1.7 Boolean Algebra, Logic Gates and Karnaugh Maps: 10-minute Pack

Total Marks

OCR H446 · Calculators not allowed · about 10 minutes

Name: _____ Date: ____ / ____ / _____

Answer ALL questions. Show all your working.

1 This question is about NAND gates as a universal (functionally complete) logic gate.

(a) State what is meant by describing the NAND gate as 'functionally complete'. **(1)**

(b) Show how a single NAND gate can be used to produce the NOT function, describing the connection required. **(1)**

(c) Using only NAND gates, show how to construct an AND gate, describing the gates used and their connections. **(2)**

(d) A student claims an OR gate can also be built entirely from NAND gates, using De Morgan's law. Using Boolean algebra, show that NOT(NOT A . NOT B) is equivalent to A OR B, and hence describe how three NAND gates could be arranged to build an OR gate. **(4)**

(Total for Question 1 is 8 marks)

© Revision Library | revisionlibrary.co.uk

Mark scheme · 1.7 Boolean Algebra, Logic Gates and Karnaugh Maps: 10-minute Pack

Question 1

(a) any Boolean/logic function (e.g. NOT, AND, OR) can be built using NAND gates alone **B1**

(a) Answer: Any logic function can be constructed using only NAND gates.

(b) connect both inputs of the NAND gate to the same signal, A: output = NOT(A.A) = NOT A **B1**

(b) Answer: Tie both NAND inputs to A; output = NOT(A.A) = NOT A.

(c) the first NAND gate takes inputs A and B, giving output NOT(A.B) **B1**

(c) a second NAND gate, wired as a NOT gate (both inputs tied together), takes the output of the first gate, giving NOT(NOT(A.B)) = A.B **B1**

(c) Answer: NAND1(A,B) = NOT(A.B); NAND2 with both inputs = NAND1's output gives NOT(NOT(A.B)) = A.B.

(d) applies De Morgan's law: NOT(NOT A . NOT B) = NOT(NOT A) + NOT(NOT B) **M1**

(d) = A + B cso **A1**

(d) NAND gates 1 and 2, each wired as inverters (inputs tied together), produce NOT A and NOT B respectively **B1**

(d) NAND gate 3 takes NOT A and NOT B as its two inputs, giving NOT(NOT A . NOT B) = A + B **B1**

(d) Answer: NOT(NOT A . NOT B) = A + B; built from two inverter-wired NANDs producing NOT A and NOT B, feeding a third NAND.

© Revision Library | revisionlibrary.co.uk