Experiment Number 01

Experiment Name: Implementing circuit with basic gates.

Theory: Digital circuits are built using basic logic gates like NOT, AND, and OR. These gates form the foundation of digital electronics, as they allow for the implementation of complex logical functions. Here's how each gate contributes to circuit design:

NOT Gate: The NOT gate, also called an inverter, outputs the opposite of its input. For example,74LS04.

AND Gate: The AND gate outputs 1 only if all inputs are 1. For example,74LS08.

OR Gate: The OR gate outputs 1 if at least one input is 1. For example,74LS32.

Apparatus: Bread Board, Jumper cable, power Supply,74LS04,74LS08,74LS32, Led light.

Boolean Function: In our experiment, we are implementing the logic function $F=x'y'z+x'yz+xy'$ using basic logic gates.

BY simplifying it: $F=x'y'z+x'yz+xy'$

$$F=x'z(y+y') +xy'$$

$$F=x'z +xy'$$

$F=x'z +xy'$ is our simplified version.

Truth Table: The function $F=x'z+xy'$ represents a logical OR operation between two terms:

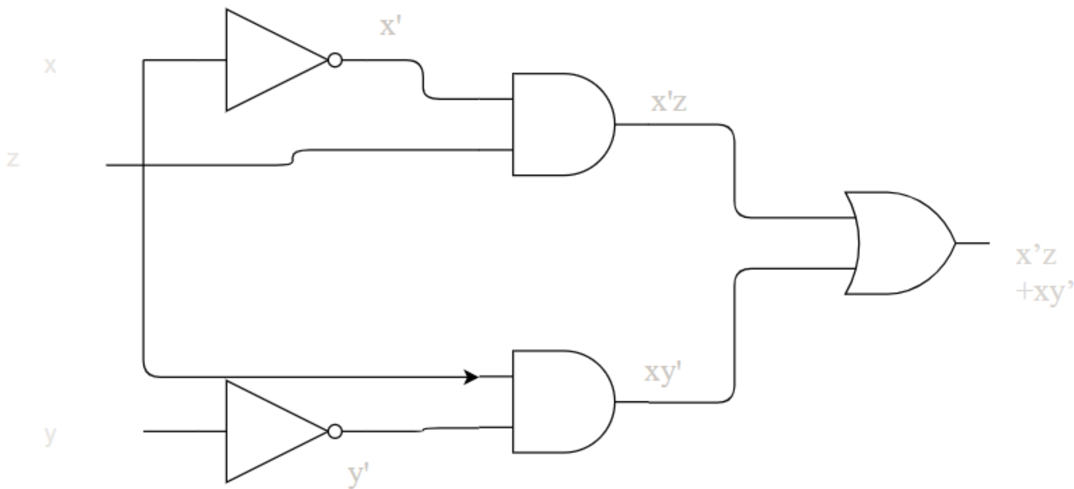
1. $x'z$: True when $x=0$ and $z=1$.
2. xy'': True when $x=1$ and $y=0$.

The truth table evaluates FF by checking when either of these terms is true, combining their outputs to produce the result. This function highlights specific input conditions where FF is true, depending on the values of x, y, z.

x	y	z	x’	y’	x’z	xy'	F
0	0	0	1	1	0	0	0
0	0	1	1	1	1	0	1
0	1	0	1	0	0	0	0
0	1	1	1	0	1	0	1
1	0	0	0	1	0	1	1
1	0	1	0	1	0	1	1
1	1	0	0	0	0	0	0
1	1	1	0	0	0	0	0

Logic Diagram: The logic function $F=x'z+xy'$ can be implemented using a logic diagram as follows:

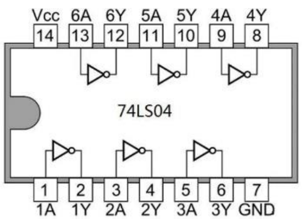
1. $x'z$:
   - Use a NOT gate to invert x, producing x' .
   - Use an AND gate to combine x' and z.
2. xy':
   - Use a NOT gate to invert y, producing y'.
   - Use an AND gate to combine x and y'.
3. Final Output:
   - Use an OR gate to combine the outputs of the two AND gates, producing F.

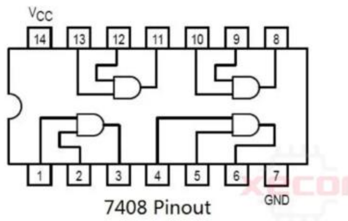


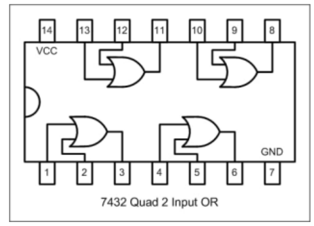
[(Software link)]

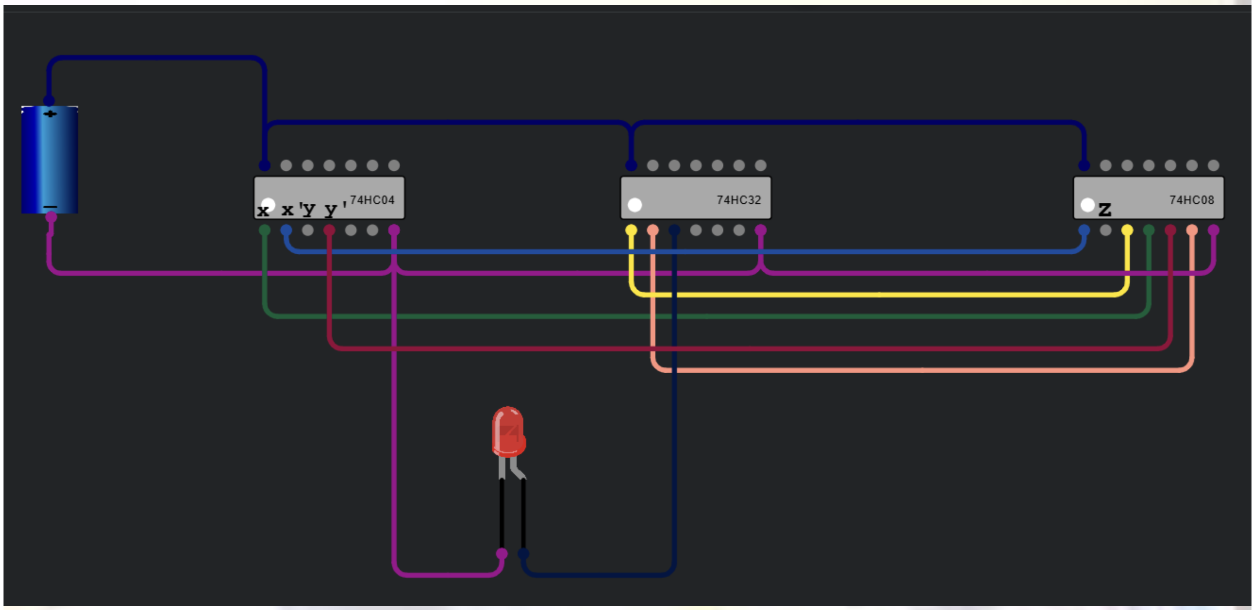
Pin Diagram: For the pin diagram implementation of F=x′z+xy′the connections are as follows:

1. Inputs:
   - Connect x, y, and z to the input pins of the NOT and AND gates.
2. Gates:
   - Use a NOT gate for xxx to generate x′, and another for y to generate y′.
   - Connect x′ and z to an AND gate for x′z.
   - Connect x and y′ to another AND gate for xy′.
3. Final Output:
   - Use an OR gate to combine the outputs of the two AND gates, producing F at the output pin.









[(Software link)]

<u>Observed Output:</u>

x	y	z	x'	y'	x'z	xy'	F
0	0	0	1	1	0	0	0
0	0	1	1	1	1	0	1
0	1	0	1	0	0	0	0
0	1	1	1	0	1	0	1
1	0	0	0	1	0	1	1
1	0	1	0	1	0	1	1
1	1	0	0	0	0	0	0
1	1	1	0	0	0	0	0

Conclusion: The function F=x′z+xy′ demonstrates how specific input conditions can be combined using basic logic gates (NOT, AND, and OR) to produce a desired output. It highlights the use of inversion and logical operations to implement a circuit, making it a fundamental example of how complex functions can be built from simple gates.