**Assignment Activity Unit 3**

Kelechi Nwa-uwa

CS 1111-01

Wednesday, September 24, 2025

**Number 1:** Simplify (I ᐧ A) + (L’ ᐧ A).

First, expand the factorized expression

I + L’ ᐧ A + L’ ᐧ I + A ᐧ A + A

Using the Idempotent Law, we can simplify the last term to be just A

I + L’ ᐧ A + L’ ᐧ I + A ᐧ A

We can then factorize the last 3 terms

I + L’ ᐧ A (1 + L’ ᐧ I + 1 ᐧ 1)

Using both the Annulment and Identity Laws, we can simplify the expression in the brackets *x* + 1 === 1; and *x* ᐧ 1 === *x*

I + L’ ᐧ A ᐧ (1 ᐧ 1 ᐧ 1)

I + L’ ᐧ A ᐧ (1)

I + L’ ᐧ A

Commutative law allows us to flip this expression, taking I + L’ as unified term

A ᐧ I + L’

or A ᐧ (I + L’)

**Number 2**: Significance of De Morgan’s Theorem, applying it to L’ ᐧ A.

De Morgan’s theorems show the relationship between complementary and equivalent logical expressions. The theorems state that the complement of a sum is the product of its individual complements; and the complement of a product is the sum of its individual complements (Read Electric Vehicle, 2024).

This means that to find the complement of a boolean expression, you must change each logical sign to its complementary sign (OR to AND and vice versa); and complement the terms of the expression (1 to 0 and vice versa).

To apply the second theory to this case, we rewrite L’ ᐧ A, bearing in mind the law of double negation:

L’ ᐧ A = ((L’ ᐧ A)’)’

Next, we expand the inner bracket:

L’ ᐧ A = ((L’)’ + A’)’

And finally, we have our equivalent expression:

L’ ᐧ A = (L + A’)’

Boolean laws show how boolean values behave in a logic function, thereby aiding the analysis and simplification of boolean expressions. Applying (and even the proofs of) De Morgan’s theorems may make use of various boolean laws, such as commutative, identity, annulment, complement, or distributive laws. This allows for an efficient transformation of boolean expressions, effectively reducing the number of gates required for compound boolean statements. (Ndjountche, 2016)

De Morgan’s theorems are the groundwork for the NAND and NOR gates. The first theorem shows that the NAND gate is equivalent to an inversion followed by an OR gate [(A ᐧ B)’ = A’ + B’]; while the second theorem shows that the NOR gate is equivalent to an inversion followed by an AND gate [(A + B)’ = A’ ᐧ B’]. These gates are considered universal gates because any logic function can be implemented with just NAND or NOR gates. (Ndjountche, 2016; Virtual Labs, n.d.)

In a practical sense, being able to perform complex logic functions reduces the cost of hardware and improves efficiency (Virtual Labs, n.d.).**Number 3a:** Logic gate diagram of A ᐧ (I + L’) using AND, OR, and NOT gates.



**Number 3b**: Truth table for A ᐧ (I + L’), including all combinations of A, I, and L.

| A | I | L | L’ | I + L’ | A ᐧ (I + L’) |
| --- | --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 1 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 1 | 1 |

**Number 4**: Comparison of the original and simplified expressions, using truth tables.

(I ᐧ A) + (L’ ᐧ A) A ᐧ (I + L’)

| A | I | L | L’ | I + L’ | A ᐧ (I + L’) |
| --- | --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 1 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 1 | 1 |

| A | I | L | L’ | I ᐧ A | L’ ᐧ A | (I ᐧ A) + (L’ ᐧ A) |
| --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 1 | 0 | 1 |

⇒ 0,0,0,0,1,1,0,1 ⇒ 0,0,0,0,1,1,0,1

For each combination of A, I, and L, the expressions *A ᐧ (I + L’)* and *(I ᐧ A) + (L’ ᐧ A)* return the same outputs, as illustrated by the truth tables above.

**References**

Ndjountche, T. (2016). *Digital electronics 1: Combinational logic circuits*. John Wiley & Sons, Incorporated.

Read Electric Vehicle. (2024, April 24). *Boolean algebra/Boolean laws/De Morgan’s Theorem* [Video]. YouTube <https://youtu.be/kNHLRoE8qNI>

Virtual Labs. (n.d.). *De Morgan’s theorems — Theory*. ADE-IITR vLabs. Retrieved September 24, 2025, from<https://ade-iitr.vlabs.ac.in/exp/de-morgans-theorems/theory.html>