Group Leader : Shoaib Akhtar(20p-0147)

Group Members : Waqar Ahmed (20P-0750)

Project Name: Password Security System

Assigned By : Sir Shakir ullah Shah

Subject : Digital Logic Design (EE227)

Table of Contents:

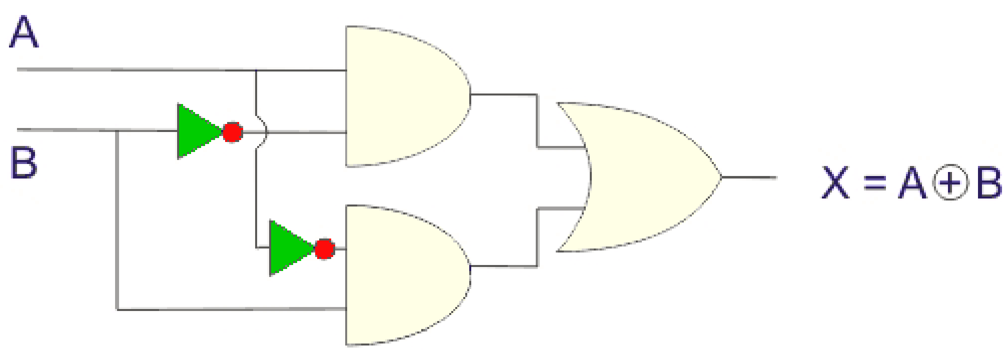
Components required:

- Switch (9 pcs)
- D Flip flop(4pcs)
- 2-input xor gate (4 pcs)
- Not Gate (4 pcs)
- And gate (3 pcs)
- Led bulb (2 pcs)`

XOR Gate:

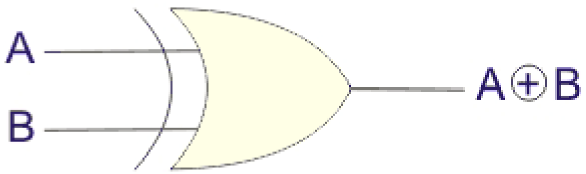
An XOR gate (also known as an EOR, or EXOR gate) – pronounced as Exclusive OR gate – is a digital logic gate that gives a true (i.e. a HIGH or 1) output when the number of true inputs is odd. An XOR gate implements an exclusive OR, i.e., a true output result occurs if one – and only one – of the gate's inputs is true. If both inputs are false (i.e. LOW or 0) or both inputs are true, the output is false.
XOR represents the inequality function, i.e. the output is true if the inputs are not alike; otherwise, the output is false. A common way to remember the XOR is "must have one or the other, but not both".
Another way to look at an XOR gate: a modulo sum of two variables in a binary system looks like this:

The logic gate performs this modulo sum operation without including carry is known as XOR gate. An XOR gate is normally two inputs logic gate where the output is only logical 1 when only one input is logical 1. When both inputs are equal, either are 1 or both are 0, the output will be logical 0.
This is the reason an XOR gate is also called an anti-coincidence gate or inequality detector. This gate is called XOR or exclusive OR gate because its output is only 1 when its input is exclusively 1.
In digital electronics, other logic gates include NOT gates, OR gates, NAND gates, and NOR Gates.



Logical Symbol of XOR Gate

An XOR gate is logically represented as,



XOR Gate Truth Table

Truth tables list the output of a particular digital logic circuit for all the possible combinations of its inputs. The truth table of an XOR gate is given below:

Inputs		Output
A	**B**	**X**
0	0	0
0	1	1
1	0	1
1	1	0

The above truth table's binary operation is known as exclusive OR operation. It is represented as A ⊕ B. The symbol of exclusive OR operation is represented by a plus ring surrounded by a circle ⊕.

Flip Flop:

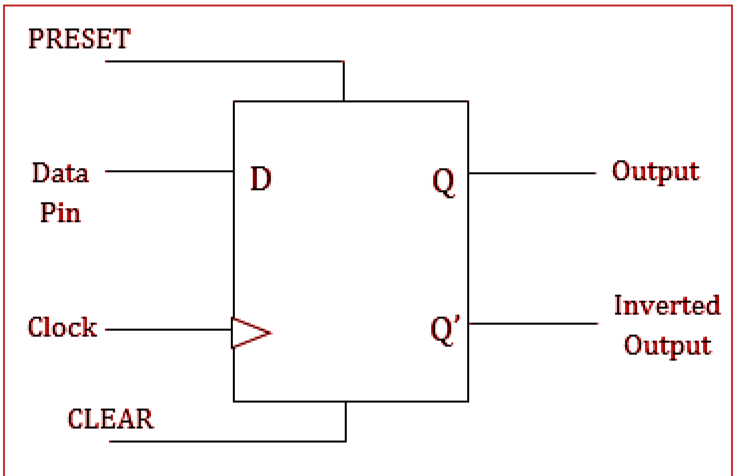
A Flip flop is an digital circuit with solid states that can be used to shop binary facts. The stored statistics may be changed by using applying varying inputs. Flip-flops and latches are essential building blocks of virtual electronics structures utilized in computers, communications, and plenty of different types of systems. Both are used as statistics garage elements.

D-Flip Flop:

D Flip-flops are used as part of reminiscence garage factors and records processors as nicely. D flip-flop can be built the usage of nand gate or with nor gate. Because of its versatility they are available as ic applications. The important packages of d flip-flop are to introduce delay in timing circuit, as a buffer, sampling information at unique periods. D

Flip-flop is easier in terms of wiring connection as compared to jk flip-flop. Here we're the usage of nand gates for demonstrating the d flip flop.

On every occasion the clock sign is low, the input is never going to affect the output state. The clock needs to be high for the inputs to get active. Accordingly, d flip-flop is a controlled bi-strong latch where the clock sign is the manipulate signal. Again, this gets divided into fine side induced d Flip flop and bad aspect brought about d flip-flop



Symbol: D Flip-flop

Truth Table :

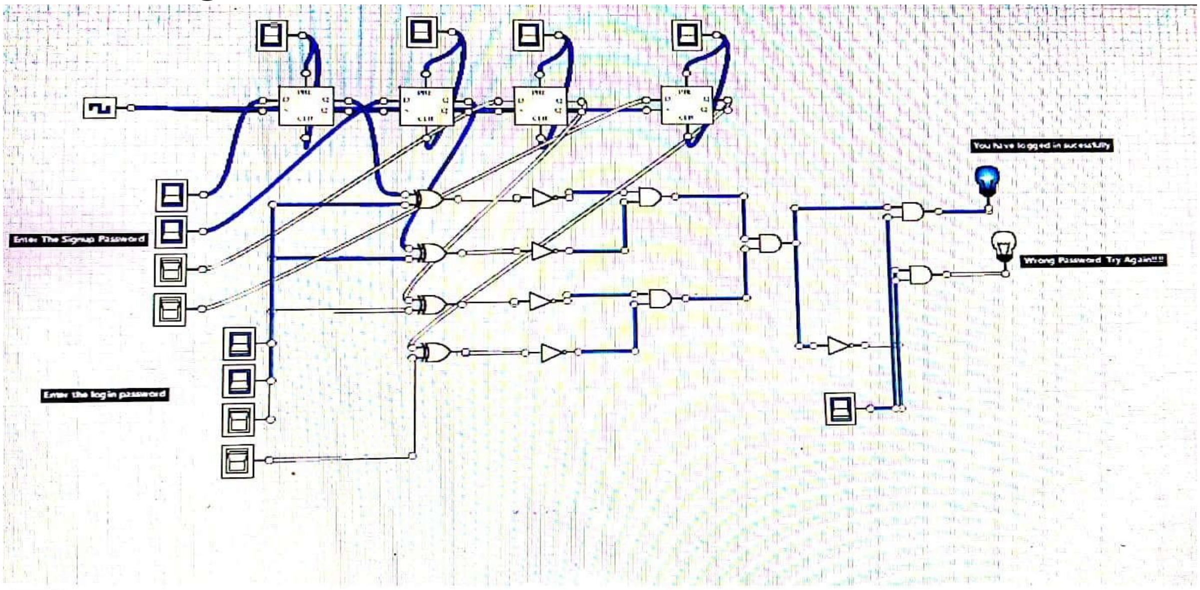
Clock	INPUT	OUTPUT	
	D	Q	Q’
LOW	x	0	1

HIGH	**0**	**0**	**1**
HIGH	**1**	**1**	**0**

Password Security System:

Working: For this project we have used XOR, NOT, SWITCHES, and LEDS. We have two set of input switches one set used for "KEY CODE SWITCHES" switches used to store password and other set of input switches "DATA ENTERY SWITCHES" used for storing current password entered by user.We have stored our password using d flip flop one set of inputs is given to flip flops which stores password different set of inputs is used to get input from user. Basically using XOR gate we compare output of respected inputs if they are same then output is 0 otherwise 1. If all inputs are same with respect to other corresponding input switch then our system by using NOT get before AND gives output that all the inputs given to AND Gate are same which also means output of all XOR are 0. NOT gate will invert all output of XOR so AND gate will provide output 1 which means our stored and current password has matched if output of AND is 0 the Incorrect bulb will give output and will glow

Circuit Diagram:



Truth Table:

A'	B	B'	C	C'	D	D'	A⊕A'	B⊕B'	C⊕C'	D⊕D'	(A⊕A')'	(B⊕B')'	(C⊕C')'	(D⊕D')'	(A⊕A')' ^ (B⊕B')'	(C⊕C')' ^ (D⊕D')'	((A⊕A')' ^ (B⊕B')') ^ ((C⊕C')' ^ (D⊕D')')	((A⊕A')' ^ (B⊕B')') ^ ((C⊕C')' ^ (D⊕D')')'
1	0	1	0	1	0	1	1	1	1	1	0	0	0	0	0	0	0	
1	0	1	0	1	1	0	1	1	1	1	0	0	0	0	0	0	0	
1	0	1	1	0	0	1	1	1	1	1	0	0	0	0	0	0	0	
1	0	1	1	0	1	0	1	1	1	1	0	0	0	0	0	0	0	
1	1	0	0	1	0	1	1	1	1	1	0	0	0	0	0	0	0	
1	1	0	0	1	1	0	1	1	1	1	0	0	0	0	0	0	0	
1	1	0	1	0	0	1	1	1	1	1	0	0	0	0	0	0	0	
1	1	0	1	0	1	0	1	1	1	1	0	0	0	0	0	0	0	
0	0	1	0	1	0	1	1	1	1	1	0	0	0	0	0	0	0	
0	0	1	0	1	1	0	1	1	1	1	0	0	0	0	0	0	0	
0	0	1	1	0	0	1	1	1	1	1	0	0	0	0	0	0	0	
0	0	1	1	0	1	0	1	1	1	1	0	0	0	0	0	0	0	
0	1	0	0	1	0	1	1	1	1	1	0	0	0	0	0	0	0	
0	1	0	0	1	1	0	1	1	1	1	0	0	0	0	0	0	0	
0	1	0	1	0	0	1	1	1	1	1	0	0	0	0	0	0	0	
0	1	0	1	0	1	0	1	1	1	1	0	0	0	0	0	0	0	

Logical Expression:

((A0 ⊕ B0)’ ^ (A1 ⊕ B1)’ ^ (A2 ⊕ B2)’ ^ (A3 ⊕ B3)’)(Key) ← Green(Correct)

((A0 ⊕ B0)’ ^ (A1 ⊕ B1)’ ^ (A2 ⊕ B2)’ ^ (A3 ⊕ B3)’)’(Key) ← Red (Incorrect)