

4.3 Evaluate Boolean expressions and fixed-width logic operations

Name: _____ Class: _____ Date: _____

Total: 24 marks

KEY WORDS Boolean algebra 布尔代数 • truth table 真值表 • logic gates 逻辑门

Objective

Evaluate Boolean expressions and fixed-width logic operations. Original GAC004-aligned practice, not an official assessment. Show the method as well as the result. All applied scenarios are invented classroom problems.

1 Key knowledge and worked examples

■ Key knowledge: logical truth and bit patterns need explicit conventions

Boolean algebra 布尔代数 uses true and false with AND, OR and NOT. Here OR is inclusive: true when either or both inputs are true. A **truth table** 真值表 evaluates every input combination; a two-input table has four rows. **Logic gates** 逻辑门 implement stated operations in a circuit.

Use brackets to make expression grouping explicit. De Morgan's identities are $\neg(A \wedge B) = (\neg A) \vee (\neg B)$ and $\neg(A \vee B) = (\neg A) \wedge (\neg B)$. Bitwise operations apply separately to each matching bit position. This sheet's bit patterns are unsigned unless stated otherwise, and NOT in a pattern inverts only the specified width.

■ Worked example: an OR output is inverted by NOT



Known: $Y = \neg(A \vee B)$. Inclusive OR is false only when both inputs are false; NOT reverses that result. In row order $(A, B) = (0, 0), (0, 1), (1, 0), (1, 1)$, the output column is 1, 0, 0, 0. De Morgan gives equivalent expression $(\neg A) \wedge (\neg B)$.

2 Practice

2.1 For A true and B false, find A AND B and A OR B, using inclusive OR. [2]

2.2 Invert the four-bit pattern 1010 using bitwise NOT, and find its unsigned decimal value. [2]

2.3 List the outputs of $A \wedge \neg B$ in row order (0,0), (0,1), (1,0), (1,1). State which input condition makes the output true. [3]

2.4 Use De Morgan to rewrite $\neg(A \wedge B)$. Evaluate both forms at A true and B true. [3]

3 Exam-style questions

3.1 For four-bit patterns $X = 1100$ and $Y = 1010$, calculate (a) bitwise AND, (b) bitwise inclusive OR, and (c) bitwise NOT of X . Give each result also as an unsigned decimal value. [6]
