**DAILY ASSESSMENT FORMAT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **28-05-2020** | **Name:** | **Varun G Shetty** |
| **Course:** | **Logic design** | **USN:** | **4AL17EC093** |
| **Topic:** | **Boolean equations for digital circuits. Combinational circuits: Conversion of MUX** | **Semester & Section:** | **6th & ‘B’** |
| **GitHub Repository:** | **Varunshetty4** |  |  |

|  |
| --- |
| **FORENOON SESSION DETAILS** |
| **Image of session**  **1.**    **2.** |
| **Report – Report can be typed or hand written for up to two pages.**   1. **Boolean equations for digital circuits.** 2. **Combinational circuits:**  * **Conversion of Mux**   **Boolean equations for digital circuits:** Boolean algebra is used to simplify Boolean expressions which represent combinational logic circuits. It reduces the original expression tequivalent expression that has fewer terms which means that less logic gates are needed to implement the combinational logic circuit. Laws of Boolean Algebra Boolean Algebra Laws are used to simplify boolean expressions. Basic Boolean Laws  1. Idempotent Law    * A \* A = A    * A + A = A 2. Associative Law    * (A \* B) \* C = A \* (B \* C)    * (A + B) + C = A + (B + C) 3. Commutative Law    * A \* B = B \* A    * A + B = B + A 4. Distributive Law    * A \* (B + C) = A \* B + A \* C    * A + (B \* C) = (A + B) \* (A + C) 5. Identity Law    * A \* 0 = 0     A \* 1 = A    * A + 1 = 1     A + 0 = A 6. Complement Law    * A \* ~A = 0    * A + ~A = 1 7. Involution Law    * ~(~A) = A 8. DeMorgan's Law    * ~(A \* B) = ~A + ~B    * ~(A + B) = ~A \* ~B  Redundancy Laws  1. Absorption    * A + (A \* B) = A    * A \* (A + B) = A 2. * (A \* B) + (A \* ~B) = A    * (A + B) \* (A + ~B) = A 3. * A + (~A \* B) = A + B    * A \* (~A + B) = A \* B   Each law is described by two parts that are duals of each other. The Principle of duality is   * Interchanging the + (OR) and \* (AND) operations of the expression. * Interchanging the 0 and 1 elements of the expression. * Not changing the form of the variables.  **DeMorgan’s Theorem** This theorem is useful in finding the **complement of Boolean function**. It states that the complement of logical OR of at least two Boolean variables is equal to the logical AND of each complemented variable.  DeMorgan’s theorem with 2 Boolean variables x and y can be represented as  x+yx+y’ = x’.y’  The dual of the above Boolean function is  x.yx.y’ = x’ + y’  Therefore, the complement of logical AND of two Boolean variables is equal to the logical OR of each complemented variable. Similarly, we can apply DeMorgan’s theorem for more than 2 Boolean variables also. Application of Boolean Algebra: [Combinational Logic Circuit Design](http://electronics-course.com/combinational-logic-design) comprises the following steps   1. From the design specification, obtain the truth table 2. From the truth table, derive the [Sum of Products](http://electronics-course.com/sum-of-products) Boolean Expression. 3. Use Boolean Algebra to simplify the boolean expression. The simpler the boolean expression, the less logic gates will be used. 4. Use [logic gates](http://electronics-course.com/logic-gates) to implement the simplified Boolean Expression.   **MUX:** In electronics, a multiplexer, also known as a data selector, is a device that selects between several analog or digital input signals and forwards it to a single output line. A multiplexer of inputs has select lines, which are used to select which input line to send to the output. |