Question 1 3 Pts F = (AB)'(A+B+C)' Is Equivalent To: O C O A'+B' O A'B'C' O A'+B'+C' O None Of The Above

Question 1 3 Pts F = (AB)'(A+B+C)' Is Equivalent To: O C O A'+B' O A'B'C' O A'+B'+C' O None Of The Above

---

Introduction

Boolean algebra is a fundamental aspect of digital logic design, enabling engineers and students to simplify logical expressions for efficient circuit implementation. In this article, we will analyze the Boolean function:

F = (AB)'(A + B + C)'

and determine which of the given options it is equivalent to:

- O C
- O A' + B'
- O A'B'C'
- O A' + B' + C'
- O None Of The Above

Understanding this problem involves applying Boolean algebra rules such as De Morgan's laws, distribution, and simplification techniques. Let's explore how to evaluate this expression step-by-step and identify the correct equivalent.

Understanding the Boolean Expression

The Given Expression

The function is:

F = (AB)'(A + B + C)'

This expression involves two complemented (NOTed) terms multiplied together (AND operation). To simplify, we'll focus on applying De Morgan's laws to each complemented term.

Components Breakdown

- (AB)': The complement of the product of A and B.
- (A + B + C)': The complement of the sum of A, B, and C.

By simplifying each, we can then perform the AND operation to find the overall equivalent expression.

Applying De Morgan's Laws

De Morgan's Law for (AB)'

(AB)' = A' + B'

- When complementing a product, the law states: (XY)' = X' + Y'
- Here, (AB)' simplifies to A' + B'

De Morgan's Law for (A + B + C)'

(A + B + C)' = A'B'C'

- When complementing a sum of multiple variables, the law states: (X + Y + Z)' = X'Y'Z'
- Here, (A + B + C)' simplifies to A' B' C'

Simplifying the Expression

Now, rewrite the original function with these simplified forms:

F = (A' + B') (A' B' C')

This is an AND operation between A' + B' and A' B' C'.

Step 1: Recognize the common factors

Since the second term is A' B' C', and the first term is A' + B', we analyze how their AND operation simplifies.

Step 2: Use distribution

Express the AND of these two terms:

F = (A' + B') A' B' C'

We can distribute:

F = [(A' + B') A' B' C'] = (A' + B') A' B' C'

But it's easier to see that A' B' C' is a subset of A' + B' because:

- A' B' C' is true only when A', B', and C' are all true.
- A' + B' is true when either A' or B' is true.

Therefore, the AND operation with A' + B' and A' B' C' simplifies to A' B' C' because:

- When A' B' C' is true, both A' + B' and A' B' C' are true.
- When A' + B' is true but A' B' C' is false, the overall expression is false.

The key insight is that A' B' C' is a subset of A' + B', so:

F = A' B' C'

Final Simplification and Matching with Options

The simplified form of the original Boolean function is:

F = A' B' C'

Now, let's compare this with the options provided:

- O C: Just C (incorrect)
- O A' + B': Sum of complements (incorrect)
- O A'B'C': Product of complements (correct)
- O A' + B' + C': Sum of complements (incorrect)
- O None Of The Above

Since A' B' C' matches the third option (A' B' C'), the correct answer is:

O A' B' C'

Significance in Digital Logic Design

Understanding how to simplify Boolean expressions like F = (AB)'(A + B + C)' is crucial in designing efficient digital circuits. Simplification reduces the number of logic gates needed, which in turn minimizes cost, power consumption, and complexity.

Summary of Key Steps

1. Apply De Morgan's Laws:

- (AB)' = A' + B'
- (A + B + C)' = A' B' C'

2. Rewrite the expression:

- F = (A' + B') (A' B' C')

3. Simplify the product:

- Recognize that A' B' C' is a subset of A' + B', so

- F = A' B' C'

4. Compare with options:

- The matching option is A' B' C'

Conclusion

By applying Boolean algebra principles, we have determined that the expression F = (AB)'(A + B + C)' simplifies to A' B' C'. Therefore, the correct choice among the options is:

O A' B' C'

This exercise highlights the importance of mastering Boolean simplification techniques for efficient digital circuit design and optimization.

Additional Resources

- Boolean Algebra Rules and Laws: Essential for simplifying logical expressions.
- De Morgan's Theorems: Key tools for converting ANDs to ORs and vice versa.
- Digital Logic Design Textbooks: For in-depth understanding of logic circuit optimization.
- Online Boolean Algebra Calculators: Useful for verifying complex expressions.

FAQs

Q1: Why is it important to simplify Boolean expressions?

A1: Simplification reduces the number of logic gates needed, making circuits more efficient, cost-effective, and easier to troubleshoot.

Q2: Can all Boolean expressions be simplified to a single term?

A2: Not necessarily. Some expressions can be simplified significantly, while others may have multiple minimal forms. The goal is to find the simplest equivalent expression.

Q3: What are common mistakes when simplifying Boolean expressions?

A3: Common mistakes include incorrect application of De Morgan's laws, neglecting to simplify fully, or misinterpreting the subset relationships between terms.

In summary, understanding Boolean algebra and the application of De Morgan's laws allows for effective simplification of complex logical expressions, essential in the realm of digital electronics and logic circuit design.

Frequently Asked Questions

What is the simplified form of the Boolean expression F = (AB)'(A+B+C)'?

The correct simplified form is A' + B' + C'.

Which of the following options correctly represents the equivalent of F = (AB)'(A+B+C)'?

A' + B' + C' is the correct equivalent.

Is the expression (AB)'(A+B+C)' equivalent to A' + B' + C'?

Yes, (AB)'(A+B+C)' simplifies to A' + B' + C'.

What Boolean law is primarily used to simplify F = (AB)'(A+B+C)' to A' + B' + C'?

De Morgan's Theorem is used to simplify the expressions.

Among the options provided, which one correctly states the equivalence of F = (AB)'(A+B+C)'?

A' + B' + C' is the correct answer.

Additional Resources

Boolean Algebra Expression Simplification: An In-Depth Analysis of F = (AB)'(A+B+C)'

Introduction

In the realm of digital logic design and Boolean algebra, simplifying complex expressions is fundamental to optimizing circuits for efficiency, cost, and performance. The question at hand — "F = (AB)'(A+B+C)' Is Equivalent To:" — is a classic example of Boolean expression simplification that often appears in exams, circuit design, and theoretical analysis. This article explores this expression in meticulous detail, providing a step-by-step breakdown, fundamental concepts, and final conclusions, all while maintaining an engaging, expert-level review style.

Understanding the Components of the Expression

Before diving into simplification, it’s essential to understand each part of the Boolean expression:

- $(AB)'$: The complement (NOT) of the conjunction (AND) of A and B.
- $(A+B+C)'$: The complement (NOT) of the disjunction (OR) of A, B, and C.
- $F = (AB)'(A+B+C)'$: The product (AND) of the two complemented expressions.

This expression involves basic Boolean operations: AND, OR, and NOT, which are the building blocks of digital logic. The goal is to simplify this expression into a form that is easier to interpret, implement, or analyze.

Step-by-Step Simplification Process

Step 1: Apply De Morgan’s Theorem to (AB)'

De Morgan's Law states:

$$
(AB)' = A' + B'
$$

Interpretation: The complement of an AND operation equals the OR of the complements.

Result after Step 1:

$$
F = (A' + B')(A+B+C)'
$$

Step 2: Simplify (A+B+C)'

Applying De Morgan’s Law again:

$$
(A+B+C)' = A'B'C'
$$

Result after Step 2:

$$
F = (A' + B') \times A' B' C'
$$

\]

Step 3: Expand the Expression

Using distribution:

$$
F = (A' + B') \times A' B' C' = (A' \times A' B' C') + (B' \times A' B' C')
$$

Since Boolean algebra is idempotent ($A' \times A' = A'$), the expression simplifies further:

$$
F = A' B' C' + B' A' B' C'
$$

Notice that in the second term, $B' \times B' = B'$, so:

$$
F = A' B' C' + A' B' C'
$$

Adding identical terms:

$$
F = A' B' C'
$$

Final Simplified Form

The simplified expression for F is:

$$
\boxed{A' B' C'}
$$

This is a minimal product term representing the AND of the complements of A, B, and C.

Analyzing the Multiple-Choice Options

Given the options:

- A. $A' + B'$
- B. $A' B' C'$

- C. $A' + B' + C'$
- D. None of the Above

Our derived expression matches option B: $A' B' C'$.

Expert Commentary on the Simplification

This problem illustrates the elegance of Boolean algebra rules, especially De Morgan’s laws, which enable transforming complex expressions into simpler, more manageable forms. Recognizing the application of these laws early on is crucial for efficient simplification.

The process underscores a key principle: complementing ANDs and ORs using De Morgan’s laws often reveals straightforward minimal forms. It also demonstrates the importance of understanding how to manipulate Boolean expressions algebraically before implementing them in logic circuits.

Practical Implications in Digital Circuit Design

Understanding how to simplify Boolean expressions directly impacts digital system design:

- Reduced Gate Count: Simplified expressions lead to fewer logic gates, reducing circuit complexity.
- Lower Power Consumption: Fewer gates translate into less power draw, a critical factor in portable devices.
- Enhanced Speed: Simpler logic circuits typically operate faster due to reduced propagation delay.
- Cost Efficiency: Fewer components mean lower manufacturing costs.

In this context, recognizing that F = A' B' C' allows engineers to implement a minimal three-input AND gate with inverted inputs, possibly using NAND gates for cost-effective design.

Additional Considerations

While the algebraic approach provides a clear pathway, engineers often verify simplifications via truth tables or Karnaugh maps to ensure correctness, especially for more complex expressions.

Truth Table for $A' B' C'$:

A	B	C	A'	B'	C'	F = A' B' C'
0	0	0	1	1	1	1
0	0	1	1	1	0	0

0	1	0	1	0	1	0
0	1	1	1	0	0	0
1	0	0	0	1	1	0
1	0	1	0	1	0	0
1	1	0	0	0	1	0
1	1	1	0	0	0	0

The only case where F=1 is when all inputs are zero, confirming the minimal expression.

Conclusion

F = (AB)'(A+B+C)' simplifies elegantly to A' B' C' through systematic application of Boolean algebra principles. Recognizing the core laws at play allows for rapid and accurate simplification, leading to efficient digital circuit implementations.

In the context of the given multiple-choice options, the correct answer is B. \(A' B' C'\).

This problem exemplifies the importance of mastering Boolean algebra for anyone involved in digital logic design, whether students, engineers, or hobbyists. Through understanding and applying these fundamental laws, one can optimize circuits, reduce costs, and improve overall system performance.

Final Thoughts

Boolean algebra is not just an abstract mathematical discipline but a practical toolkit that directly influences modern electronic systems. Mastery of these simplification techniques empowers designers to craft smarter, faster, and more economical digital solutions.

Question 1 3 Pts F Ab A B C Is Equivalent To O C O A B O A B C O A B C O None Of The Above

Question 1 3 Pts F Ab A B C Is Equivalent To O C O A B O A B C O A B C O None Of The Above

Related Articles

- Question 1 Of 10How Was Production Different After The Industrial Revolution Than It Wasbefore?A. Before
- In A Collection Of 50 Cards, Sal Has Some Valuable And Some Regular Cards. He Has 16 More Valuable Cards
- Please See The Class Below And Explain Why Each Line Of Code Is Correct Or Incorrect. Write Your Answers