NPTEL

NATIONAL PROGRAMME ON TECHNOLOGY ENHANCED LEARNING

IIT BOMBAY

CDEEP
IIT BOMBAY

Quantum Information and Computing

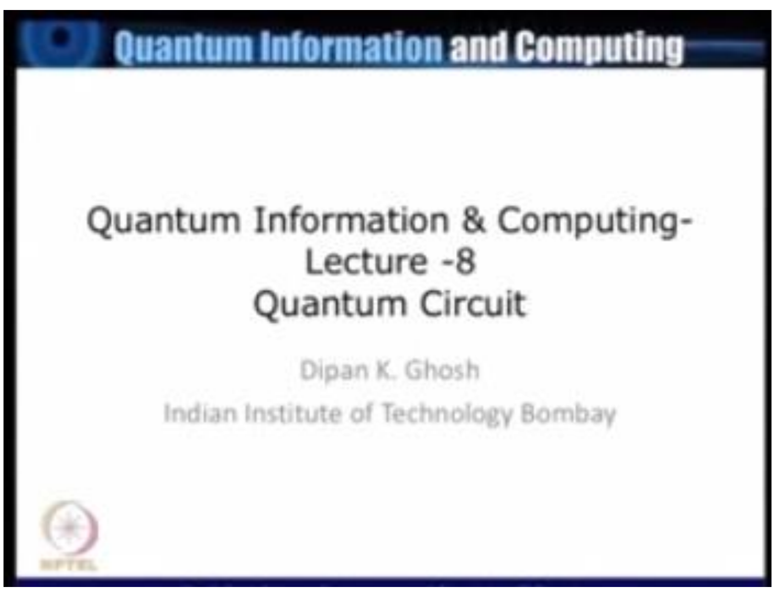
Prof. D.K.Ghosh
Department of Physics IIT Bombay

Modul No.02

Lecture No.8

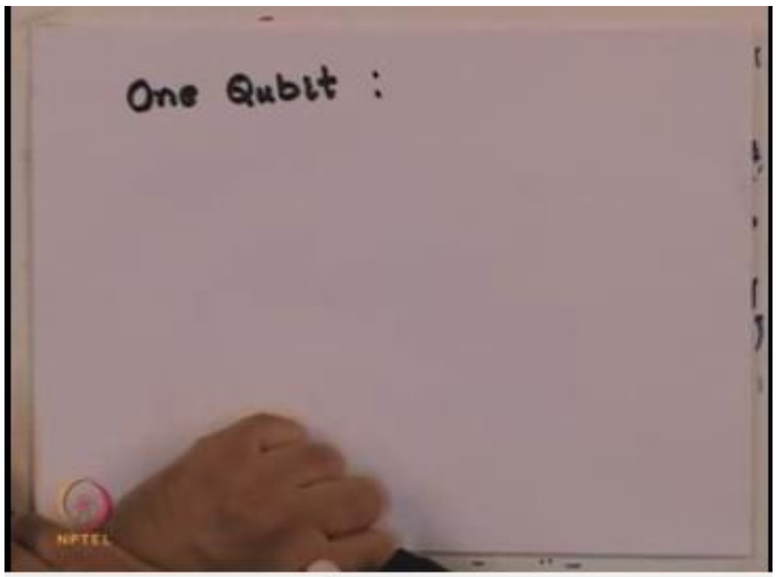
Quantum Circuits

(Refer Slide Time: 00:16)



In the last lecture we had introduced certain elements which we called as the gates, which work on single qubit, 2 qubits and 3 qubits and essentially what we said is a few of this elements are good enough to construct what we will call as a quantum circuit as we go along. So let me just review for you these elements for example.

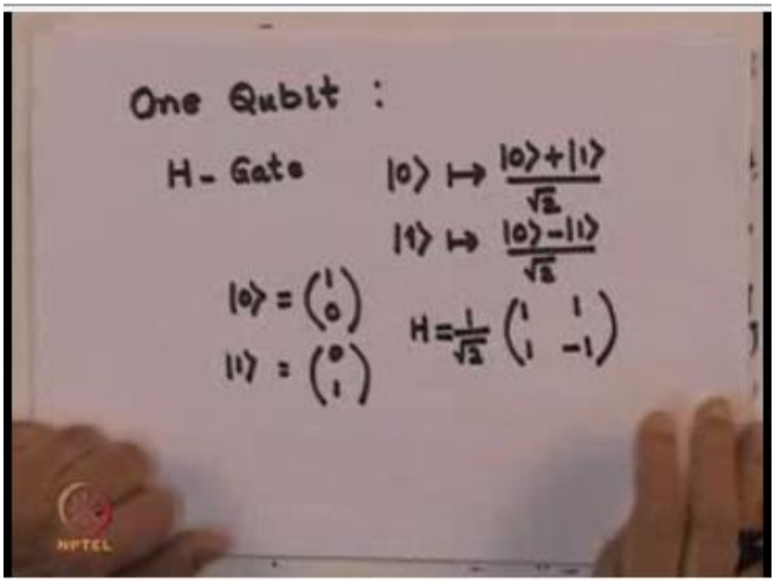
(Refer Slide Time: 00:45)



What happens to a one qubit gates, so basically as we have seen since a single qubit can be represented as a point on a bloch sphere, so basically all operations which take one point on the blochs get to another which consists of the rotations and reflections which will take a point on the sphere to another point on the sphere are permitted. And in general we have seen in our representation of the gates by means of matrices.

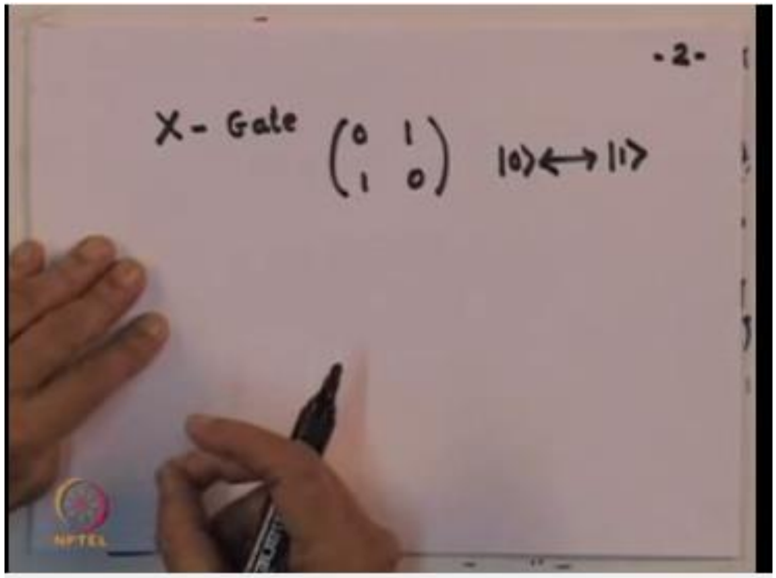
Since a single qubit state is represented by a column vector with two components my operators or the matrices corresponding to single qubit gates will be 2/2 matrices, I will just be mentioning again because we will be coming across them regularly some more common gates, for example.

(Refer Slide Time: 01:46)



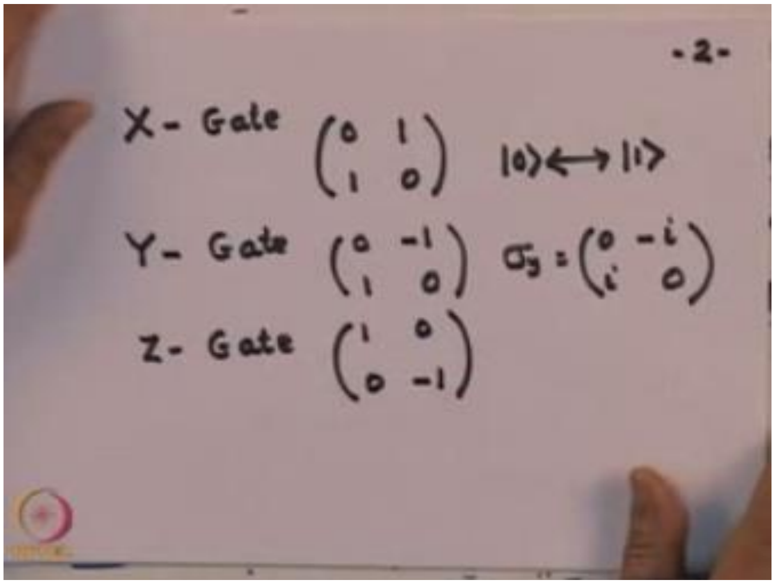
The Hadamard gate or H gate is a gate which takes the state |0> → |0> + |1> by root 2 and to state |1> → |0> - |1> by root 2 and with our matrix representation 0 being given by 10 and 1 being given by 01 the representation for the Hadamard matrix is 1 over square root of 2 with (1, 1, 1, -1). So this is an important gate in projects on a single qubit state, the there are others which we have mentioned last time but let me talk about X gate.

(Refer Slide Time: 02:43)



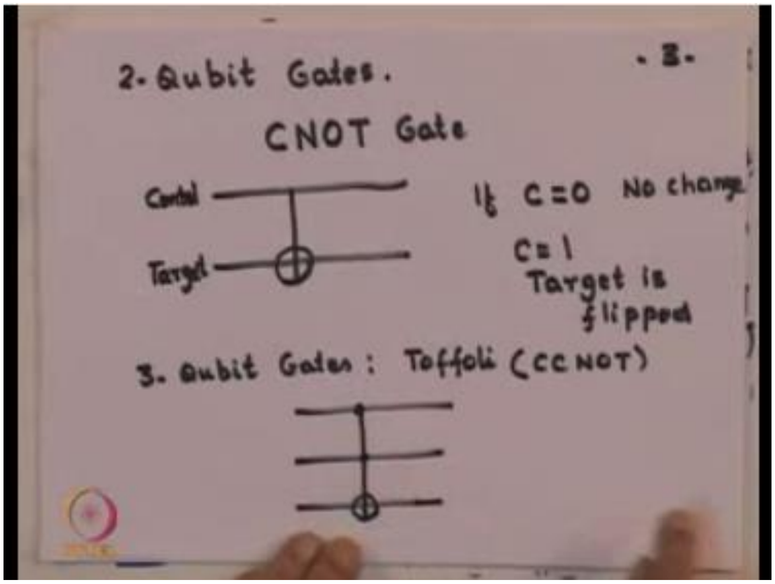
X-gate is simply given by 0110 is standard Pauli matrix and its job is to take the |0> → |1> and vice versa. So this is basically equivalent to a classical not gate, like X-gate we have Z-gate which changes the sign of the component 1 and also we have a Y-gate which is given by a matrix very similar to the Pauli Y-matrix, I will write it down.

(Refer Slide Time: 03:26)



Because there seems to be different notations slightly. So Y-gate is generally written as (0, -1, 1, 0) though my σ_y is (0, -i, i, 0), but it is a matter of simply putting in the another constant i in front of that so it should not really cause any confusion and off course we have Z –gate which is given by the Pauli z matrix which is (1, 0, 0, -1) so these are the more important one qubit gate, so talking about 2 qubit gates.

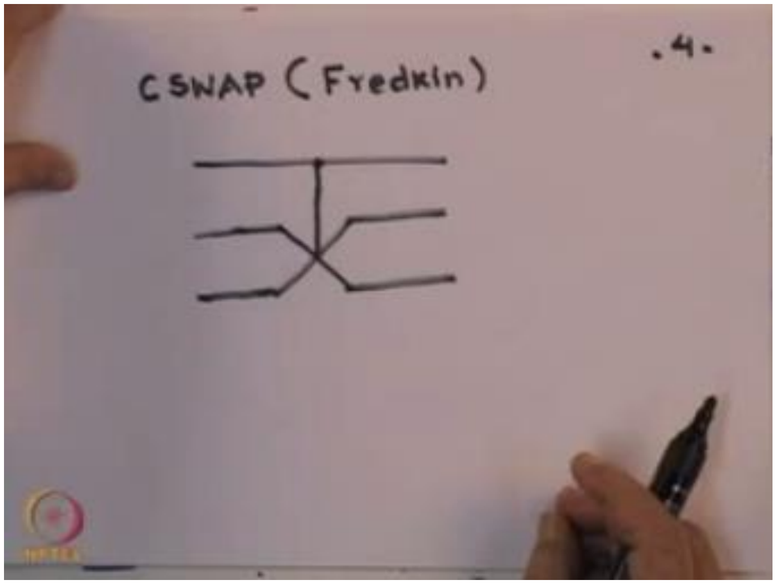
(Refer Slide Time: 04:11)



The most significant is what we called as the control NOT gate or CNOT gate if you recall what CNOT gate is that a CNOT gate basically has a control and a target. So the, its representation and circuit is like this. So if control = 0, target remains target, no change if control = 1 the target bit is flipped that is 0 becomes 1 and 1 becomes 0. So if C = 1 target if flipped, we also talked about a few 3 qubit gates.

The first one was a Toffoli gate which is also known as a CCNOT gate control, control not gate represented like this, so basically the target bit which is this, this has two controls the target bit will be flipped if work this control and that control happen to be equal to 1 and nothing happens to the target bit if one of the one or both of them happen to be equal to 0. And finally we had talked about a controlled swap.

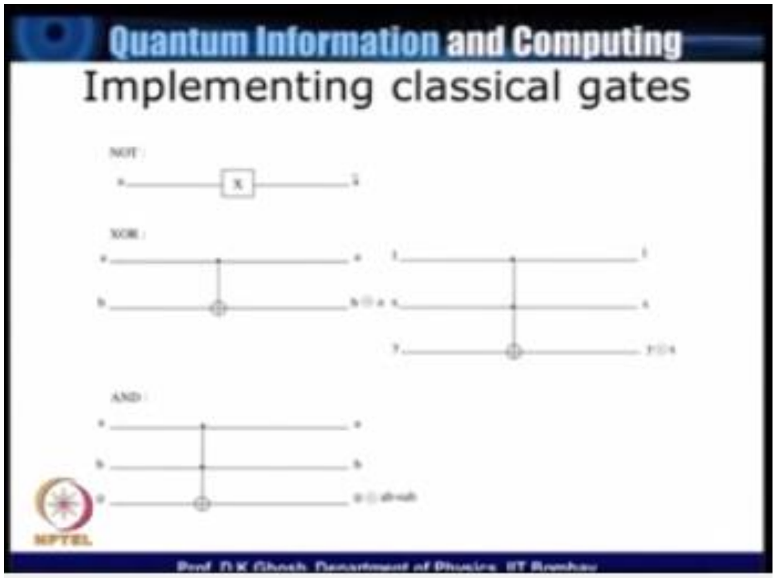
(Refer Slide Time: 05:59)



CSWAP which is also called a Fredkin gate, so basically a Fredkin gate again has a control, but there are targets on which it will act and so this is essentially flip the targets. So this is your control it will flip the target if the control bit happens to be equal to one. So these where the elementary gates which we talked about and what we are going to do today is to talk about quantum circuits.

Which basically consist of the gates who are called the inputs and then finally of course give rise to an output. So let us look at some of the implementation of the classical circuits based on that.

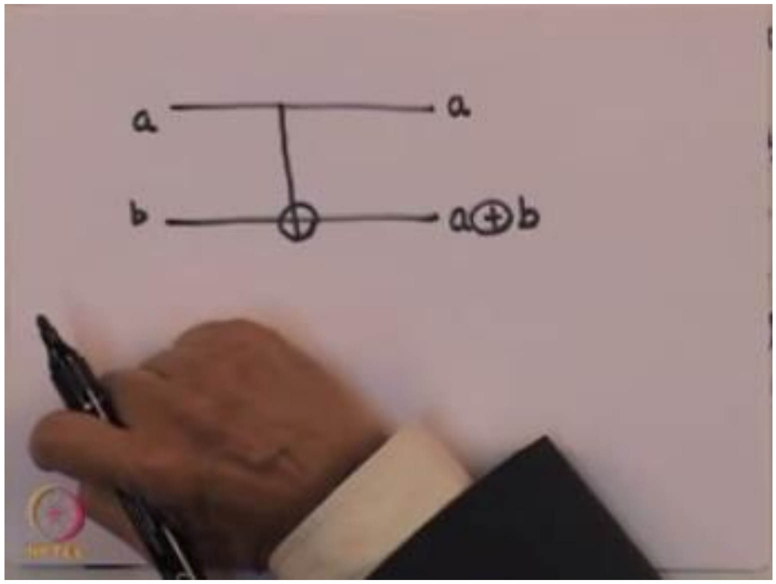
(Refer Slide Time: 06:55)



So if we look at the slide now, you notice that the classical NOT gate which is actually the only reversible classical gate is essentially remains the same NOT gate because all that you need is the unitary matrix corresponding to the Pauli X gate and so therefore if the input if bit is A then X get acts on it and it becomes a bar. Look at XOR, XOR as you know is addition modulo 2 that is only if, now we are talking about but one of the bits to be 0 the other bit to be equal to 1 then only we get 1 otherwise we get 0.

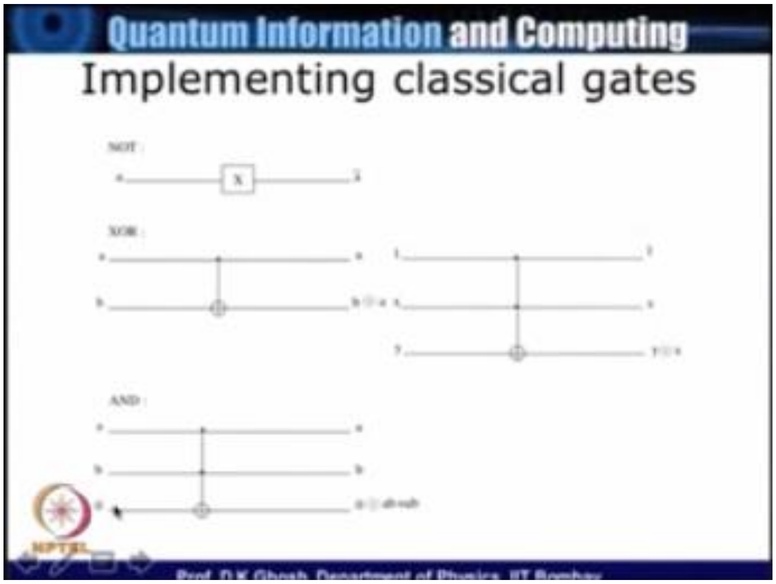
So this is same as addition modulo 2 that which is nothing but CNOT gate so as you can see it that supposing I have this as the bit a and this as the bit b.

(Refer Slide Time: 07:52)



So this is a and this is obviously a XOR b, so this is addition modulo 2 or the XOR referring back to the slides again.

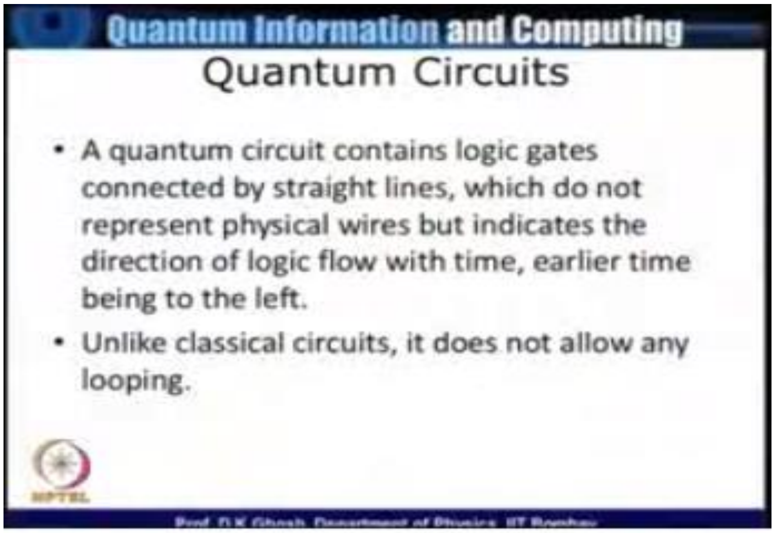
(Refer Slide Time: 08:11)



The you will look at the right hand side of this picture, if you look at the right hand side of the picture I have given the implementation of XOR using a CCNOT gate. So look at this so for example, what happens here is that one of the controls I have set as the bit 1 another control is x, so therefore the CCNOT gate the target will be flipped only if x=1. So therefore, the product must be good one and so y then becomes y+ x because if x=0 then of course this is simply y x or 0 and so therefore either of these can be considered as an equivalent of the XOR gate.

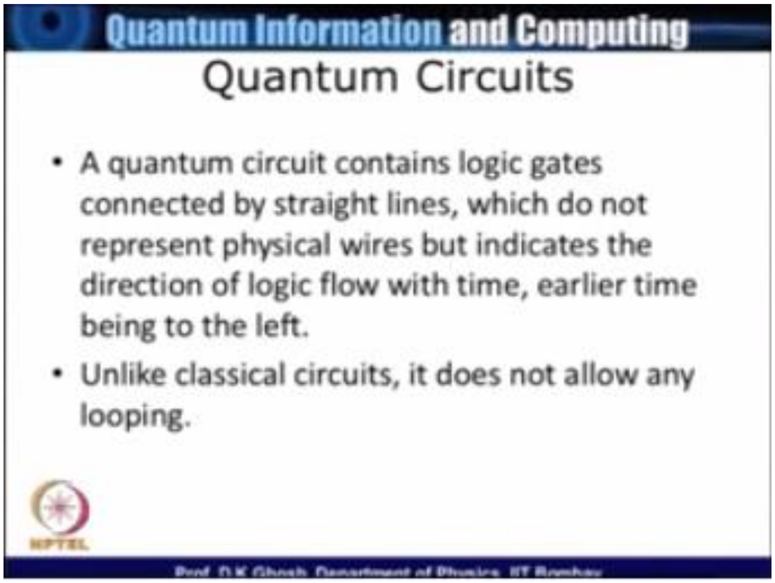
Now a CCNOT gate can be used to implement a classical AND gate so you look at what we have done here I have got two controls at each side so therefore these two controls can be in principle a and b. And what we are done is the target bit we have put to be equal to 0, now remember 0 x or anything okay, is the same thing itself. So therefore, if I am having both the controls let be a and b equal to 1, then my target bit it bit becomes 1 and that is what a classical and always does because a x b must be equal to 1.

(Refer Slide Time: 09:54)



So let us go to the general idea of what is the quantum circuit. So basically a quantum circuit contains logic gates and there you have noted that I have been connecting these various logic gates by means of straight lines. Now in normal classical circuits these straight lines represent wires and physical wires. Now in this particular case, in case of quantum circuits or they need not stand for any physical wire, but they essentially indicate the direction in which the logic flows, the, essentially what we are saying is you should have the input at the extreme left most part of the circuit then as time progresses the, we continue with the application of logic gates.

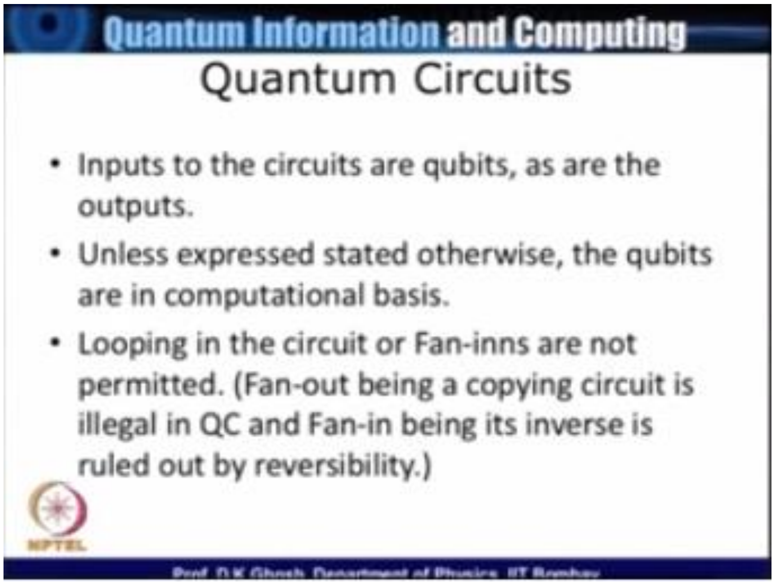
(Refer Slide Time: 10:48)



In the same order in bits they come and that is always logic flows with earlier time being to the left. Now there is some restrictions remember in a classical algorithm we allow what are called looping or fan-ins as they are called, so basically what we do is that you can loop at a particular logic. Now this is not permitted here because as I told you the diagram essentially represents a flow of logic with time. Now if you allowed the looping then of course you could been you have to travel back in time.

So therefore unlike classical circuits looping is not permitted in a quantum circuit. So inputs are qubits the outputs of course are also qubits.

(Refer Slide Time: 11:47)

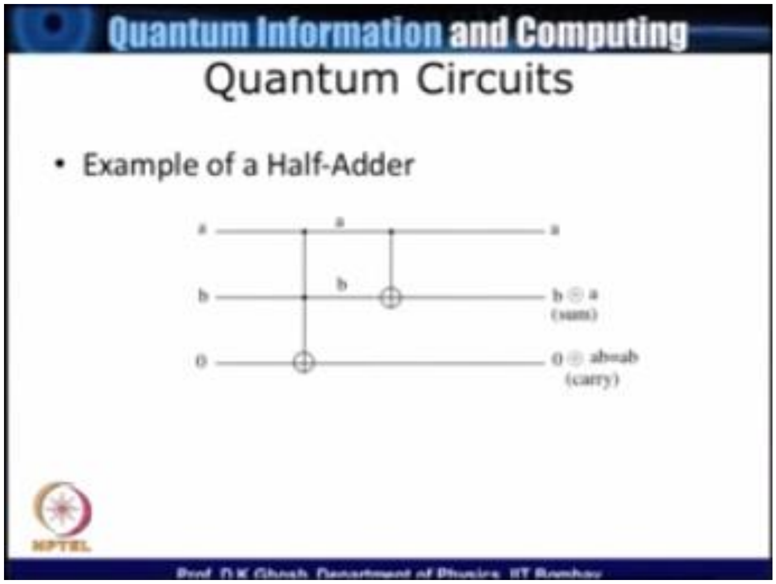


Now unless otherwise stated very expressively stated the qubits are usually expressed in computational basis this is not a requirement for a quantum circuit, but this is what that traditional practices. So therefore, we have said now sometimes what happens is you want to make your measurement in for example at Bell basis instead of a computational basis. But then the somewhere we must state that, that is what is happening. Then going back to the slide, what we said is looping or fan-ins are not permitted.

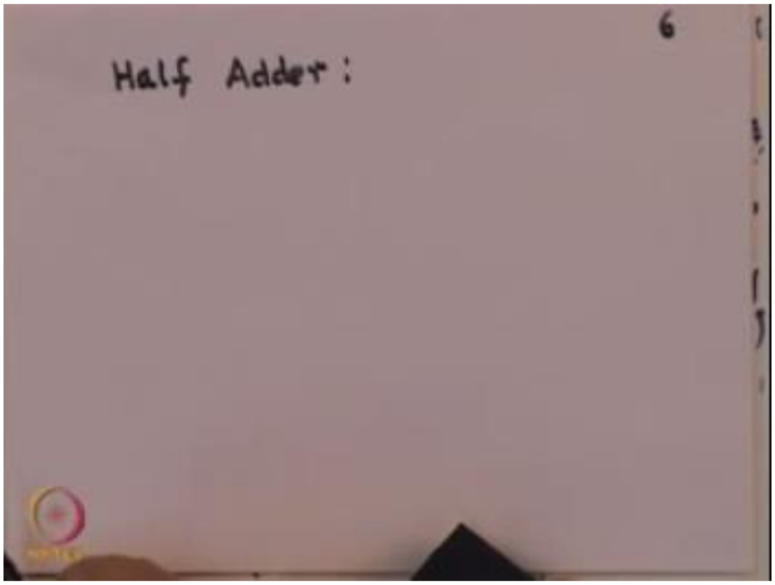
In other words, you cannot have several inputs giving rise to the same output. Now the point is this that the inverse process of that they is the fan-out circuit. Now fan-out circuit is basically a coping circuit, so therefore as we will see later in this lecture or the next depending up on time the, a quantum circuit which copies a state to another state in general state to another state is not permitted by a theorem known as the quantum no cloning theorem. And as a result fan-out is not permitted.

Like fan-out is not permitted then of course it is universe or the reverse fan-in which is also not permitted, because as we know quantum computation happens in a reversible manner. So let me give you an example of a quantum specific quantum circuits namely a quantum half-adder.

(Refer Slide Time: 13:40)

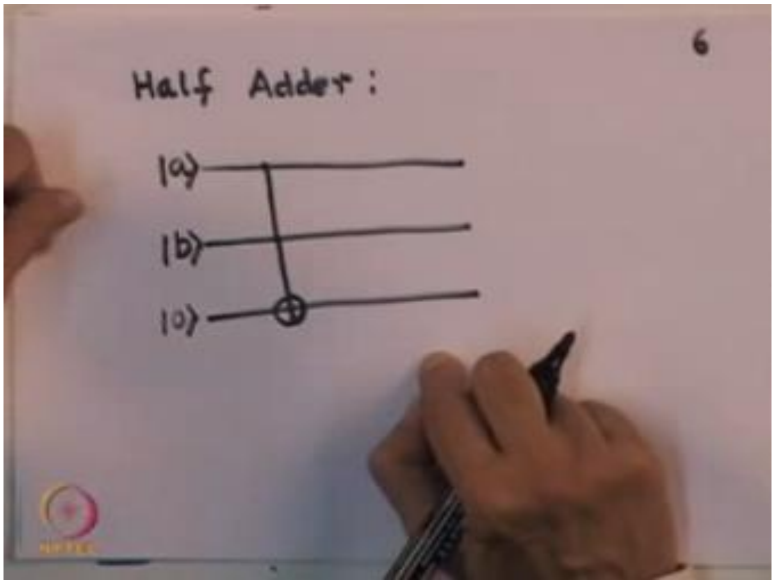


(Refer Slide Time: 13:44)



So as you know that basically I have this problem I want to add two single qubits or single even C bits classical bits. But then when the two classical bits are equal to one then I have a carry. So half adder essentially gets this sum which means 1 + 0 must be equal to 1, 0 = 0 must be equal to 0, and 0 + 1 is of course equal to 1 again but 1 + 1 is 1 0 which means my sum must be 0, and my carry must be 1. So a half adder circuit has essentially this situation.

(Refer Slide Time: 15:21)



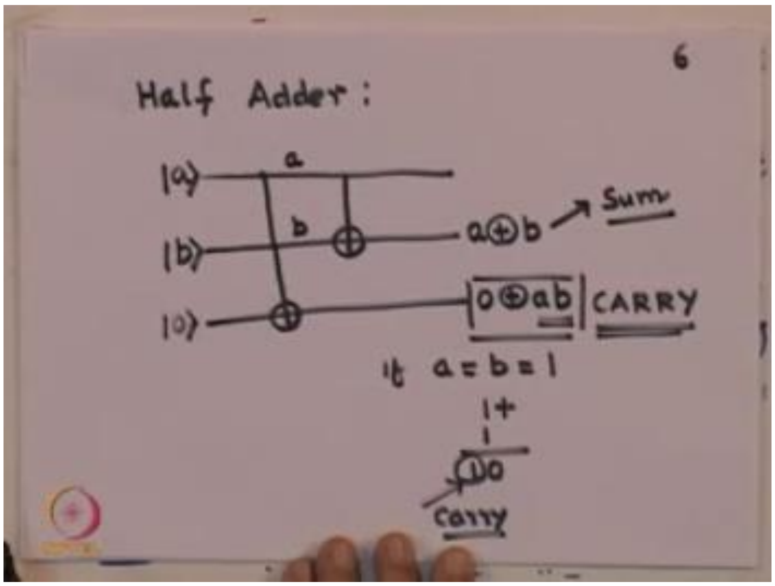
So let us look at how does a half adder circuit work. so this is a and this is b and what I do is I set this third one to 0 remember each one of them a single cubit so a and b can take value, you know 0 or 1 each. Now I start with the circuit which is a CCNOT gate so notice remember what is the CCNOT gate.

So a CCNOT gate what the inter change or would flip this bit 0 to 1 only both a and b are equal to 1, but remember the CCNOT gate does not do anything to the control themselves. So therefore this will remain a and this will remain b, and this is at this stage at the end of the CCNOT gate, what I get is a 0 + a b the 0 + a b, what is this now notice that a b has a product 1 only is both a and b happen to be equal to 1.

So therefore this will be XOR of 0 with 1 giving me 1, so if a and b each equal to 1 the last bit here would essentially have the carry, because 1 +1 in binary addition is 1, 0 and this is the carry there is no carry if either a is 0 or b is 0 or both 0 of course because in that case I get equal to because there is XOR operation I get equal to 0. But notice what is happening here this is still a and this is still B.

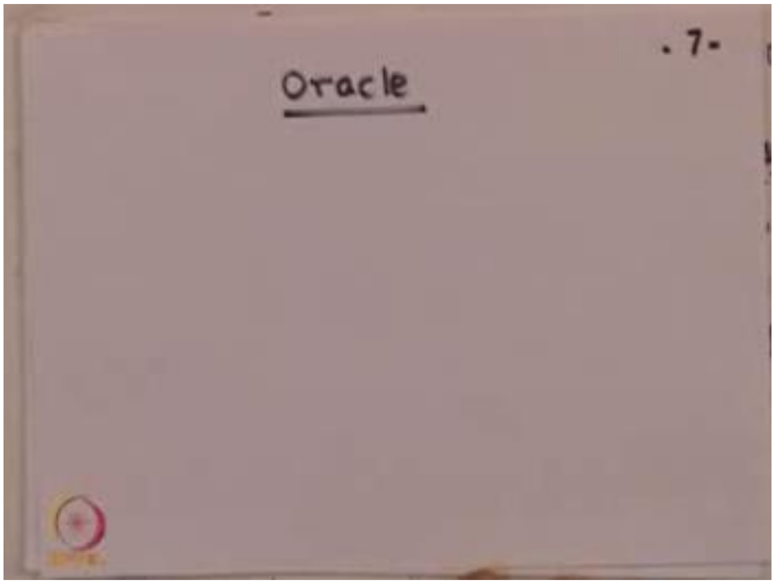
So supposing with a as the control and b as the on target I apply I single CNOT gate. Now this is nothing but a XOR b. So therefore when a = 0, b = 0, I of course get 0, a0, b1 or b0, a1 I get 1 which is what I expect from an addition and if a both b are equal to 1, then 1 XOR 1 is equal to 0, so this is by sum.

(Refer Slide Time: 17:40)



So this is the way we implement half adder you can do some practice on this type of a implementation of classical circuits, for example try what is the full adder and thinks like that. Now there is another important component of a quantum circuit before we write down a full fledged quantum circuit we need to understand this a little bit and this is what is known as a oracle not to be confuse with the corporations having the same name.

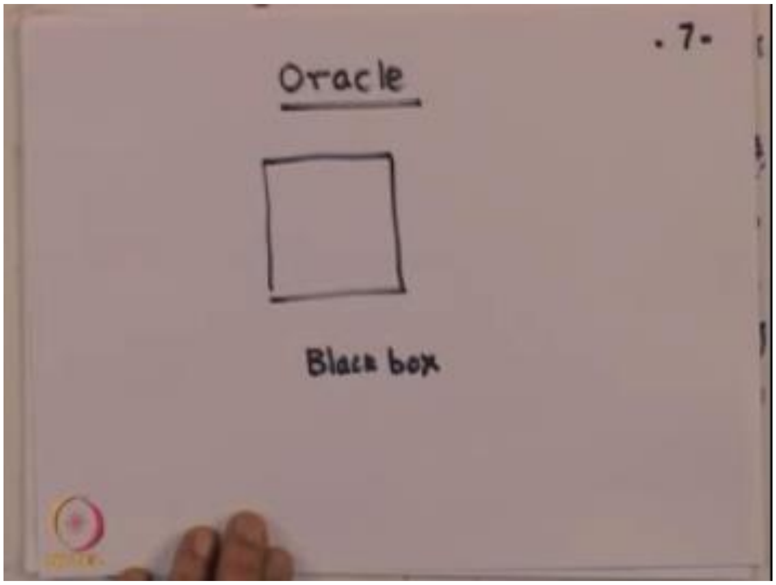
(Refer Slide Time: 18:15)



Oracle as you know is what whole side send people like that used to essentially, you know predict or in as is occasionally known as the Akashvani or whatever we learned from having. Now what is an oracle, the oracle in quantum computer have a very specific purpose, very offend we want to compute some functions and what we do in classical computer in this, we give a sub routine column. Now remember this sub routine calls we simply say column this, what is assumptive is that there is an algorithm corresponding to that available in the part of the program.

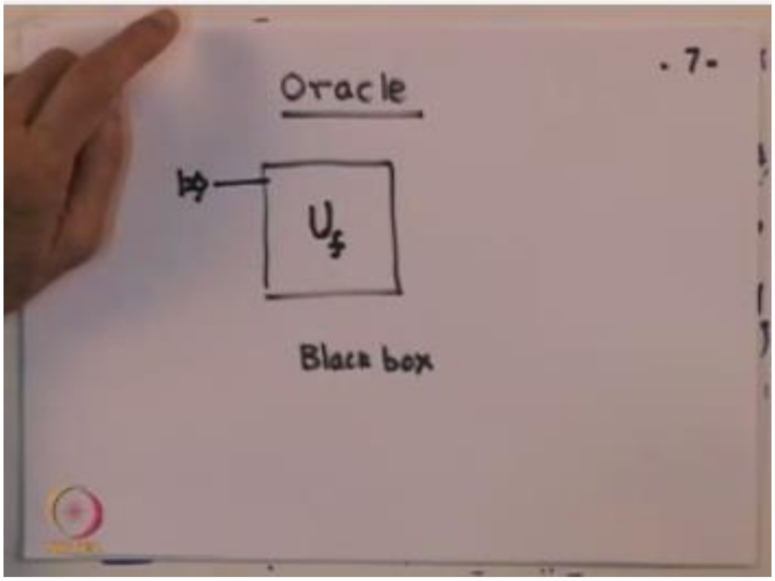
Now an oracle essentially does the same thing, so basically an oracle is something which takes in certain amount of input computes the function, we at this moment do not care how does it, but we assume that there is a quantum circuit which is enabled and which can compute this function and gives outputs the corresponding results. So therefore, it is like a black box so oracle is represented like this it is the black box.

(Refer Slide Time: 20:00)



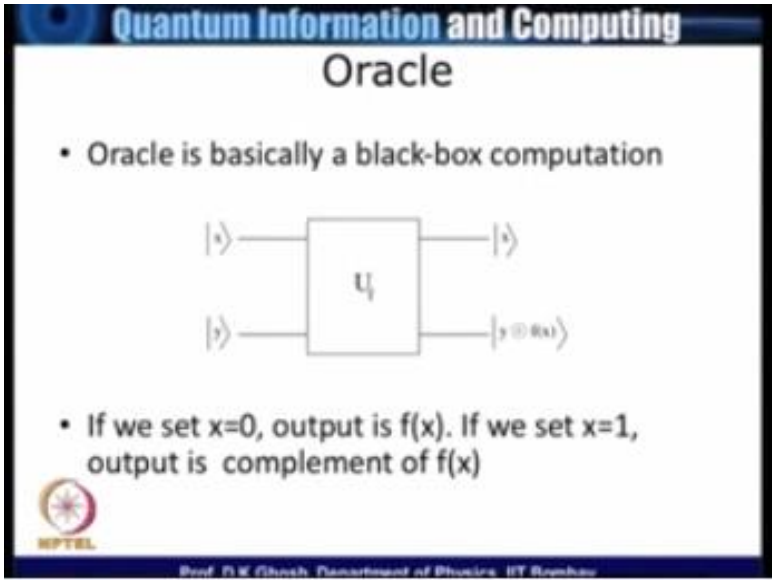
So suppose I am computing a function which you need not worry about what that function is at the moment all done by unitary operation the, what is there is the unitary operator computing this function is appearing at the center of this. Now one of the inputs here is x, now so the x is an input.

(Refer Slide Time: 20:35)



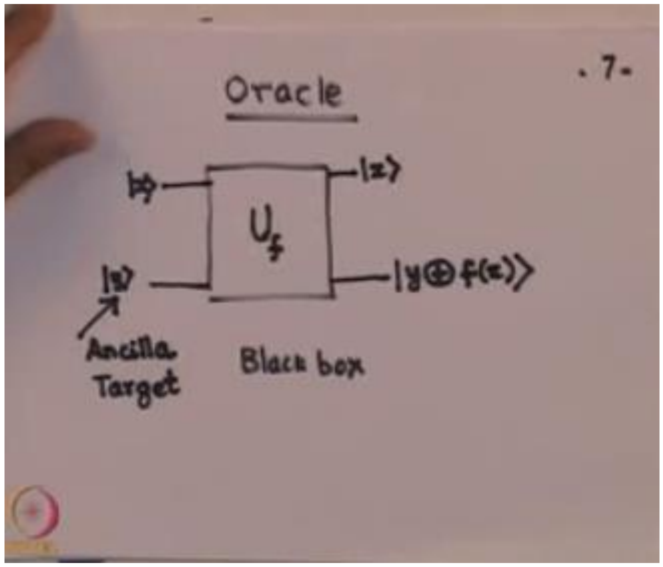
So let me look at the slide again.

(Refer Slide Time: 20:41)



Now there is a secondary or auxiliary occasionally called Ancilla this is why.

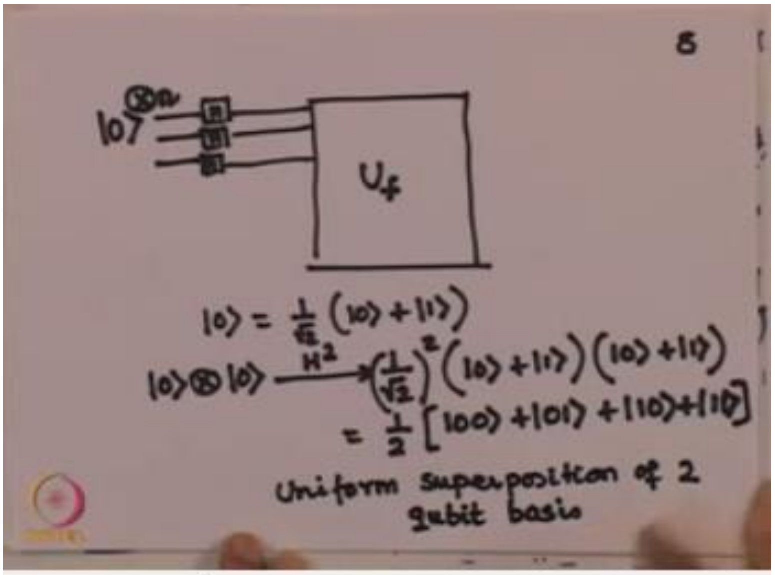
(Refer Slide Time: 20:48)



So this is Ancilla also that target bit, this is where the output will be there. Now this U_f function takes X at the input computes effects and output is here leaving the input the same. So this is unchanged and here what you get is the EXOR of y with the effects. So this is the way an oracle is represent, now you notice that supposing you said y=0 which means this is zero plus effects which by definition effects.

So therefore, we compute the effects itself, you should take y=1 this is one plus effects one EXOR effects which is nothing but the compliment of f(x). Now so this is how an oracle is represented, but that is not the only thing that it does. Remember that our biggest power of a quantum computing was the super position principle that is quantum parallelism. So the quantum parallelism says that the computation of effects corresponding to x need not be restricted to one particular value of x that is one particular state supposing instead of a single state I have a linear combination of states then my effects will be computed for every component of that linear super position, just to give you an example, that suppose the x that we talked about here.

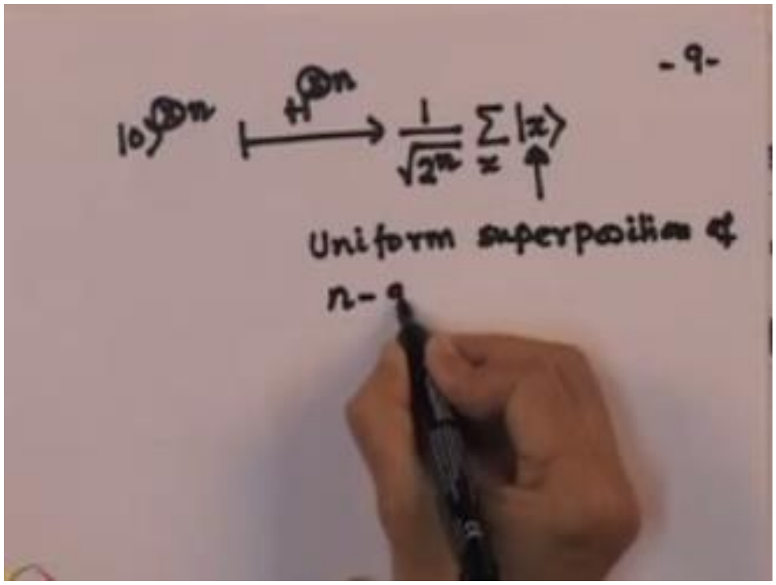
(Refer Slide Time: 22:59)



So the x that we talked about here supposing where it was an n qubit input. Now what happens to this n qubit input, if you send each one of these lines remember n qubit inputs is basically n number of lines, lines, normally if we write like this is a single line will do, now supposing each one of them I pass through a Hadamard gate, then what you get here is remember my zero becomes 1 over square root of 2, 0 + 1, so therefore for instance, what happens to 0 x 0 that is 0^2 pass through Hadamard gates.

So this will be 1 over root 2 square, 0 + 1 x 0+1. And what is this, this is equal to ½ 00+01+10+11 which is nothing but a linear superposition of 2 qubit basis states. Remember the single qubit basis states are 0 and 1, the 2 qubit basis states are 0, 0, 0, 1, 1, 0, and 1, 1 so this is the uniform superposition, uniform meaning each one of this comes with the same strength, uniform superposition of 2 qubit basis states.

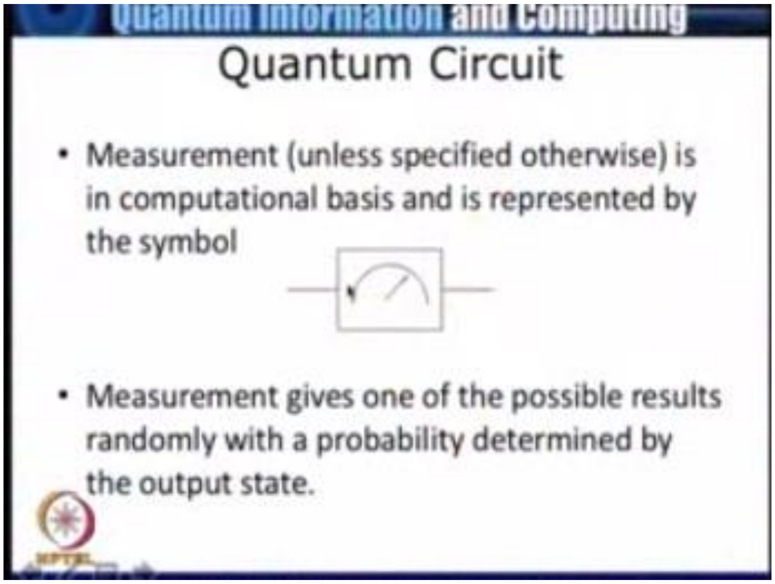
(Refer Slide Time: 25:05)



Now extending this if you have n number of zeros and pass them through Hadamard gate n of them, then what is one over square root of 2^n sum over x where this x that I have written down here is the uniform superposition of n qubit basis states. This is a very important result because this uniform superposition of n qubit basis state we will be using a quiet frequently in our discussion.

So therefore, suppose I passed my, this thing to Hadamard gate of which my input was the linear superposition then my, whatever function I am calculating is calculated corresponding to each one. The last component of a quantum circuits is the process of measurement. As I have said already that the process of measurement we assume to happen always in the computational basis look at the slide.

(Refer Slide Time: 26.48)



So the measurement process is represented by a meter like symbol and so notice that even if my input or we are measuring an output which is a linear combination when I make a measurement the measurement comes up with certain probability. So this gives you one of the possible results of the outputs. Now we will see that this has a lot of importance. So therefore, the quantum circuit consists of input, the logic gates, the oracle, and the measurement meters. So this is the complete quantum circuit and we will be talking about specific quantum circuits as we go along in this course.

NATIONAL PROGRAMME ON TECHNOLOGY
ENHANCED LEARNING
(NPTEL)

NPTEL
Principal Investigator
IIT Bombay

Prof. R.K. Shevgaonkar

Head CDEEP

Prof. V.M. Gadre

Producer

Arun kalwankar

Online Editor
& Digital Video Editor

Tushar Deshpande

Digital Video Cameraman
& Graphic Designer

Amin B Shaikh

Jr. Technical Assistant

Vijay Kedare

Teaching Assistants

Pratik Sathe
Bhargav Sri Venkatesh M.

Sr. Web Designer

Bharati Sakpal

Research Assistant

Riya Surange

Sr. Web Designer

Bharati M. Sarang

Web Designer

Nisha Thakur

Project Attendant

Ravi Paswan
Vinayak Raut

NATIONAL PROGRAMME ON TECHNOLOGY ENHANCED LEARNING (NPTEL)