

Solutions

2.1

A AND B is false because both inputs are not true. A OR B is true because at least one input is true.

2.2

The inverted four-bit pattern is 0101. Unsigned value is 5. The width remains four bits.

2.3

In order the outputs are 0, 0, 1, 0. Applying NOT before AND earns M1; correct output column earns A1. Output is true exactly when A is true and B false.

2.4

Equivalent form $(\neg A) \vee (\neg B)$. Original form is NOT true, hence false. Rewritten form is false OR false, also false. A single row check alone is not a proof of full equivalence.

3.1

AND gives 1000. Decimal 8. OR gives 1110. Decimal 14. Four-bit NOT of X gives 0011. Decimal 3. NOT does not create an unrestricted or signed value here.

3.2

(a) Use row order 00, 01, 10, 11. Column NOT B is 1, 0, 1, 0. Column A AND B is 0, 0, 0, 1. Column A AND NOT B is 0, 0, 1, 0. Final Y is 0, 0, 1, 1.

(b) Factor A using distributivity.

$$Y = A \wedge (B \vee \neg B)$$

The bracket is always true, so $Y = A$. This agrees with the A column in every truth-table row, proving Boolean equivalence.

(c) The truth table tests this abstract rule only; actual authentication, implementation and policy are not established by its simplification.