29-8-2020

DAY2

INTRODUCTION TO QUANTUM COMPUTATION

TOPICS

- Vector Space
- Qubit
- Quantum Gates
- Creating Classical Logical Gates.

VECTOR SPACE

A vector space is a set that is closed under finite vector addition and scalar multiplication.

vector addition:

If $v, u \in V$ then $v+u \in V$ where V is Vector Space

scalar multiplication:

If $v \in V$ and $c \in F$ then $c*v \in V$

VECTOR SPACE

Let's see for a special vector space V defined as:

$$V = \begin{bmatrix} x \\ y \end{bmatrix} \quad x, y \epsilon \mathbb{C} \qquad F = \mathbb{C}$$

$$u, v \epsilon V \qquad u = \begin{bmatrix} u1 \\ u2 \end{bmatrix} \; v = \begin{bmatrix} v1 \\ v2 \end{bmatrix}$$

$$c \epsilon \mathbb{C}$$

VECTOR SPACE

$$u + v = \begin{bmatrix} u1 + v1 \\ u2 + v2 \end{bmatrix}$$

$$c.v = \begin{bmatrix} c.v1 \\ c.v2 \end{bmatrix}$$

$$u * v = u^{+}.v = \begin{bmatrix} \bar{u}1 & \bar{u}2 \end{bmatrix} . \begin{bmatrix} v1 \\ v2 \end{bmatrix}$$

$$= \bar{u}_1.v_1 + \bar{u}_2.v_2 \epsilon \mathbb{C}$$

VECTOR SPACE

Find the basis set for V

VECTOR SPACE

Find the basis set for V

$$Basis\ Set1 \Rightarrow \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$Basis\ Set2 \Rightarrow \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

VECTOR SPACE

Let's take a subset of V,

So $Q \subseteq V$

Now, Q is defined as:

$Q = \{q \mid q \in V , q^*q = 1\}$

What's its BASIS set?

Is it the same of something different?

VECTOR SPACE

Basis of Q

$$Set\ 1 \Rightarrow \begin{bmatrix}1\\0\end{bmatrix}, \begin{bmatrix}0\\1\end{bmatrix}$$

$$Set\ 2 \Rightarrow \frac{1}{\sqrt{2}}\begin{bmatrix}1\\1\end{bmatrix}, \frac{1}{\sqrt{2}}\begin{bmatrix}1\\-1\end{bmatrix}$$

Notation:

$$|0> = \begin{bmatrix}1\\0\end{bmatrix}, |1> = \begin{bmatrix}0\\1\end{bmatrix}$$

$$|+> = \frac{1}{\sqrt{2}}\begin{bmatrix}1\\1\end{bmatrix}, |-> = \frac{1}{\sqrt{2}}\begin{bmatrix}1\\-1\end{bmatrix}$$

QUBIT

- More than the classical bit, qubit can stay in 1 or 0 and even in the combination of the two, aka superposition state.
- What's a superposition state?
- Representation of qubit.
- Classical 0 is mapped to |0> and 1 to |1>
- And the whole set Q is the set of all superposition state of qubit.

QUBIT

Intresting videos on formation and working on qubits.

- https://www.youtube.com/watch?v=zNzzGgr2mhk&t=329s
- https://www.youtube.com/watch?v=ZuvK-od647c
- https://www.youtube.com/watch?v=QuR969uMICM
- https://www.youtube.com/watch?v=jDW9bWSepB0

QUBIT

Visualizing the state vector of qubit.

$$|\varphi> = \begin{bmatrix} x \\ y \end{bmatrix} = x.|0> + y.|1>$$

$$|\varphi> \in Q \quad \Rightarrow |\varphi>^{+}.|\varphi> = 1$$

$$Notation \ : |\varphi>^{+} = <\varphi|$$

$$So, <\varphi|\varphi> = 1$$

QUBIT

Points to remember :

If **Ψ(x)** is the wave function of the quantum state and the probabity amplitude of the quantum state will be mod(**Ψ(x)**) ^2.

Here our |**Ψ**> is given by:

$|\boldsymbol{\Psi}> = x*|0>+y*|1>$

So probability amplitude will be

For |0> **Ψ(**|0>**)** is x making probability of |0> to be x^2

For |0> **Ψ(**|1>**)** is y making probability of |1> to be y^2

QUBIT

Points to remember :

If **Ψ(x)** is the wave function of the quantum state and the probabity amplitude of the quantum state will be mod(**Ψ(x)**) ^2.

Here our |**Ψ**> is given by:

|**Ψ**> = x*|0>+y*|1>

So probability amplitude will be

For **Ψ(**|0>**)** is x making probability of |0> to be x^2

For **Ψ(**|1>**)** is y making probability of |1> to be y^2

QUBIT

So qubit can take any state from the set Q but a classical bit has only 2 states to be in. You can clearly see the advantage of working with the qubit.

In simple words a single qubit compared to classical bit has a huge computational advantage because of that.

Now with a bunch of qubits which can interact together these superposition states can interact in a constructive or destructive manner and we use these for our advantage.

HOW TO MANIPULATE QUBIT

We do this using Quantum Gates.

There are bunch of gates for different types of interaction and change of state.

HOW TO MANIPULATE QUBIT

We will start with some simpler ones and then get into the complex gates.

- X Gate
- CX Gate
- CCX Gate

Then we will try to replicate classical gates on qubits.

X GATE

Let's start with the most basic gate : X Gate

Apply the X gate and the qubit state gets flipped.

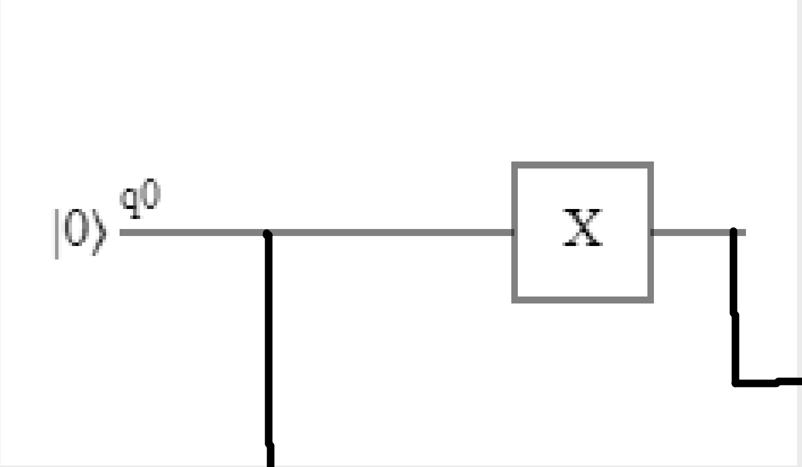
|1> to |0> and |0> to |1>

If we start with a state $|\boldsymbol{\Psi}> = a^*|0> +b^*|1>$

Applying X gate will change the state to

$|\boldsymbol{\Psi}>_{new} = b^*|0> +a^*|1>$

X GATE

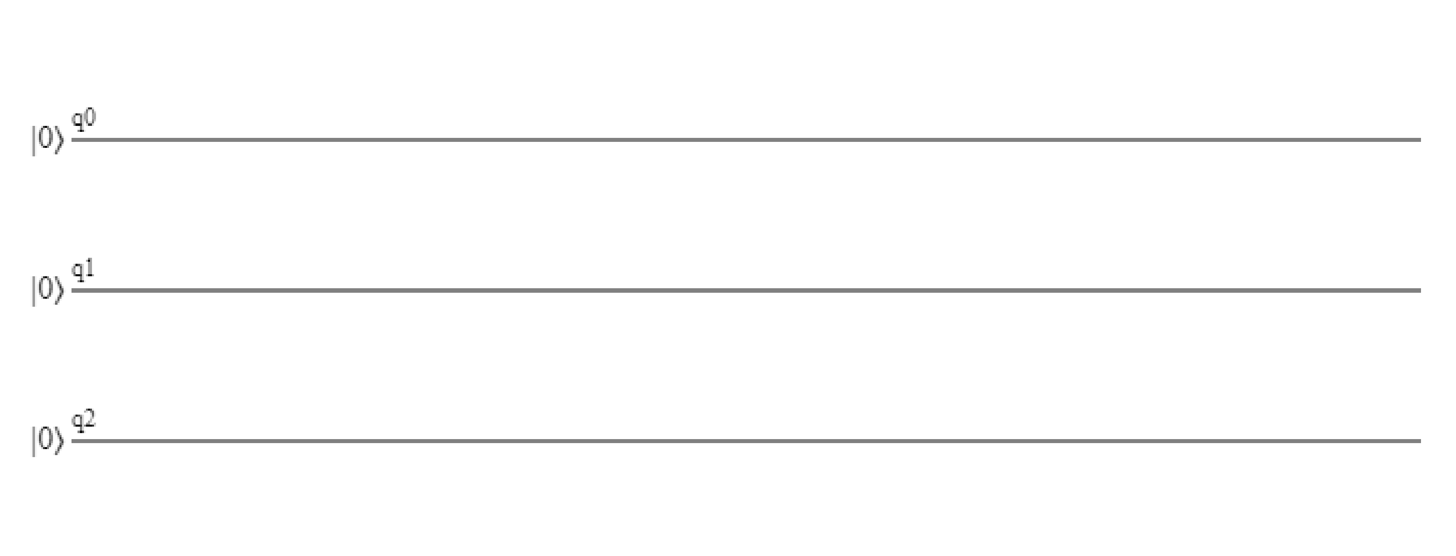


$$|\varphi\rangle = |0\rangle$$

$$|\varphi\rangle_{new} = |1\rangle$$

MULTIPLE QUBITS

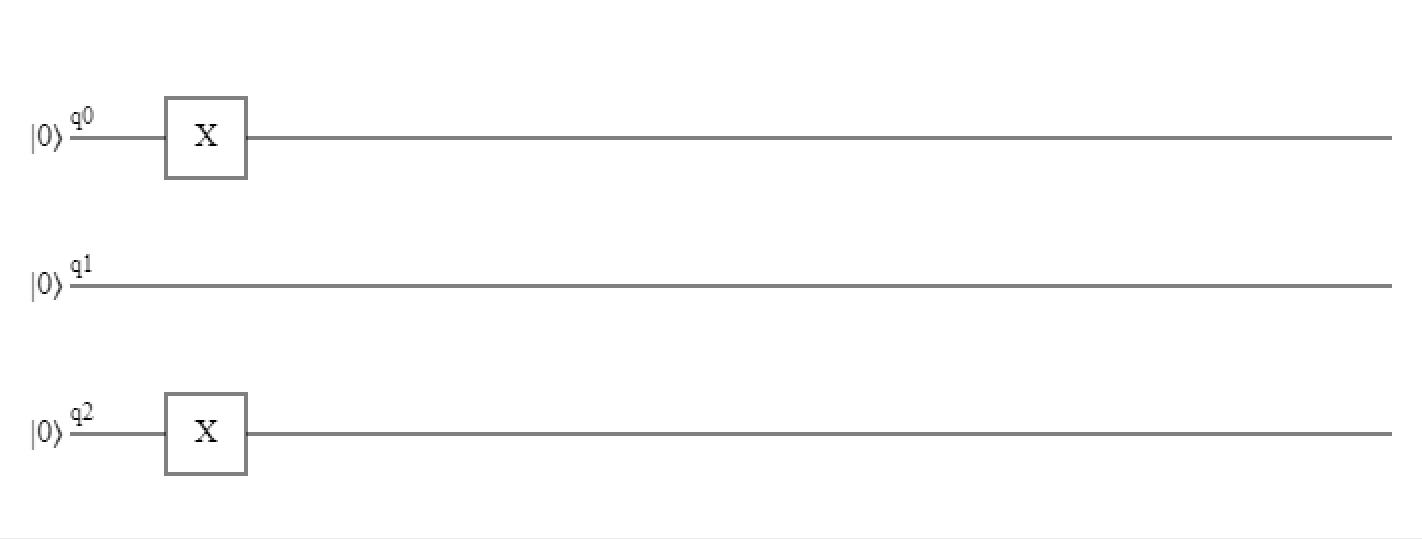
How to represent multiple qubit.



We write the qubits together as |q2q1q0>

MULTIPLE QUBITS

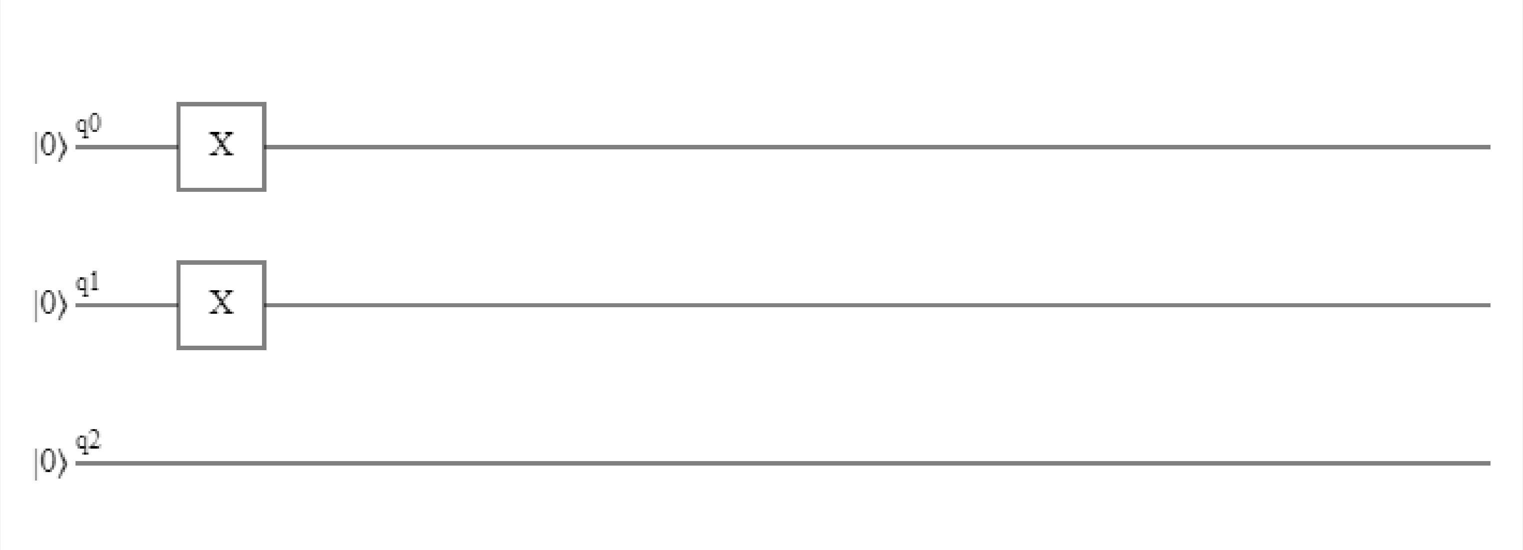
How to represent multiple qubit.



|q2q1q0> gives us |101>

MULTIPLE QUBITS

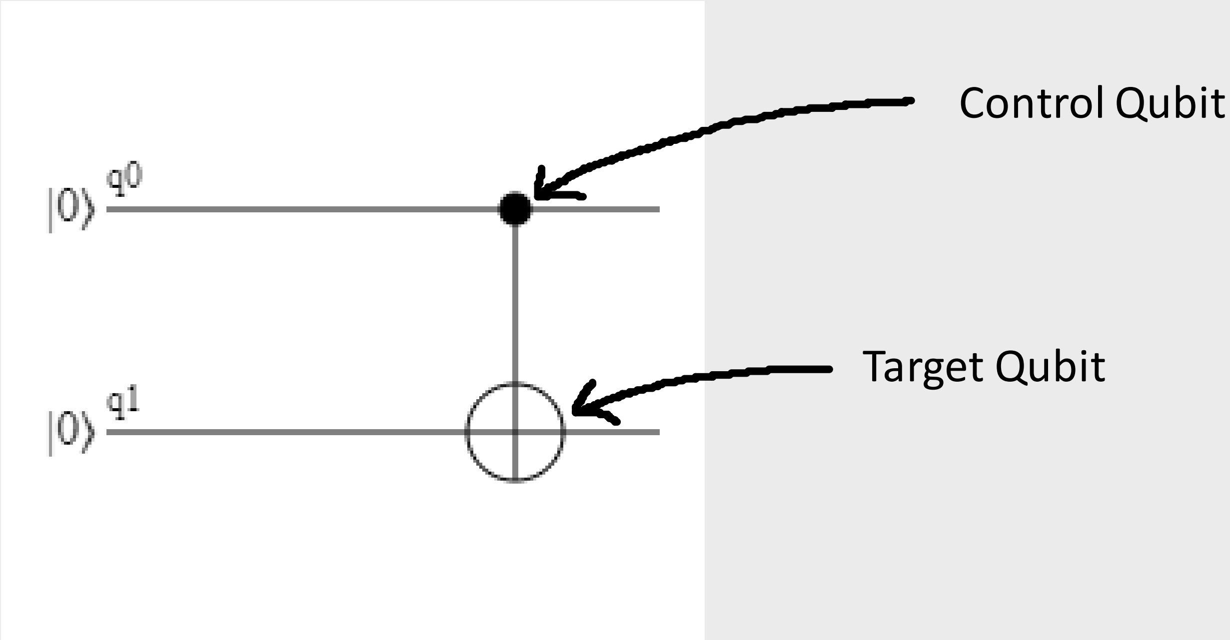
How to represent multiple qubit.



|q2q1q0> gives us |011>

CX GATE

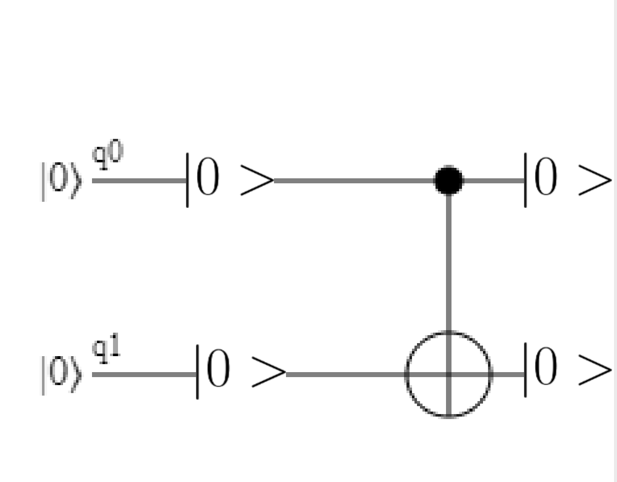
CX Gate is Controlled X gate. It has a control qubit and a target qubit.



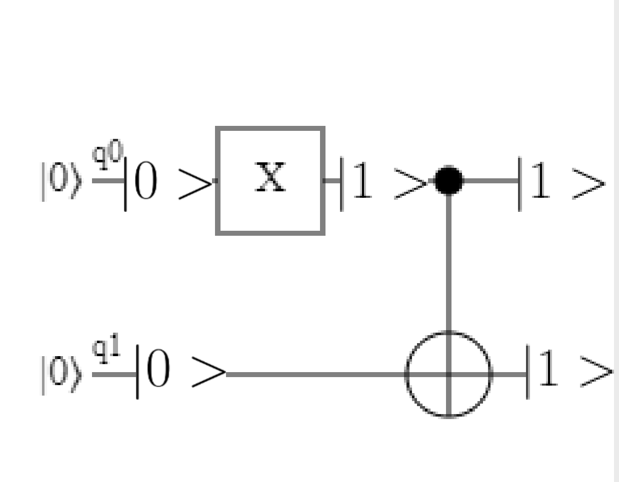
CX GATE

Whenever control qubit is in |1> state it flips the target qubit.

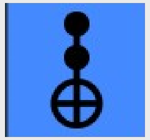
If the control qubit is |0> state, then it just leaves the target qubit as it was.



Target Qubit is not |1> so the control doesn't get activated



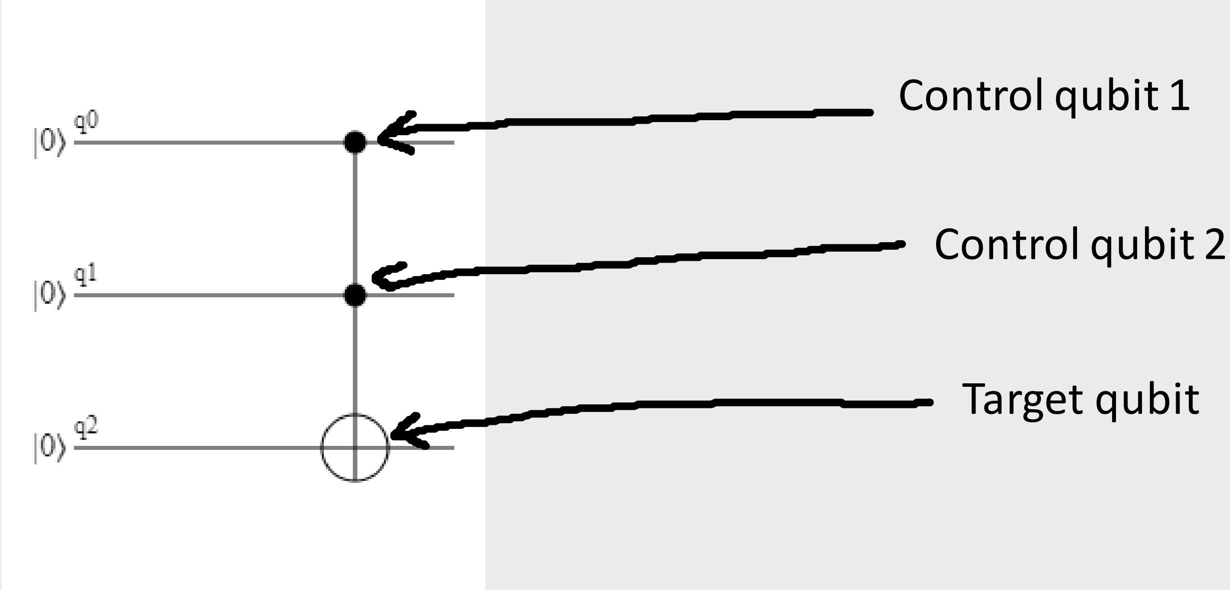
Target Qubit is in |1> so the control gets activated flipping the target qubit to |1>

CCX GATE 

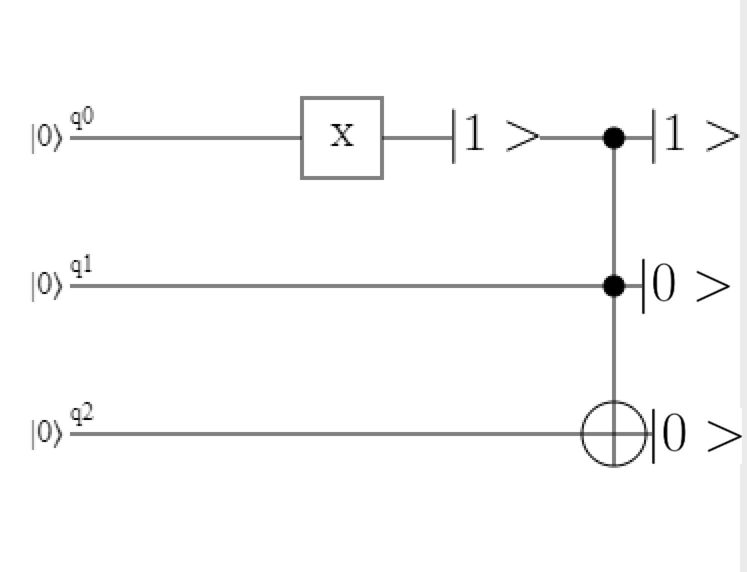
CCX is Controlled Controlled X Gate aka Double Control X Gate.

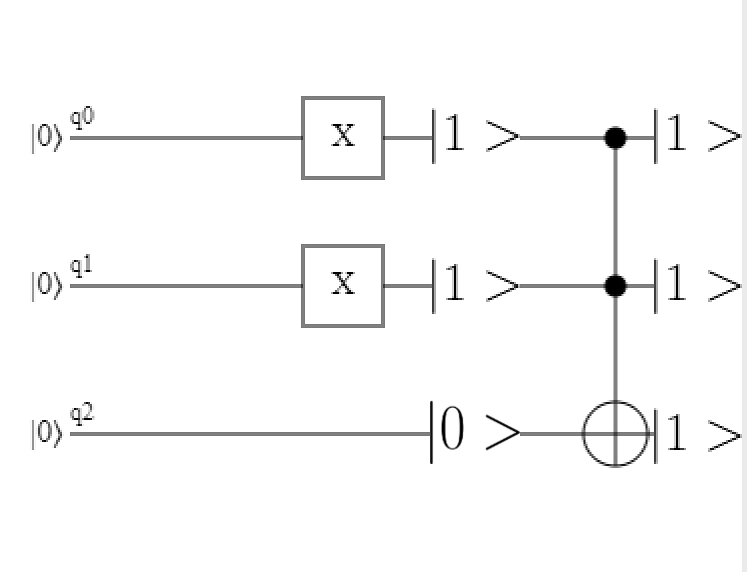
It has 2 control qubits and 1 target qubit.

When both the control qubits are in |1> state only then it flipes the target qubit.



CCX GATE





In this only one control qubit is activated and the other one is not. So the target qubit is not flipped.

We started with |001> ended up as |001>

In this both control qubit has been activated resulting in the flip of the target qubit.

We started with |011> ended up as |111>

LOGIC GATES

- NOT
- AND
- OR
- NAND
- NOR
- XOR
- XNOR

INPUT		OUTPUT					
A	B	AND	NAND	OR	NOR	XOR	XNOR
0	0	0	1	0	1	0	1
0	1	0	1	1	0	1	0
1	0	0	1	1	0	1	0
1	1	1	0	1	0	0	1

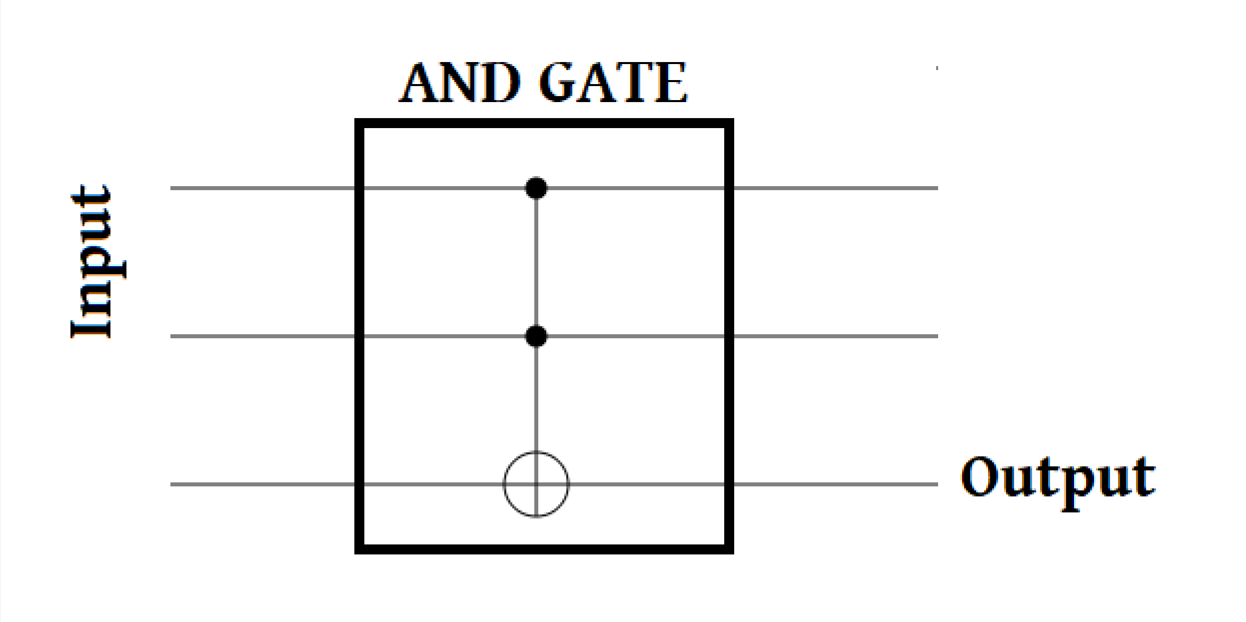
AND GATE

If we take q0 and q1 as the input and q3 as out output, we have :

<table>
<tr><th colspan="2">Input_Qubits</th><th>Input_State</th><th>Output_Qubit</th><th>Output_State</th></tr>
<tr><td>Qubit(0)</td><td>Qubit(1)</td><td>|q2q1q0></td><td>Qubit(2)</td><td>state</td></tr>
<tr><td>|0></td><td>|0></td><td>|000></td><td>|0></td><td>|000></td></tr>
<tr><td>|0></td><td>|1></td><td>|010></td><td>|0></td><td>|010></td></tr>
<tr><td>|1></td><td>|0></td><td>|001></td><td>|0></td><td>|001></td></tr>
<tr><td>|1></td><td>|1></td><td>|011></td><td>|1></td><td>|111></td></tr>
</table>

You can simply see that it's just a CCX gate.

AND GATE



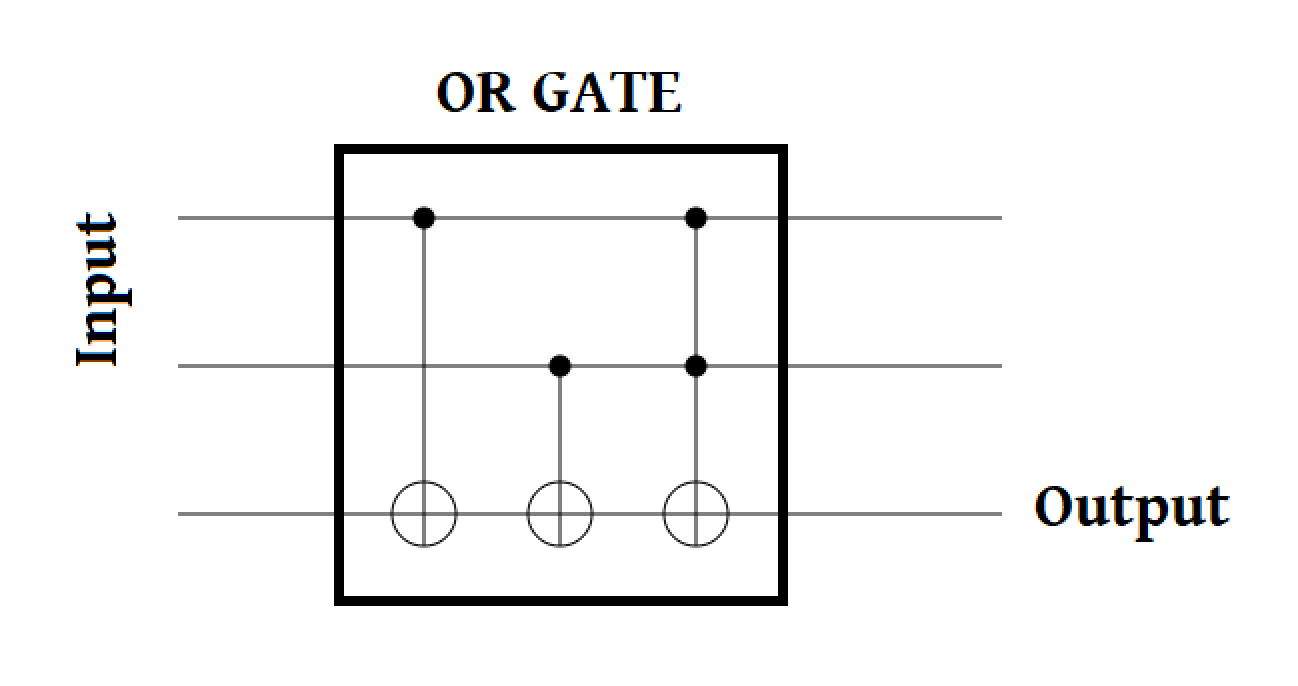
OR GATE

Taking q0 and q1 as inputs and q2 as output, we have:

Input_Qubits		Input_State	Output_Qubit	Output_State
Qubit(0)	Qubit(1)	|q2q1q0>	Qubit(2)	state
|0>	|0>	|000>	|0>	|000>
|0>	|1>	|010>	|1>	|110>
|1>	|0>	|001>	|1>	|101>
|1>	|1>	|011>	|1>	|111>

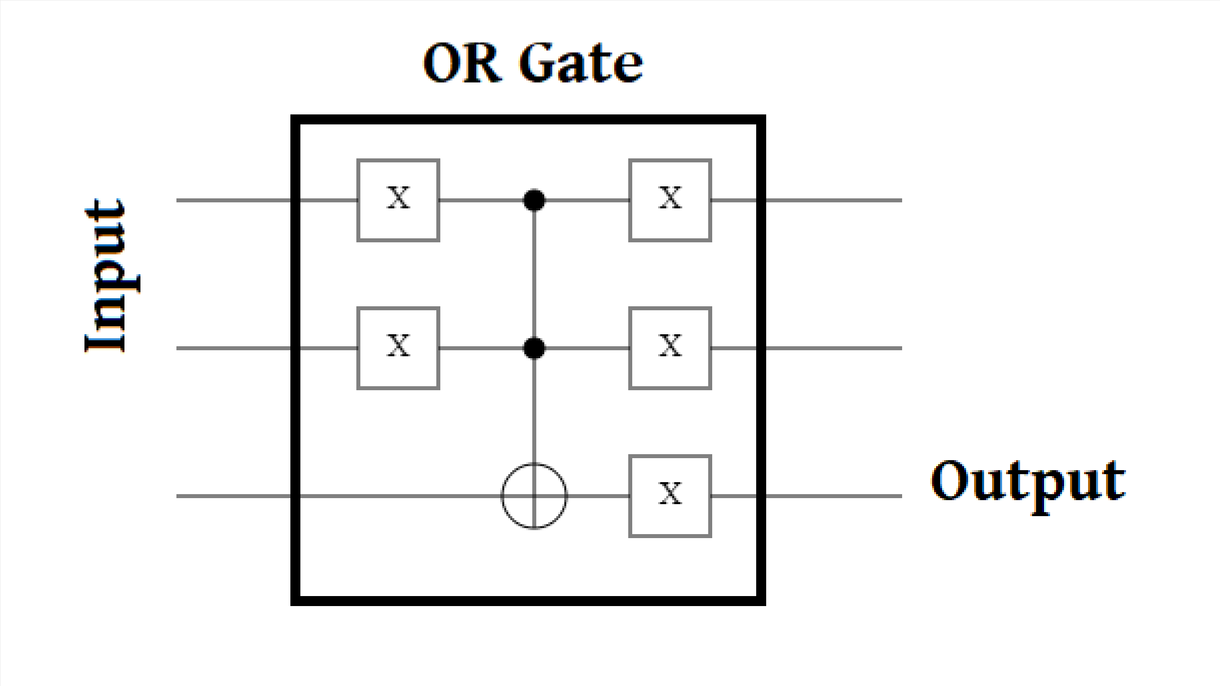
Looks a bit complicated, so let's work on it.

OR GATE



There are other circuits that can do the same thing.
Let's see if you guys find any.

OR GATE



HOMEWORK

Build the rest of the LOGIC GATES.

- NAND
- NOR
- XOR
- XNOR

THANK YOU !

LET'S MEET TOMORROW WITH THE CIRCUITS OF THE OTHER 4 GATES AND I WILL GIVE YOU NEW PROBLEMS TO PLAY WITH.