AQA Computer Science A-Level

4.6.4 Logic gates

Past Paper Mark Schemes

January 2009 Comp 2

10 (a)

OR

Input A	Input B	Output
0	0	**0**
0	1	**1**
1	0	**1**
1	1	**1**

AND

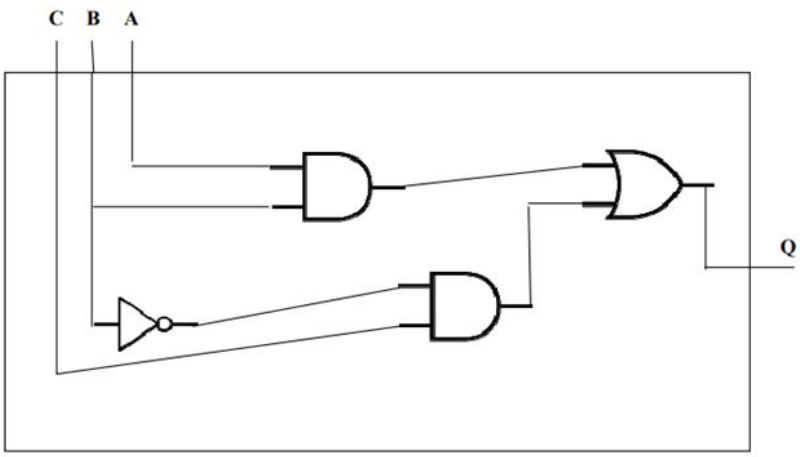
Input A	Input B	Output
0	0	**0**
0	1	**0**
1	0	**0**
1	1	**1**

1 mark per correct table **2**

(b) (i) $Q = A.B + C.\overline{B}$

1 mark for $A.B$ *or for* $C.\overline{B}$
2 marks for $A.B + C.\overline{B}$
A AND instead of . **A** OR instead of + **2**

(ii) *1 mark for each gate with correct inputs;;;;*
Allow two lines from B



4

January 2010 Comp 2

7	(a)		1; **A** True 1; **A** True 0; **A** False	3
7	(b)	(i)	AND and NOT	1
7	(b)	(ii)	NAND // NAND gate **R** NOT AND	1
7	(c)		Minimise cost of production; Reduce propagation delay//speed up processing; Minimise heat generated; Reduce power consumption; **NE** simpler to produce/makes circuit simpler **NE** reduce number of gates in chip	1

January 2011 Comp 2

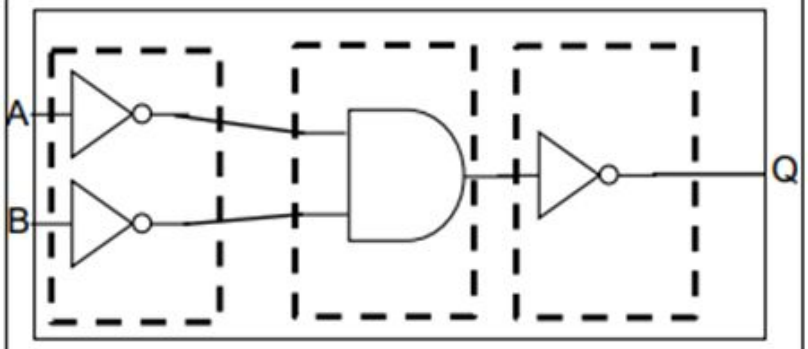
2			G H K 0 1 1 1 1 0 1 0 1 1 0 1 ; ; ; 1 mark for each correct column	3	

January 2012 Comp 2

2	a		$X \oplus Y$; $X\overline{Y} + \overline{X}.Y$; **A** alternative notations : X XOR Y X EOR Y X AND NOT Y OR NOT X AND Y	1	Acceptable notation for symbols : For X.Y allow X∧Y,X∩Y,XY For X+Y
					allow X∨Y,X∪Y For X allow ~X
2	b		$X.\overline{Y}$; A alternative notations : X AND NOT Y;	1	
2	c	i	<table><tr><th colspan="2">Inputs</th><th colspan="2">Outputs</th></tr><tr><th>X</th><th>Y</th><th>C</th><th>S</th></tr><tr><td>0</td><td>0</td><td>**0**</td><td>**0**</td></tr><tr><td>0</td><td>1</td><td>**0**</td><td>**1**</td></tr><tr><td>1</td><td>0</td><td>**0**</td><td>**1**</td></tr><tr><td>1</td><td>1</td><td>**1**</td><td>**0**</td></tr></table> One mark for C column; One mark for S column;	2	
2	c	ii	Addition // adder; **A** sum;	1	

January 2013 Comp 2

4	a			3



1 mark – logic of first part satisfies NOT A, NOT B;

1 mark – inputs into an AND gate;

1 mark – output from AND gate passes through a NOT gate and connected to Q;

4	b			4

A	B	A + B
0	0	0
0	1	1
1	0	1
1	1	1

1 mark for correct A + B column;

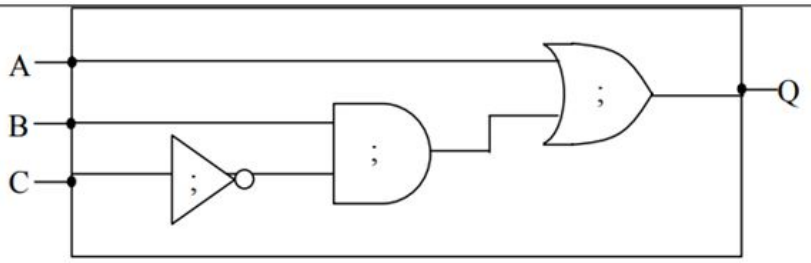
A	B	$\overline{A}$	$\overline{B}$	$\overline{A}.\overline{B}$	$\overline{\overline{A}.\overline{B}}$
0	0	1	1	1	0
0	1	1	0	0	1
1	0	0	1	0	1
1	1	0	0	0	1

1 mark for columns $\overline{A}$ and $\overline{B}$ being correct;
1 mark for $\overline{A}.\overline{B}$ column being correct;
1 mark for $\overline{\overline{A}.\overline{B}}$ column being correct;

NOTE: Can follow through into $\overline{A}.\overline{B}$ column from previous two

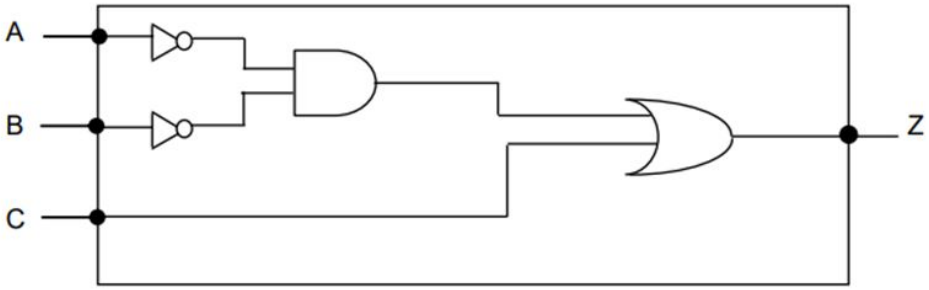
June 2010 Comp 2

9	(a)		**OR Gate** **Input A** / **Input B** / **Output Q** 0 / 0 / **0** 0 / 1 / **1** 1 / 0 / **1** 1 / 1 / **1** ↑ 1 mark **XOR Gate** **Input A** / **Input B** / **Output Q** 0 / 0 / **0** 0 / 1 / **1** 1 / 0 / **1** 1 / 1 / **0** ↑ 1 mark **1 mark** for each correct output column **A** True for 1, False for 0	2

9	(b)		 **1 mark** for NOT gate correctly linked to input C; **1 mark** for AND gate correctly linked to B and $\overline{C}$ as input; **A** if AND gate linked directly to C	
			1 mark for OR gate with inputs from A and the output of an AND gate and output connected to Q;	3

June 2011 Comp 2

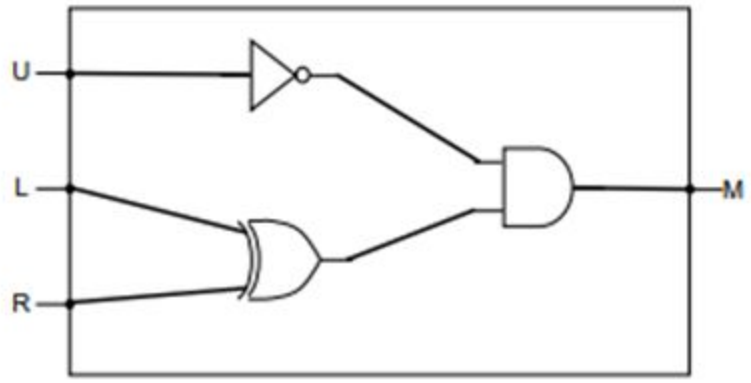
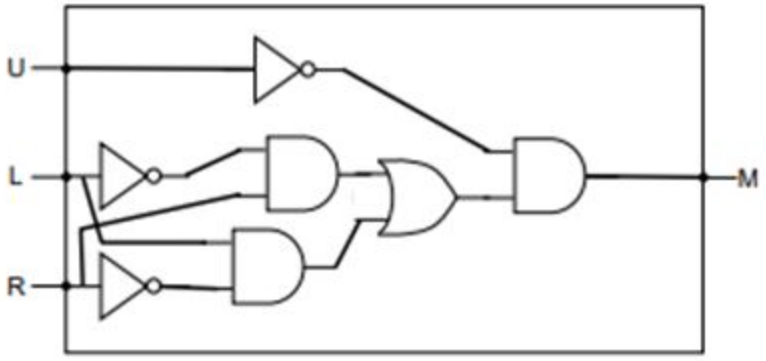
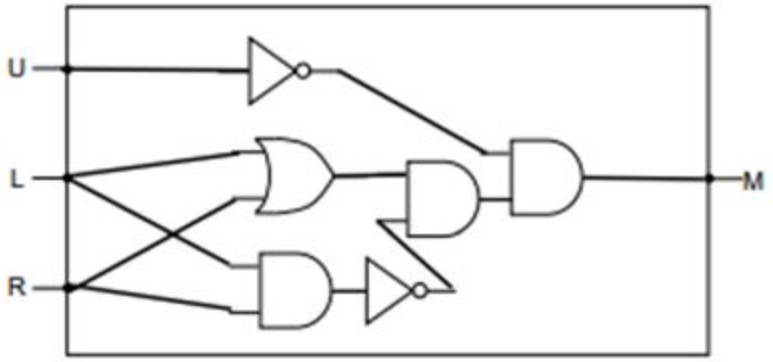
3	a		NAND 1 1 1 0;	NOR 1 0 0 0;	**2**

3	b		 1 mark for NOT gates on both A and B; 1 mark for AND with inputs from $\overline{A}$ and $\overline{B}$; **A** inputs from A and B 1 mark for OR gate with inputs from AND gate output and C and output connected to Z;	3

June 2012 Comp 2

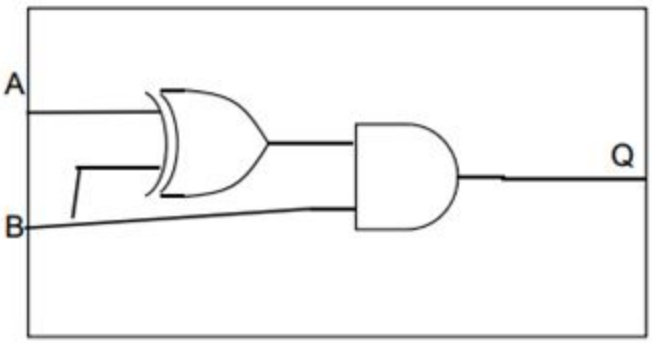
8	a		**AND Gate** **Input X** / **Input Y** / **Output Q** 0 / 0 / 0 0 / 1 / 0 1 / 0 / 0 1 / 1 / 1 **XOR Gate** **Input X** / **Input Y** / **Output Q** 0 / 0 / 0 0 / 1 / 1 1 / 0 / 1 1 / 1 / 0 1 mark for each of the output columns	2

8	b	i	$(L \oplus R) . \overline{U}$ [Brackets are not necessary] 1 mark for use of correct operands (L,R,U); 1 mark for use of XOR with L,R; 1 mark for NOT U anded with other part; alternative : $(L + R) . \overline{(L . R)} . \overline{U}$ 1 mark for use of correct operands (L,R,U); 1 mark for alternative XOR expression; 1 mark for AND NOT U; alternative : $(L . \overline{R} + \overline{L} . R) . \overline{U}$ 1 mark for use of correct operands (L,R,U); 1 mark for alternative XOR expression; 1 mark for AND NOT U;	3	Acceptable notation for symbols : ~ for NOT X.Y allow X AND Y, X∋Y,X)Y, XY X+Y allow X OR Y, X(Y, X∗Y

8	b	ii	 L, R connected to XOR gate; U connected to NOT gate; Output of a two input AND gate connected to M; **MAX 2** if circuit does not reflect the correct logic **Alternative :**  U connected to NOT gate; Correct gates used for L and R before last AND gate; Output of a two input AND gate connected to M; **Alternative :**  Marked as above alternative.	3

June 2013 Comp 2

6	(a)		AND; NOR; XOR; **A.** EXOR // EOR // NEQ // exclusive OR;	**3**

6	(c)		 Inputs A and B connected to an XOR gate; Input from B and output of XOR gate connected to an AND gate with output going to Q;	**2**

June 2016 AS Paper 2

05	1	**Mark is for AO1 (knowledge)** XOR // EOR // Exclusive OR;	1

05	2	**Mark is for AO2 (apply)** $\overline{C}$;	1
05	3	**Mark is for AO2 (apply)** C;	1
05	4	**Marks are for AO2 (apply)** **Mark as follows:** **1 mark:** column T correct; **1 mark:** column S correct; **1 mark:** column R correct;	3

C	B	A	T	S	R
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	1
0	1	1	0	1	0

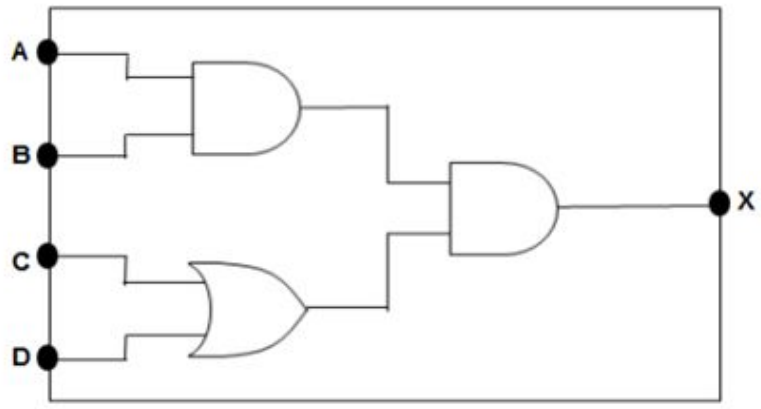
05	5	**Mark is for AO2 (analyse)** Use this circuit on the binary number to be subtracted and add the result to the other binary number; **A.** Any equivalent answers **R.** Number to be added IS negative	1

June 2017 AS Paper 2

05	1	**Mark is for AO1 (knowledge)** NAND; **A.** NOT AND	**1**

05	2	**Marks are for AO2 (apply)** |	| 1 | 2 | 3 | 4 | 5 | | **A** | **B** | $\overline{\mathbf{B}}$ | $\mathbf{A}+\overline{\mathbf{B}}$ | $\overline{\mathbf{A}}$ | $\overline{\mathbf{A}}\cdot\mathbf{B}$ | $\overline{\overline{\mathbf{A}}\cdot\mathbf{B}}$ | | 0 | 0 | 1 | 1 | 1 | 0 | 1 | | 0 | 1 | 0 | 0 | 1 | 1 | 0 | | 1 | 0 | 1 | 1 | 0 | 0 | 1 | | 1 | 1 | 0 | 1 | 0 | 0 | 1 | **Mark as follows:** **1 mark** for column 1 and 3 correct **1 mark** for column 4 correct **1 mark** for columns 2 and 5 correct and identical **I.** order of columns	**3**

June 2017 Paper 2

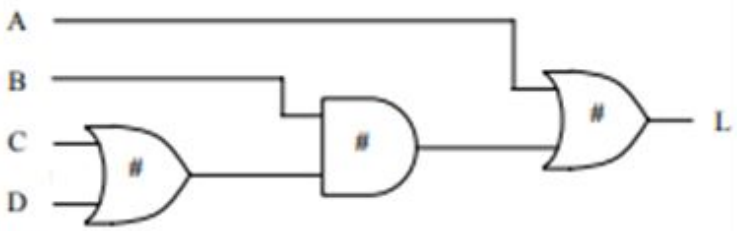
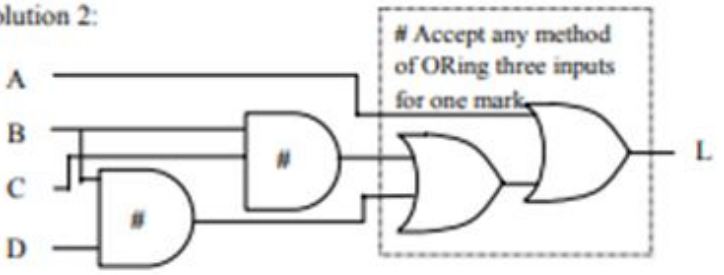
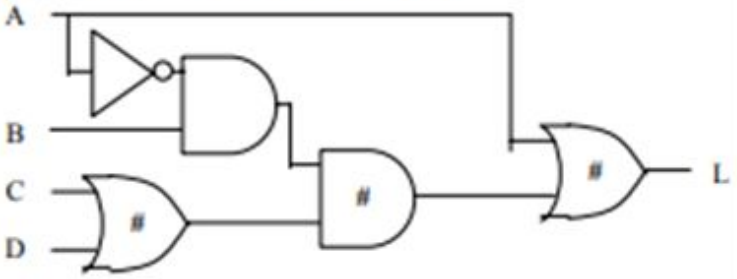
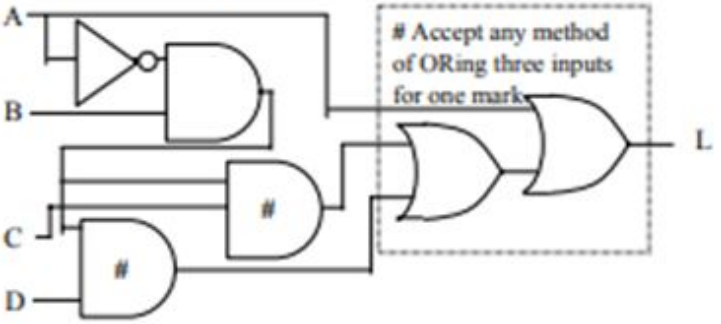
04	**1**	**All marks AO2 (apply)**  **1 mark:** inputs A and B connected to an AND gate; **1 mark:** inputs C and D connected to an OR gate; **1 mark:** output of an AND gate (but not the same one as connected to inputs A and B)	**3**

		connected to X; **MAX 2** if circuit does not fully represent the logic of the system OR the circuit diagram contains any errors	
04	**2**	**All marks AO2 (apply)** $X = A \cdot B \cdot (C + D)$ **1 mark:** either $A \cdot B$ or $C + D$ somewhere in an incorrect expression **2 marks:** fully correct expression **A.** A logically equivalent expression for **2 marks**	**2**

04	**4**	**Mark is for AO1 (knowledge)** Used to store state (of data input) // used as a memory (unit); **R.** If stated that maintains state when power turned off	**1**
04	**5**	**All marks AO1 (knowledge)** Input is: Clock / trigger / enable; **R.** Set / reset Used For: State of data input is stored // output is updated to reflect current status of input; **A.** Synchronise operation of a group of flip-flops **R.** Changes state/value of flip-flop	**2**

June 2009 Comp 2

4	(a)	NOR (Gate) **I** case of answer i.e. nor is allowed	**1**

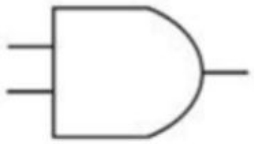
4	(b)	(i)	Solution 1:  Solution 2:  Solution 3 (solution 1 plus check for A off):  Solution 4 (solution 2 plus check for A off):  **1 mark for each correctly linked gate that is marked with a #** **A** 3-input OR gate **P1** for any unnecessary gates in a solution that would otherwise get 3 marks. **P1** for any solution that would not correctly implement the logic but would otherwise get 3 marks. Mark from left to right until first mistake encountered then from right to left. When marking left to right award 1 mark for each gate correctly connected to its inputs. When marking right to left award 1 mark for each gate correctly connected to its output.	**3**

Specimen AS Paper 2

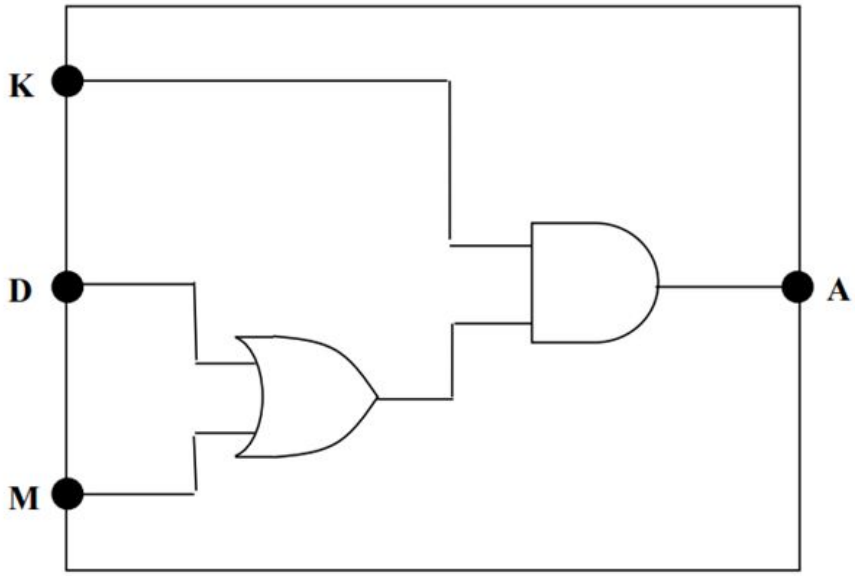
		(ii)	$A+B\cdot(C+D)$ $A+\overline{B}\cdot C+B\cdot D$ $A+\overline{A}\cdot B\cdot(C+\overline{D})$ $A+\overline{A}\cdot B\cdot C+\overline{A}\cdot B\cdot D$ **A** Insertion of extra brackets that do not affect logic of expression Note: Expression does not need to match diagram drawn in (i). **A** alternative notations : • For $X\cdot Y$ allow X AND Y, $X\wedge Y$, $X\cap Y$, XY • For X+Y allow X OR Y, $X\vee Y$, $X\cup Y$ • For $\overline{X}$ allow NOT X, $\neg X$	**1**

09	**1**		**Marks are for AO1 (knowledge)** **1 mark:** Table completed correctly; **1 mark:** AND gate symbol drawn;	2

A	B	Q
0	0	0
0	1	0
1	0	0
1	1	1



Specimen Paper 2

03	1	**All marks AO2 (apply)**  **1 mark:** inputs D and M connected to an OR gate; **1 mark:** inputs K and output of OR gate connected to AND gate plus output connected to A; **A.** a logically equivalent circuit	2

03	2	**All marks AO2 (apply)** $A = (D + M) \cdot K$ **1 mark:** D + M somewhere in expression, even if full expression incorrect **1 mark:** fully correct expression **A.** A logically equivalent expression	2

03	**3**	**1 mark for AO1 (understanding), 1 mark for AO2 (application) and 1 mark for AO1 (knowledge)** **AO1 (understanding): 1 mark:** Flip-flop will store the state of its input // Flip-flop acts as memory; **AO2 (application): 1 mark:** Insert into circuit between the output of the OR gate and the AND gate // after the AND gate; **AO1 (knowledge): 1 mark:** Clock signal // trigger // signal to indicate when the value (of the input) should be stored/read;	3