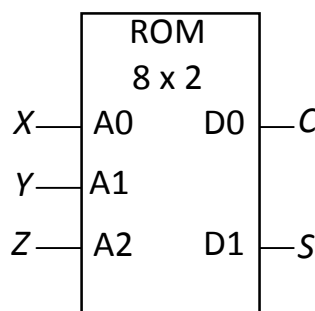


Figure 2.1

- b) Design a combinational logic circuit which finds the absolute value of the signed 4-bit binary number  $X[3:0]$ . You may use basic logic gates, NOT, OR, AND, XOR, or XNOR, only.

[10]

- c) A 4-input line-to-binary priority encoder, with active-low input/output lines, is shown in Figure 2.2. The circuit ensures that if more than one of the input lines  $A[3:0]$  is activated, then the output  $Z[1:0]$  will correspond to the highest numbered input. Furthermore, if all input lines are de-asserted, then the output lines may be considered as 'don't care'.

- i) Draw the truth table for the circuit.

[6]

- ii) By using Karnaugh maps or otherwise, find the gate level implementation for this circuit in sum-of-products form.

[6]

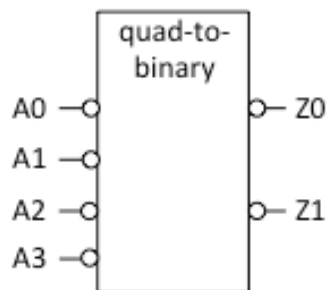


Figure 2.2

**Question 3**

3. a) A finite state machine (FSM) is to be designed such that it counts in the following sequence:

$$0 \rightarrow 1 \rightarrow 3 \rightarrow 6 \rightarrow 0 \rightarrow 1 \text{ etc.}$$

- i) Draw the Moore state diagram for the FSM.

[4]

- ii) Derive the state transition table for the FSM.

[4]

- iii) Hence, design the FSM using JK-type flip-flops. You may assume that any undefined states can be ignored.

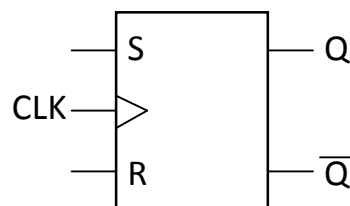
[10]

- iv) Sketch the circuit diagram for your design. This should show the JK-type flip-flops, any gates you have used, and the connections necessary for these.

[4]

- b) Figure 3.1 shows a clocked SR-type flip-flop. Implement this flip-flop using a D-type flip-flop. You may assume that any disallowed input combinations do not occur.

[8]



*Figure 3.1*

[THE END]



**Question 1**

1. a) (i)

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

Boolean algebra applied here after missing variables introduced (2<sup>nd</sup> line).

[4]

ii)

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

Unwrap the expression using De Morgan's theorem, do this step by step.

[4]

b)

$$f(A, B, C, D) = \Sigma(2, 3, 5, 8, 11), \text{ with a 'don't care' at } 12$$

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

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

Use the don't care as a '1' to allow a pair to be formed.

Here, 1 mark for drawing the Karnaugh map, 1 for filling it out correctly, 1 for the correct grouping, and 1 for the final expression.

[4]

c)

$$f = A\bar{B} + \bar{A}BC + ABC\bar{C}$$

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

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

The POS expression is written by choosing common variables, and ORing the inverted versions of these to form the expression in each bracket.

Here, 1 mark for drawing the Karnaugh map, 1 for filling it out correctly, 1 for the correct grouping, and 1 for the final expression.

[4]

d)

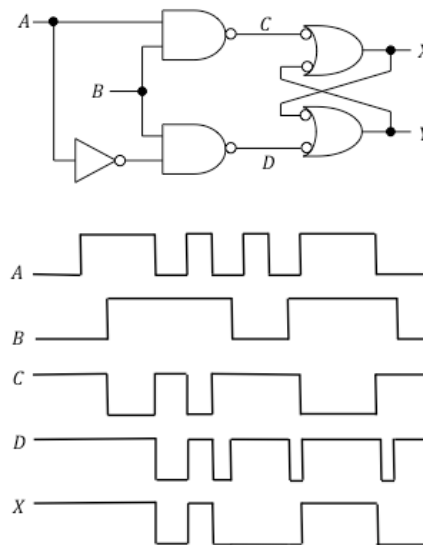
Hexadecimal	Octal	Unsigned decimal	Signed decimal	BCD
C5		197	-59	
15	25			0010 0001

Hexadecimal to unsigned decimal is straightforward, by multiplying by powers of 16. Hexadecimal to signed decimal can be obtained by converting to binary. As the MSB is '1', this is a negative number in 2s complement. The final step is to convert the 2s complement number into its decimal (negative) value. Hexadecimal to Octal can be found by converting to binary and then to Octal. For BCD, convert to decimal and then to BCD.

Give 2 marks per answer.

[8]

e)



This has a flip-flop at the end (Active low SR latch) at the end, with NAND gates enabled by 'B'.

Give 2 marks per waveform  $C$ ,  $D$  and  $X$ .

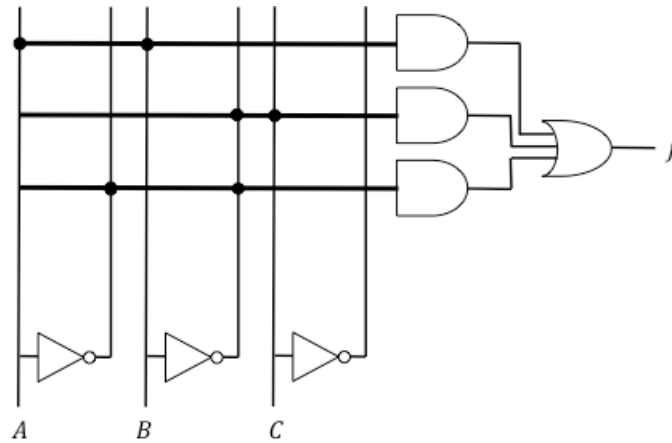
[6]

f)

$$\begin{aligned}
 f &= (\bar{A} + B + C)(A + \bar{B}) \\
 &= AB + AC + \bar{A}\bar{B} + C\bar{B} \\
 &= AB + AC(B + \bar{B}) + \bar{A}\bar{B} + C\bar{B}(A + \bar{A}) \\
 &= AB + ABC + A\bar{B}C + \bar{A}\bar{B} + A\bar{B}C + \bar{A}\bar{B}C \\
 &= AB(C + 1) + \bar{A}\bar{B} + \bar{B}C(A + \bar{A}) \\
 &= AB + \bar{A}\bar{B} + \bar{B}C
 \end{aligned}$$

An equivalent, equally simplified expression is  $f = AB + AC + \bar{A}\bar{B}$

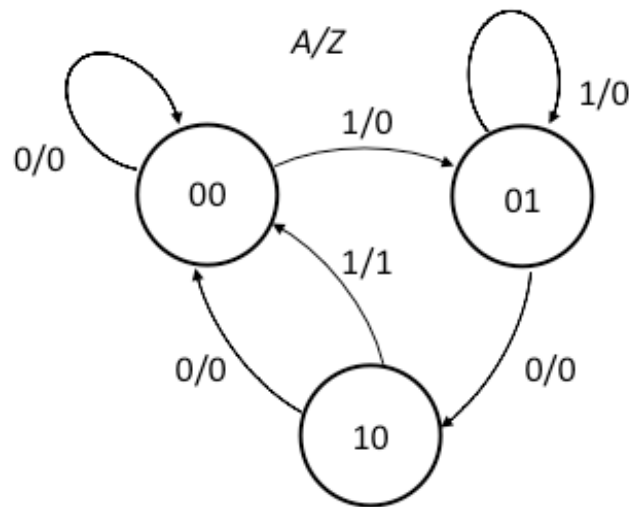
The PAL implementation for the first of the two expressions is shown below, the other can be used as well. The 3 lines to each AND gates are shown as single, 'bus', lines for simplicity.



Give 2 marks for simplification of the given expression using Boolean algebra or a Karnaugh map. Give 2 marks for the PAL implementation

[4]

g)



The outputs Q0 and Q1 are fed back into select lines of the multiplexers. Note the priority of the select lines. A further input A is also used. Note the position of where Z and A are written with respect to the states.

Give 2 marks for correct number of states, 2 mark for correct connections, 2 marks for correct labelling

[6]

**Question 2**

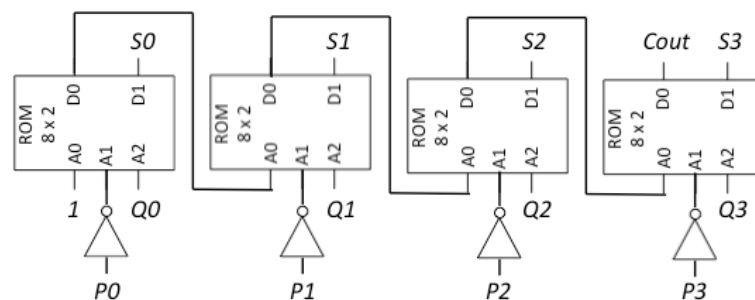
2. a) (i)

A2	A1	A0	D1	D0	ROM contents
0	0	0	0	0	0
0	0	1	1	0	2
0	1	0	1	0	2
0	1	1	0	1	1
1	0	0	1	0	2
1	0	1	0	1	1
1	1	0	0	1	1
1	1	1	1	1	3

Note the priority of the address lines and the variables connected to these. You also need to know what is the truth table for a full-adder. The ROM contents are given in hexadecimal.

Give 1 mark for correct input addresses, and 1 mark for each of the binary and hexadecimal columns.

[4]

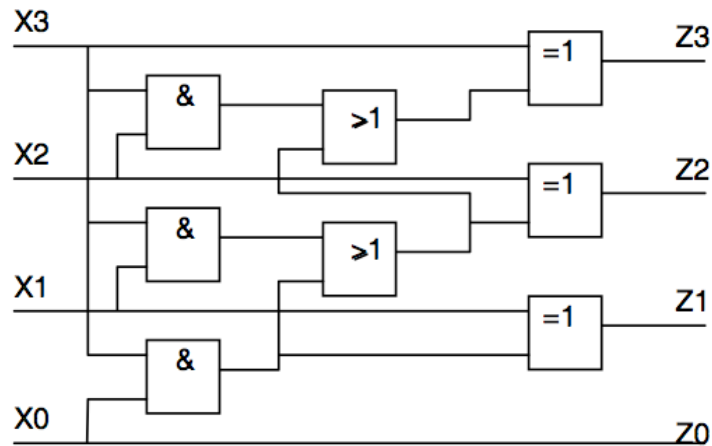


Straightforward application of 4 bit ripple FA to a subtractor. Works by converting P to 2s complement form, and adding to Q.

Give 2 marks for correct interconnections, 1 for the inversion of the bits comprising P, and 1 for the correct labels and inputs.

[4]

b)



This is a study-group question.

Give 2 marks for identifying absolute values.

Give 2 marks for using the MSB to determine if X is negative or positive.

Give 2 marks for identifying that X and  $-X$  have the same LSBs, up to and including the least significant bit that equals 1.

Give 2 marks for the logic of the circuit, i.e. if the number is negative, identify the first LSB = 1 and then invert the more significant bits.

Give 2 marks for the correct implementation.

If 'brute-force' approach, with a 4 input truth table, Karnaugh maps and Boolean expressions, give up to 6 marks.

[10]

c) (i)

A3	A2	A1	A0	Z1	Z0
0	X	X	X	0	0
1	0	X	X	0	1
1	1	0	X	1	0
1	1	1	0	1	1
1	1	1	1	X	X

Note the active-low inputs and outputs here.

Give 1 mark for each correct column.

If 'active high' inputs/outputs are shown, but logic is fine, then give up to 5 marks.

[6]

(ii) The expanded truth table is as follows:

A3	A2	A1	A0	Z1	Z0
0	0	0	0	0	0
0	0	0	1	0	0
0	0	1	0	0	0
0	0	1	1	0	0
0	1	0	0	0	0
0	1	0	1	0	0
0	1	1	0	0	0
0	1	1	1	0	0
1	0	0	0	0	1
1	0	0	1	0	1
1	0	1	0	0	1
1	0	1	1	0	1
1	1	0	0	1	0
1	1	0	1	1	0
1	1	1	0	1	1
1	1	1	1	X	X

This gives the following Karnaugh maps and expressions:

		A1A0			
		00	01	11	10
Z1	A3A2				
	00	0	0	0	0
	01	0	0	0	0
	11	1	1	X	1
	10	0	0	0	0

$$Z1 = A3A2$$

		A1A0			
		00	01	11	10
Z0	A3A2				
	00	0	0	0	0
	01	0	0	0	0
	11	0	0	X	1
	10	1	1	1	1

$$Z0 = A3A1 + A3\overline{A2}$$

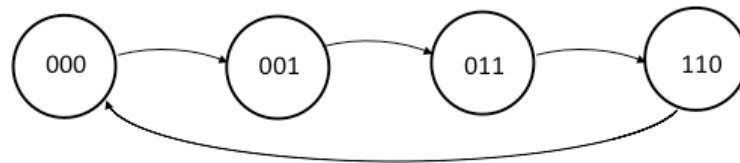
Give 2 marks for either showing the extended truth table, or directly filling out the Karnaugh map correctly. Give 1 mark for the correct SOP grouping in each Karnaugh map, and 1 for each of the final expressions.

[6]



**Question 3**

3. a) i) Using states  $Q_2Q_1Q_0$  to define the FSM:



Give 2 marks for four, labelled states and 2 for correct interconnections.

[4]

- ii) Using  $Q_2Q_1Q_0$  as the present state of the FSM, and  $Q_2^+Q_1^+Q_0^+$  as the next state:

$Q_2$	$Q_1$	$Q_0$	$Q_2^+$	$Q_1^+$	$Q_0^+$
0	0	0	0	0	1
0	0	1	0	1	1
0	1	1	1	1	0
1	1	0	0	0	0

Give 1 mark for each correct row of the table.

[4]

- iii) The FSM can be implemented using three J-K flip-flops, one each to store  $Q_2$ ,  $Q_1$  and  $Q_0$ .

We first require the output transition table for the J-K flip-flop:

Output transition	J	K
$0 \rightarrow 0$	0	X
$0 \rightarrow 1$	1	X
$1 \rightarrow 0$	X	1
$1 \rightarrow 1$	X	0

Using this table, it is possible to fill out the J-K inputs needed for each output, where we assume undefined states are don't cares:

For  $Q0^+$ :

$$J0$$

	$Q1Q0$	00	01	11	10
$Q2$					
0		1	X	X	X
1		X	X	X	0

$$\Rightarrow J0 = \overline{Q2}$$

$$K0$$

	$Q1Q0$	00	01	11	10
$Q2$					
0		X	0	1	X
1		X	X	X	X

$$K0 = Q1$$

For  $Q1^+$ :

$$J1$$

	$Q1Q0$	00	01	11	10
$Q2$					
0		0	1	X	X
1		X	X	X	X

$$\Rightarrow J1 = Q0$$

$$K1$$

	$Q1Q0$	00	01	11	10
$Q2$					
0		X	X	0	X
1		X	X	X	1

$$K1 = \overline{Q0}$$

For  $Q2^+$ :

$$J2$$

	$Q1Q0$	00	01	11	10
$Q2$					
0		0	0	1	X
1		X	X	X	X

$$\Rightarrow J2 = Q1$$

$$K2$$

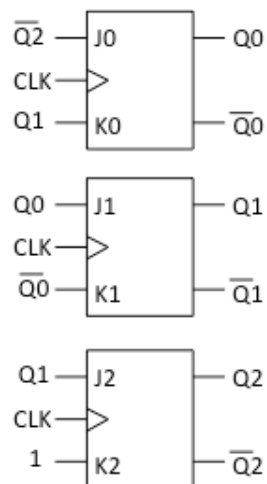
	$Q1Q0$	00	01	11	10
$Q2$					
0		X	X	X	X
1		X	X	X	1

$$K2 = 1$$

Give 1 mark for the output transition table, 1 mark for each of the six Karnaugh maps, and 1 mark for each of the three J, K expressions.

[10]

iv)



Give 1 mark for three J-K flip-flops, and 1 mark for the correct inputs at each flip-flop.

[4]

b) This is a study-group question.

Truth table for the clocked SR flip-flop:

S	R	CLK	Q
0	0	↑	Q
0	1	↑	0
1	0	↑	1
1	1	↑	Undefined

[1]

For a D-type flip-flop implementation, we can then draw the following state transition table:

S	R	Q	Q+
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	X
1	1	1	X

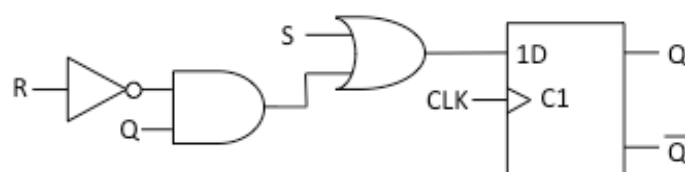
[2]

This gives the Karnaugh map, Boolean expression and implementation:

		$SR$			
		00	01	11	10
$Q$	0	0	0	X	1
	1	1	0	X	1

$$\Rightarrow Q^+ = S + Q\bar{R}$$

[3]



[2]

[Total marks: 8]