

## GCE A LEVEL – COMPUTER SCIENCE UNIT 3 QUESTION PACK

1500U30-1 · 2015 spec Unit 3 Topic 2 · A2 unit, first sat 2017, 100 marks, 2h paper

**REVISE**.wales**COMPUTER SCIENCE – UNIT 3 · Boolean Algebra – Truth Tables, Simplification & Masking**

Topic 3.2 – Boolean identities, De Morgan's laws, deriving expressions from truth tables, and bitwise masking

*Building truth tables for 2- and 3-variable expressions, proving De Morgan's laws, simplifying expressions step-by-step citing the law used at each stage, expressing truth-table outputs as a sum-of-products, and using AND masks to extract or test specific bits in a register.*

2015 specification · current

**Estimated time for entire question pack: ~2 h 10 min**

*Derived from the Unit 3 pace of ~1.5 min/mark, padded for written-prose answers (87 marks over 13 questions).*

*You are advised to **not** attempt to complete all of this in one sitting.*

**ABOUT THIS QUESTION PACK**

This is a **comprehensive topic question pack**, not a single mock paper. It contains every question from the WJEC A2 Unit 3 papers (Summer 2017 – Summer 2024, COVID gap) that maps onto Topic 3.2 of the 2015 specification.

Questions are ordered by source paper date.

**INSTRUCTIONS**

Use black ink or black ball-point pen. Show all working. A calculator is allowed where useful.

*All question content is © WJEC CBAC Ltd. and reproduced for revision purposes.*

For Examiner's use only

Q	Source	Max	Mark
1	S17 Q3	7	
2	S17 Q5	8	
3	S18 Q3	3	
4	S18 Q4	8	
5	S19 Q2	10	
6	S19 Q3	3	
7	S19 Q8	4	

Q	Source	Max	Mark
8	S22 Q3	11	
9	S22 Q10	4	
10	S23 Q3	4	
11	S23 Q4	10	
12	S24 Q3	12	
13	S24 Q10	3	
<b>Total</b>		<b>87</b>	

# Boolean Algebra – Truth Tables, Simplification & Masking – what the spec asks

WJEC GCE A Level Computer Science (from 2015) · Unit 3: Programming & System Development · Topic 3.2.

## De Morgan's Laws

- $\text{NOT}(A \cdot B) = \text{NOT } A + \text{NOT } B$
- $\text{NOT}(A + B) = \text{NOT } A \cdot \text{NOT } B$
- Use to push negations across AND/OR or convert one form to the other.
- Often the unlocking move in long simplifications.

## Boolean identities & rules

- Idempotent:  $A \cdot A = A$   $A + A = A$
- Identity:  $A \cdot 1 = A$   $A + 0 = A$   $A \cdot 0 = 0$   $A + 1 = 1$
- Complement:  $A \cdot \text{NOT } A = 0$   $A + \text{NOT } A = 1$
- Absorption:  $A + A \cdot B = A$   $A \cdot (A + B) = A$

## Truth tables

- $n$  inputs  $\Rightarrow 2^n$  rows; fill input combinations in binary-count order.
- Build output column-by-column for each sub-expression, then combine.
- Compare resulting columns of two expressions to test for equivalence (proof).
- Easy way to verify a simplified expression is correct.

## Deriving expressions from truth tables

- Sum-of-products (SOP): for each row where output = 1, AND together the inputs (negated if 0).
- OR together those product terms.
- Example: rows  $(0,0,1) \rightarrow 1$ ,  $(0,1,0) \rightarrow 1$  give  $\text{NOT } A \cdot \text{NOT } B \cdot C + \text{NOT } A \cdot B \cdot \text{NOT } C$ .
- Simplify the resulting expression with identities / De Morgan.

## Simplification strategy

1. Expand brackets (distributive).
2. Apply identities to collapse constants ( $A \cdot 1$ ,  $A + 0$ ,  $A + 1 \dots$ ).
3. Apply absorption to drop redundant terms.
4. Apply De Morgan if it helps factor or convert form.
5. Show each step on a new line citing the law used.

## Masking

- Use bitwise AND with a mask to extract or test specific bits.
- Mask 00001111 isolates the least significant 4 bits.
- Mask 10000000 tests the most significant bit (sign in two's complement).
- Result  $\neq 0 \Rightarrow$  bit was set; result = 0  $\Rightarrow$  bit was clear.

# Boolean Algebra – Truth Tables, Simplification & Masking in one page

Quick-reference notes – revisit before each question.

## De Morgan

$$\text{NOT}(A \cdot B) \equiv \text{NOT } A + \text{NOT } B$$

$$\text{NOT}(A + B) \equiv \text{NOT } A \cdot \text{NOT } B$$

Push negations through brackets or convert AND $\leftrightarrow$ OR.

## Identities

$$A \cdot A = A \quad A + A = A \text{ (idempotent)}$$

$$A \cdot 1 = A \quad A \cdot 0 = 0$$

$$A + 1 = 1 \quad A + 0 = A$$

$$A + \text{NOT } A = 1 \quad A \cdot \text{NOT } A = 0$$

## Absorption

$$A + A \cdot B \equiv A$$

$$A \cdot (A + B) \equiv A$$

Spot these early – they collapse whole sub-expressions.

## Truth-table rows

2 inputs  $\Rightarrow$  4 rows.

3 inputs  $\Rightarrow$  8 rows.

n inputs  $\Rightarrow 2^n$  rows.

Fill input cols in standard binary count, build output cols one expression at a time.

## SOP from truth table

For each row where output = 1, AND the inputs (NOT'ing those that are 0).

OR those product terms together.

Simplify the result with identities / De Morgan.

## Masking

AND with 00001111  $\Rightarrow$  isolate low 4 bits.

AND with 10000000  $\Rightarrow$  test sign bit.

Result  $\neq 0 \Rightarrow$  bit was set; result = 0  $\Rightarrow$  bit was clear.

Answer **all** questions.

1. Two human computer interfaces (HCI) are voice input and touch screen.
- (a) Give **two** benefits of using a touch screen interface on a mobile device. [2]
- (b) Describe the difficulties in creating a natural language interface for voice input. [4]

2. Explain the terms object, class and method in object-oriented programming. [3]

3. (a) Draw a truth table to show the value of **P** for all possible values of **A**, **B** and **C** for the following Boolean expression:

$$P = \bar{A} \cdot B + C \quad [4]$$

- (b) Using the data in the 8 bit register below, design a mask and use it to demonstrate how a logical operation can be used to extract the least significant 4 bits. [3]

Bit number	7	6	5	4	3	2	1	0
Register contents	1	1	1	0	0	0	1	1

4. (a) Describe the processes carried out during the lexical, syntax and semantic analyses stages of compilation. [6]
- (b) Describe **one** advantage of using a programming language that requires compiling compared with a programming language that requires interpreting. [2]
- (c) Describe **two** advantages of using a programming language that requires interpreting compared with a programming language that requires compiling. [4]
- (d) State the purpose of an assembler. Describe the difference between the source code of an assembler and the source code of a compiler. [4]

5. (a) Using the laws of Boolean algebra and De Morgan's theorem simplify the following Boolean expression:

$$A \cdot (\overline{A \cdot B}) \quad [4]$$

- (b) Simplify the following Boolean expression:

$$A \cdot B + A \cdot (B + C) + B \cdot (B + C) \quad [4]$$

6. Write a bubble sort algorithm, using pseudo-code, to sort an array of integers into ascending order. [9]

7. The name of a constant in a certain computer language must either be a single uppercase letter, or a single uppercase letter followed by one or more uppercase letters or digits.

- (a) Produce an appropriate syntax diagram to define a constant in this language. [4]

- (b) Produce an appropriate Backus-Naur Form (BNF) definition for a constant in this language. [4]

8. Below is part of an algorithm that initialises a two dimensional array named GRID of size  $N \times N$  with zeros.

```
for i = 0 to N - 1
    for j = 0 to N - 1
        GRID(i,j) = 0
    next j
next i
```

Evaluate the efficiency of the algorithm and, using Big O notation, determine the growth rate for the time performance. [4]

9. (a) Define the term algorithm. Other than pseudo-code, state **one** method of expressing an algorithm. [2]

- (b) Describe the main characteristics of a recursive algorithm. [2]

- (c) Describe **two** disadvantages of using a recursive algorithm. [2]

Answer **all** questions.

1. A binary tree structure is designed to contain strings and uses the following rules:
  - The left pointer indicates the condition “earlier or at the same position in the alphabet”
  - The right pointer indicates the condition “later in the alphabet”
  - (a) Construct a binary tree using these rules and the data entered in the following order:  
*Goat, Duck, Fox, Bear, Ant, Cat, Leopard, Owl, Mayfly, Insect, Jaguar, Emu.*  
You may use the initial letters if you wish. [2]
  - (b) Carry out a pre-order traversal of the tree and give **one** use of pre-order traversals. [2]
  - (c) Carry out an in-order traversal of the tree and give **one** use of in-order traversals. [2]
  - (d) Carry out a post-order traversal of the tree and give **one** use of post-order traversals. [2]
  
2. The evaluation of a computer based solution should consider system functionality and system performance.
  - (a) Identify a criterion for the evaluation of the functionality of a system and a criterion for the evaluation of the performance of a system. [2]
  - (b) Developments in human-computer interaction include natural, immersive and experiential interfaces.  
  
Describe, giving examples, the main characteristics of natural and immersive human computer interfaces. [6]
  
3. Use a truth table to prove De Morgan's Law  $\overline{A + B} = \bar{A} \cdot \bar{B}$ . [3]
  
4. (a) Simplify the following using De Morgan's Law's and Boolean identities. Identify which law or identity you are using: [3]
 
$$\overline{A \cdot B} + A$$
- (b) Simplify the following expression using Boolean identities and rules: [5]
 
$$A \cdot B \cdot (\bar{B} + C) + B \cdot C + B$$

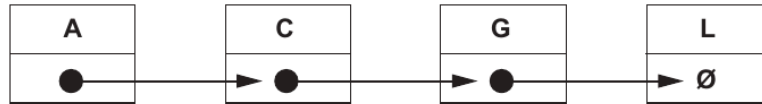
Answer **all** questions.

1. A binary tree structure is designed to contain strings and uses the following rules:
- The left pointer indicates the condition “earlier or at the same position in the alphabet”
  - The right pointer indicates the condition “later in the alphabet”
- (a) Construct a binary tree using these rules and the data entered in the following order:  
*Goat, Duck, Fox, Bear, Ant, Cat, Leopard, Owl, Mayfly, Insect, Jaguar, Emu.*  
You may use the initial letters if you wish. [2]
- (b) Carry out a pre-order traversal of the tree and give **one** use of pre-order traversals. [2]
- (c) Carry out an in-order traversal of the tree and give **one** use of in-order traversals. [2]
- (d) Carry out a post-order traversal of the tree and give **one** use of post-order traversals. [2]
2. The evaluation of a computer based solution should consider system functionality and system performance.
- (a) Identify a criterion for the evaluation of the functionality of a system and a criterion for the evaluation of the performance of a system. [2]
- (b) Developments in human-computer interaction include natural, immersive and experiential interfaces.  
Describe, giving examples, the main characteristics of natural and immersive human computer interfaces. [6]
3. Use a truth table to prove De Morgan's Law  $\overline{A + B} = \bar{A} \cdot \bar{B}$ . [3]
4. (a) Simplify the following using De Morgan's Law's and Boolean identities.  
Identify which law or identity you are using: [3]  
$$\overline{A \cdot B} + A$$
- (b) Simplify the following expression using Boolean identities and rules: [5]  
$$A \cdot B \cdot (\bar{B} + C) + B \cdot C + B$$

Answer all questions.

1. (a) Explain the differences between stack and queue data structures. [4]

(b) This is a diagram of a linked list in alphabetical order.



- (i) Redraw the linked list after the data item 'E' has been added. [2]
- (ii) Redraw the amended linked list after the data item 'C' has been deleted. [2]

2. Clearly showing each step, simplify the following Boolean expressions using Boolean algebra, identities and De Morgan's Law.

(a)  $A\bar{A} + A.B + A\bar{B} + B\bar{B}$  [5]

(b)  $(\bar{A}\bar{B}) + A.C + B$  [5]

3. This is an eight-bit number:

$01101001_2$

Include this number in a worked example to demonstrate how masking can be used to determine the state of the most significant bit. [3]

4. (a) Describe the term natural language interface. [2]

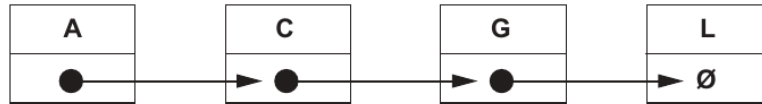
(b) Using an example, describe a potential use for natural language interfaces. [2]

(c) Describe the potential problems that can be associated with natural language interfaces. [3]

Answer all questions.

1. (a) Explain the differences between stack and queue data structures. [4]

- (b) This is a diagram of a linked list in alphabetical order.



- (i) Redraw the linked list after the data item 'E' has been added. [2]
- (ii) Redraw the amended linked list after the data item 'C' has been deleted. [2]
2. Clearly showing each step, simplify the following Boolean expressions using Boolean algebra, identities and De Morgan's Law.

(a)  $A\bar{A} + A.B + A\bar{B} + B\bar{B}$  [5]

(b)  $(\bar{A}.B) + A.C + B$  [5]

3. This is an eight-bit number:

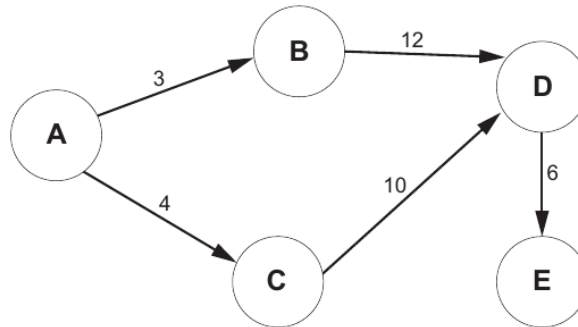
$$01101001_2$$

Include this number in a worked example to demonstrate how masking can be used to determine the state of the most significant bit. [3]

4. (a) Describe the term natural language interface. [2]
- (b) Using an example, describe a potential use for natural language interfaces. [2]
- (c) Describe the potential problems that can be associated with natural language interfaces. [3]

6. (a) Explain the purpose of a shortest path algorithm. [4]

(b) This is a diagram of the costs of traversing a network.



The traversal cost for each node is 2.

- (i) Show how this network and its traversal costs can be represented using a two dimensional array. [2]
- (ii) State the shortest path from node A to E and calculate its cost. [2]
7. (a) Explain what is meant by the term programming paradigm. [2]
- (b) Describe the difference between procedural and event-driven programming paradigms. [4]
- (c) Using examples, describe which programming paradigms would be most suitable when developing different types of software applications. [4]
8. Draw a truth table to prove the following: [4]

$$A \text{ NOR } B = \text{NOT } A \text{ AND NOT } B$$

Answer **all** questions.

1. (a) Explain, giving a suitable example, the operation of a stack data structure. [4]

- (b) This is a graphical representation of a two-dimensional array:

alphabet[ ]

	0	1	2	3	4	5	6
0	'a'	'b'	'c'	'd'	'e'	'f'	'g'
1	'h'	'i'	'j'	'k'	'l'	'm'	'n'
2	'o'	'p'	'q'	'r'	's'	't'	'u'
3	'v'	'w'	'x'	'y'	'z'	'.'	'.'

- (i) Describe how you would update the value '.' to '!' in the alphabet array. [2]
- (ii) Demonstrate how a two-dimensional array could be used to store pupil data in school. [2]
2. (a) Define the term algorithm. [2]
- (b) Describe one method of defining algorithms. [2]
- (c) Explain the benefits and drawbacks of recursive and non-recursive algorithms. [4]
3. Clearly showing each step, simplify the following Boolean expressions using Boolean algebra, identities and De Morgan's Law where appropriate.
- (a)  $(A + C).(A.C + A.\bar{C}) + A.C$  [6]
- (b)  $\bar{X}.\bar{Y} .(\bar{X} + Y) + \bar{Z}$  [5]
4. Write a Binary Search algorithm to search for a value in an array of characters and output its position. You may assume that the search value exists in the array. [8]

6. (a) Describe the approach to analysis and design in the following software development methodologies:
- (i) Waterfall [4]
  - (ii) Agile [4]
- (b) (i) Describe **one** piece of documentation that should be produced during the analysis stage of software development. [2]
- (ii) Describe **one** piece of documentation that should be used during the maintenance stage of software development [2]
7. Every book contains a unique 13-digit International Standard Book Number (ISBN). An ISBN comprises five parts: a GS1 assigned prefix, registration group, publisher, title and a check digit. Each individual part is separated with a space or hyphen.
- The GS1 assigned prefix must be 978 or 979.
  - The registration group must be a number between 01 and 99
  - The publisher must be a number between 00001 and 99999.
  - The title must be a number between 01 and 99.
  - The check digit must be a single digit.
  - A separator of each part which can be either a space ( ' ') or hyphen ('-').
- Example: 978-11-08412-72-8
- Produce a Backus-Naur Form (BNF) definition for a 13-digit ISBN. [6]
8. Functional programming and logic programming are both declarative programming paradigms. Explain these two paradigms, giving an example language in each case:
- (a) Functional programming [3]
  - (b) Logic programming [3]
9. All computer languages should follow the same standards.
- (a) Explain the need for standardisation of computer languages. [2]
  - (b) Describe two potential difficulties involved in agreeing these standards. [2]
10. Draw a truth table to prove the following:
- $$B \text{ AND NOT } (A \text{ NOR } B) = B \quad [4]$$

Answer **all** questions.

1. (a) Explain the operation of a hash table. [4]

- (b) This is a diagram of a hash table:

Key	Value
1001	Apple
1002	Berry
1003	Kiwi
1004	Lime
1005	Mango
1006	Pear
1007	Pineapple

- (i) Redraw the hash table after “1008, Orange” has been added and “1003, Kiwi” has been deleted. [2]
- (ii) Starting from the modified hash table resulting from (b)(i) redraw the hash table after “1006, Peach” has been added. [2]
2. Explain the differences between procedural and object orientated programming paradigms, giving suitable examples for each. [8]
3. Using the following Truth Table express  $p$  as a Boolean expression:

$A$	$B$	$C$	$p$
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	0

[4]

4. Clearly showing each step, simplify the following Boolean expressions using Boolean algebra, identities and De Morgan's Law where appropriate.

(a)  $A.(1+C) + \bar{B}.(A+B)$  [5]

(b)  $X.(\overline{Y+Z}) + \bar{Z}.X$  [5]

5. Write a quicksort algorithm, to sort an array of integers into ascending order. [8]

6. State the purpose of validation and verification, giving a suitable method for each. [4]

7. An online grocery store uses binary trees. These binary trees can be traversed using a variety of methods.

- (a) Describe the following methods of traversal and give an example of why each method would be used in the grocery store.

(i) In-order traversal [3]

(ii) Post-order traversal [3]

(iii) Pre-order traversal [3]

- (b) Draw an example of a balanced binary tree for grocery items. [1]

8. Giving suitable examples, describe the types of software tool that have been designed to assist in the following:

(a) Analysis and planning [3]

(b) Software development [3]

(c) Version management [3]

2. Computer languages such as HTML and CSS are standardised.
- (a) Describe the need for the standardisation of computer languages. [4]
- (b) Explain, giving a suitable example, the potential difficulties involved in agreeing and implementing standards. [4]
3. Clearly showing each step, simplify the following Boolean expressions using Boolean algebra, identities and De Morgan's Law where appropriate.
- (a)  $(1.X + \overline{Y.Y}).\overline{X} + X.Z$  [6]
- (b)  $(\overline{A}.A + A.A).1 + B.C.0$  [6]
4. Write an algorithm to search an unsorted one-dimensional array of strings and replace any duplicate values with the string "X" and then output the array. [8]

10. This is a signed 8-bit integer:

00001111<sub>2</sub>

Include this integer in a worked example to demonstrate how masking can be used to determine the sign of the integer.

[3]

11. This algorithm merges two one-dimensional arrays of length  $n$ . Assume the arrays (`myArray1` and `myArray2`) have already been populated with data.

`num(array)` returns the number of elements currently in the array.

```
1  declare i,j as integer
2  declare myArray1[n]
3  declare myArray2[n]
4  declare myArray3[]
5
6  set i = 0
7  set j = 0
8
9  for i = 0 to len(myArray1) - 1
10     set myArray3[num(myArray3)] = myArray1[i]
11 next i
12
13 for j = 0 to len(myArray2) - 1
14     set myArray3[num(myArray3)] = myArray2[j]
15 next j
16
17 output myArray3
```

- (a) Evaluate the efficiency of the algorithm and, using Big O notation, determine the growth rate for time performance. [5]
- (b) Determine the growth rate of memory space during a single run of the algorithm. [2]
- (c) Identify the type of time complexity and draw a graph of the algorithm to illustrate the order of time performance. Graph paper is not required. [2]
12. Explain, giving a suitable example, the shortest path algorithm. [4]
13. Discuss the potential use of natural language interfaces in human-computer communication to address the problem of communicating with computers.

You should draw on your knowledge, skills and understanding from a number of areas across your computer science course when answering this question. [12]

**END OF PAPER**

**END OF QUESTION PACK**

13 questions · 87 marks · ~2 h 10 min

Source: WJEC A2 Computer Science Unit 3 (1500U30-1), Summer 2017–2024, COVID gap  
Curated for WJEC Computer Science 2015 spec A2 Unit 3 – Topic 2 (3.2)

© WJEC CBAC Ltd. Pack layout © revise.wales for revision purposes only.