**Experiment 2 DEC LAB**

**22.09.2020**

**Aim** :- To solve various expressions and represent them using minimum number of gates using Tinkercad software.

**Apparatus required** :-

1) Tinkercad Software for designing circuits

2) Breadboard

3) Power Supply

4) Slideswitch

5) IC of Logic gates

6) Resistor

7) Led bulb for signal

8) Connecting Wires

**Theory**:- Boolean expressions are logical expressions that are either true or false. They are associated with Boolean Algebra and Laws which are used to analyse gates and circuits. A set of rules or Laws of Boolean Algebra expressions have been invented to help reduce the number of logic gates needed to perform a particular logic operation resulting in a list of functions or theorems known commonly as the **Laws of Boolean Algebra**.

**Boolean Algebra** is the mathematics we use to analyse digital gates and circuits. We can use these “Laws of Boolean” to both reduce and simplify a complex Boolean expression in an attempt to reduce the number of logic gates required. Boolean Algebra is therefore a system of mathematics based on logic that has its own set of rules or laws which are used to define and reduce Boolean expressions.

Another way to solve Boolean Expressions is by the use of K map or Karnaugh Map. K map is a special arrangement of truth table. In a K map, various terms of a complex expression are placed in cells of the K map and the cells are grouped to eliminate unwanted variables. This in turn gives a simplified expression , which gives the same output as the complex expression.

Don’t care condition is the condition in which for any input the output doesn’t matter. They are also used in grouping the cells as well.

**Observation :-**

1. F(P,Q,R,S)= ∑0,2,5,7,8,10,13,15 with 2,7,8,13 as don’t care condition

The K Map for above expression is

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PQ RS | R’S’ | R’S | RS | RS’ |
| P’Q’ | 1 |  |  | X |
| P’Q |  | 1 | X |  |
| PQ |  | X | 1 |  |
| PQ’ | X |  |  | 1 |

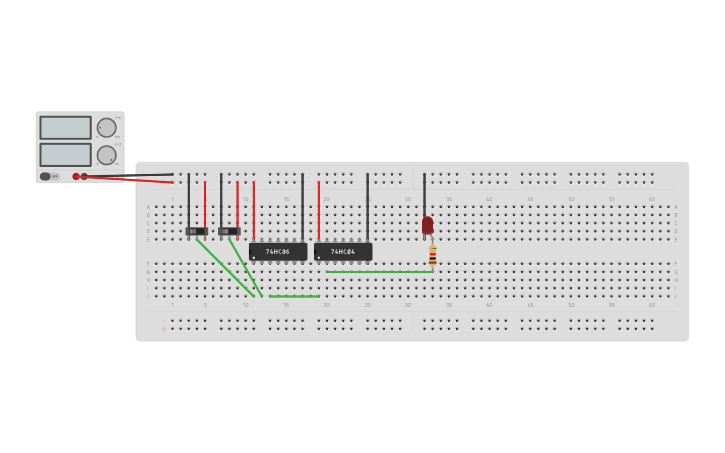
On Simplifying, we get the expression as QS+Q’S’

The truth table for this expression is

|  |  |  |
| --- | --- | --- |
| Q | S | Output |
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

This has output similar to XNOR gate.

The circuit of given expression is



1. F(A,B,C,D)= ABC’D’+ABC’D+AB’C’D+ABCD+AB’CD+ABCD’+AB’CD’

The K Map for above expression is

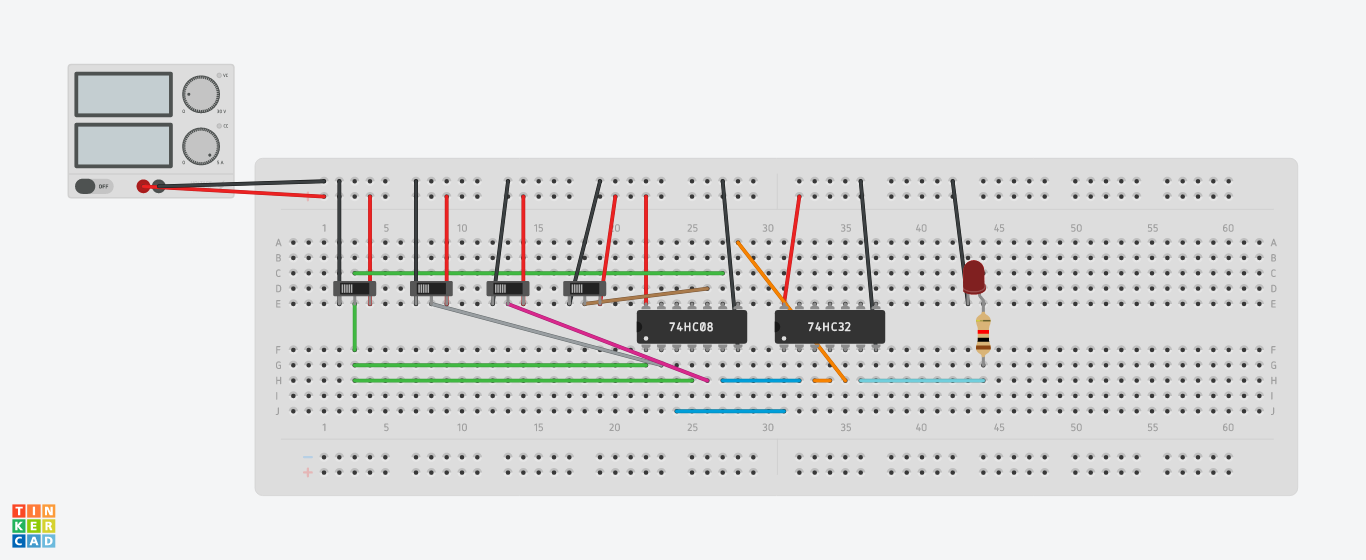
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| AB CD | C’D’ | C’D | CD | CD’ |
| A’B’ |  |  |  |  |
| A’B |  |  |  |  |
| AB | 1 | 1 | 1 | 1 |
| AB’ |  | 1 | 1 | 1 |

On Simplifying, we get the expression as AB+AC+AD

The truth table for this expression is

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A | B | C | D | Output |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |

The circuit of given expression is



1. F(A,B,C,)= A’B’C’+A’BC’+A’BC+ABC’

The K Map for above expression is

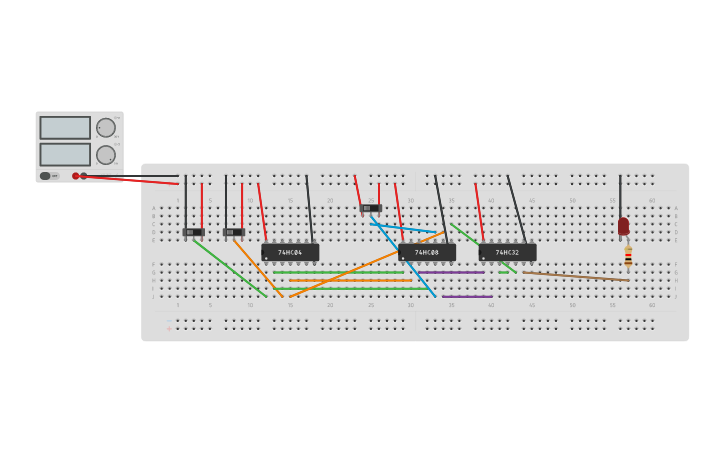
|  |  |  |
| --- | --- | --- |
| AB C | C’ | C |
| A’B’ | 1 |  |
| A’B | 1 | 1 |
| AB | 1 |  |
| AB’ |  |  |

On Simplifying, we get the expression as BC’+A’C’+A’B

The truth table for this expression is

|  |  |  |  |
| --- | --- | --- | --- |
| A | B | C | Output |
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 |
| 0­ | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 |

The circuit of given expression is



**Conclusion :-** Hence all the expressions were simplified by using K map and represented in the circuit.

Name: Harsh Kumar

Branch: CE-3rd Semester

ID- B219027