Lecture - #8

Quantum Circuits

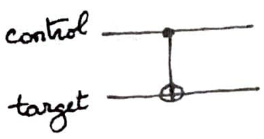
Last class, we encountered the following gates:

one qubit gates

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \qquad Y = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}, \qquad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

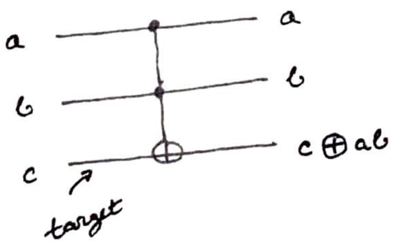
2-qubit gates



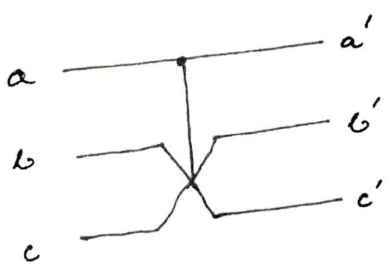
$$|x, y\rangle \longmapsto |x, y \oplus x\rangle$$

3-qubit gates

Toffoli gate / CCNOT gate



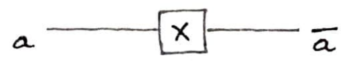
CSWAP gate:



Classical logic circuits implemented with quantum gates: ②

- CCNOT gate implements all classical logic gates

NOT Gate:



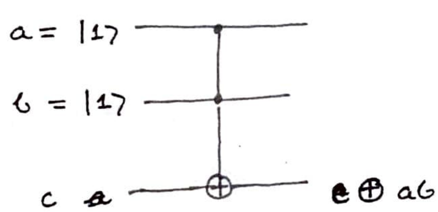
$$\text{NOT}(x) = \neg x = \begin{cases} 0 & x = 1 \\ 1 & x = 0 \end{cases}$$

$|0\rangle \leftrightarrow |1\rangle$
$|1\rangle \leftrightarrow |0\rangle$

Here $\neg x$ stands for negation

$X|x\rangle = |\neg x\rangle$
$= |\text{NOT}(x)\rangle$
$(x = 0, 1)$

NOT gate using a CCNOT gate:



$$U_{CCNOT} |1, 1, x\rangle = |1, 1, \neg x\rangle$$

XOR gate

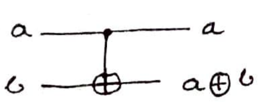
This classical operation has no inverse.

Classical XOR gate yields: $x, y \longmapsto x \oplus y \quad (x, y \in \{0, 1\})$

A	B	A ⊕ B
0	0	0
0	1	1
1	0	1
1	1	0

$0 \oplus 0 = 0$
$0 \oplus 1 = 1$
$1 \oplus 0 = 1$
$1 \oplus 1 = 0$

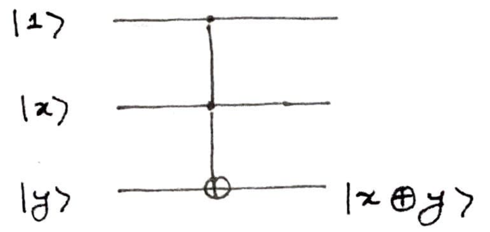
The quantum gate that does this operation is nothing but the CNOT gate



$$|i\rangle|j\rangle \longmapsto |i\rangle|i \oplus j\rangle$$

$$U_{XOR} = U_{CNOT} = |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes X$$

Note that the XOR gate can also be obtained from the CCNOT gate, as follows:



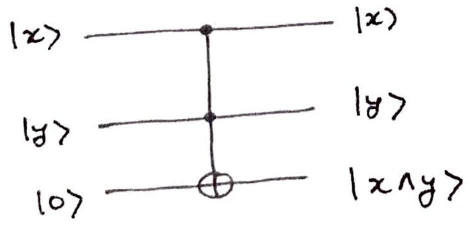
The first qubit is fixed at to $|1\rangle$.

$$U_{CCNOT} \, |1, x, y\rangle = |1, x, x \oplus y\rangle$$

AND gate

$$\text{AND}\,(x, y) \equiv x \wedge y \equiv \begin{cases} 1 & x = y = 1 \\ 0 & \text{otherwise} \end{cases} \qquad x, y \in \{0, 1\}$$

$$U_{AND} = \left(|00\rangle\langle 00| + |01\rangle\langle 01| + |10\rangle\langle 10|\right) \otimes I + |11\rangle\langle 11| \otimes X$$



$$U_{AND} \, |x, y, 0\rangle = |x, y, x \wedge y\rangle \, , \quad x, y \in \{0, 1\}$$

OR Gate

$$OR(x,y) = x \vee y = \begin{cases} 0 & x = y = 0 \\ 1 & \text{otherwise} \end{cases} \qquad x, y \in \{0,1\}$$

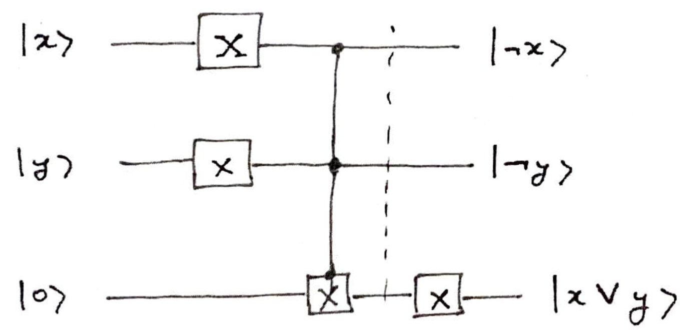
$$x \vee y = \neg(\neg x \wedge \neg y)$$

$\neg$: negation

(de Morgan theorem)

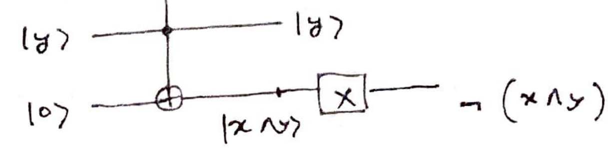
$$U_{OR} = |00\rangle\langle 11| \otimes X + |01\rangle\langle 10| \otimes X + |10\rangle\langle 01| \otimes X$$
$$+ \, |11\rangle\langle 00| \otimes I$$

$$U_{OR}|x, y, 0\rangle = |\neg x, \neg y, x \vee y\rangle \, , \quad x, y \in \{0,1\}$$



NAND gate

$$NAND(x,y) = \neg(x \wedge y) = \begin{cases} 0 \, , & x = y = 1 \\ 1 & \text{otherwise} \end{cases} \qquad x, y \in \{0,1\}$$



Quantum Circuits – must know basics :

(i) Inputs to the circuits are qubits, as are the outputs.

(ii) Unless expressed or stated otherwise, the qubits are in computational basis.

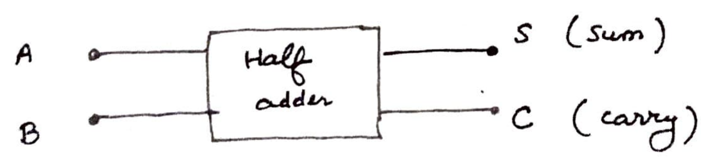
(iii) Looping in the circuit or Fan-inns are not permitted. Fan-out being a copying circuit is illegal in QC and Fan-in being its inverse is ruled by reversibility.

→ one cannot give several inputs giving rise to the same output

Quantum Half-adder

First look back, what a classical half-adder is!

Half adder is used to add single bit numbers. It does not take carry from previous sum.



Binary addition

$$\begin{array}{rl} 2^2\ 2^1\ 2^0 & \\ 1\ 1\ 0 & \rightarrow 1\times2^2+1\times2^1+0\times2^0 \\ +\ 1\ 0\ 1 & \rightarrow 1\times2^2+0\times2^1+1\times2^0 \\ \hline & \underbrace{1\times2^3+0\times2^2+1\times2^1\ \ 1\times2^0} \end{array}$$

$\underline{\underline{1011}}$

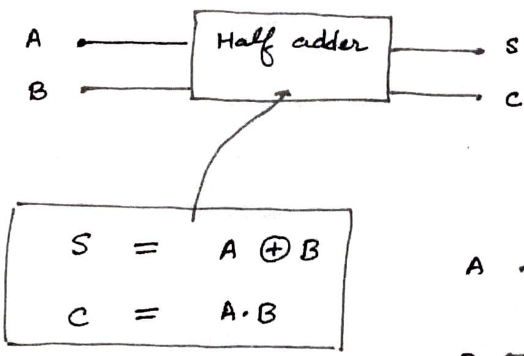
base = 2
0 and 1
sum ≤ 1

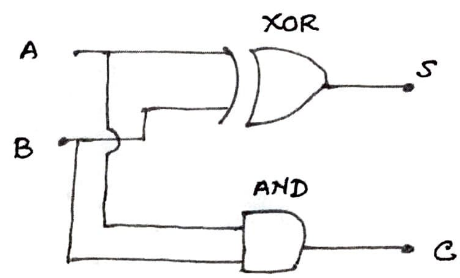
$1+1=2$
$2=(2+0)\times2^2$
$=2^3+0\times2^2$
$=1\times2^3+0\times2^2$

Truth Table :

~~A B S C 0 0 0 1~~

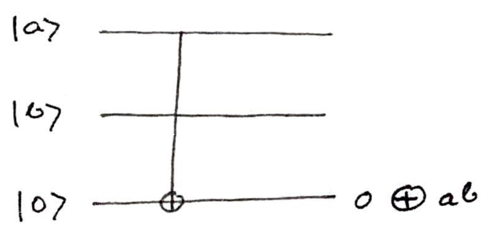
A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1





Now go back to quantum half-adder ckt !

Consider the following CCNOT gate :

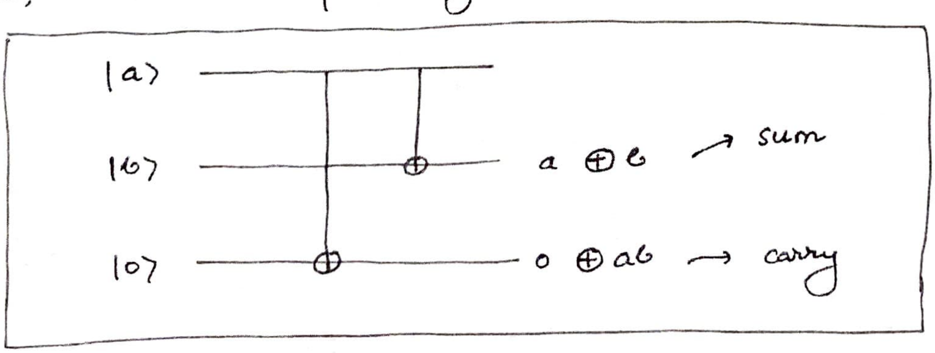


If $a = b = 1$, then only we have a carry, it is 1

$1 + 1 = 10$
carry

If $a = 0 = b$ or $a \neq 0 = b$ or $a = 0 \neq b$
we donot have a carry

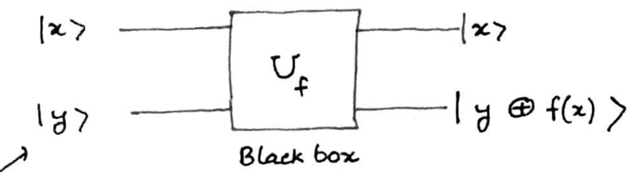
Now, consider the following:



≡ Half-adder

ORACLE

It has a very specific ~~purs~~ purpose in quantum computation. Very often, we want to compute some functions. In classical computing we call some subroutine program for the purpose. An oracle essentially does the same thing. Oracle takes certain amounts of inputs and computes the function and gives the output. Oracle is basically a black box computation.



ancilla or target bit

If we set $y=0$, output is $f(x)$

If we set $y=1$, output is complement of $f(x)$.

U_f takes $|x\rangle$ as the ~~ios~~ input and computes $f(x)$.

Say $|y=0\rangle$ then $|y \oplus f(x)\rangle = |f(x)\rangle$

$|y=1\rangle$ then $|1 \oplus f(x)\rangle =$ complement of $|f(x)\rangle$

complement of a function

De Morgan's theorem:

$$\overline{XY} = \bar{X} + \bar{Y}$$

$$\overline{X+Y} = \bar{X}.\bar{Y}$$

$$F = \bar{X}Y\bar{Z} + \bar{X}\bar{Y}Z$$

$$\bar{F} = \overline{\bar{X}Y\bar{Z} + \bar{X}\bar{Y}Z}$$

$$= \overline{\bar{X}Y\bar{Z}} \,.\, \overline{\bar{X}\bar{Y}Z}$$

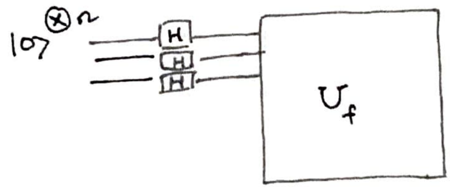
$$= (\bar{\bar{X}} + \bar{Y} + \bar{\bar{Z}})(\bar{\bar{X}} + \bar{\bar{Y}} + \bar{Z})$$

$$= (X + \bar{Y} + Z)(X + Y + \bar{Z})$$

Oracle does much more : e.g. it is very effective in the so-called quantum parallelism.

Suppose the input is a linear combination of states. Then $f(x)$ will be computed for each component of that linear superposition.

Lets say. input is a n-qubit input, each passed through a Hadamard gate.



$$|0\rangle \xmapsto{H} \frac{|0\rangle + |1\rangle}{\sqrt{2}}$$

consider 2 qubit case first.

$$|0\rangle \otimes |0\rangle \xrightarrow{H^2} \left(\frac{1}{\sqrt{2}}\right)^2 \left(|0\rangle + |1\rangle\right)\left(|0\rangle + |1\rangle\right)$$

$$= \frac{1}{2}\left(|00\rangle + |01\rangle + |10\rangle + |11\rangle\right)$$

This is nothing ~~to~~ but a linear superposition of 2 qubit basis states. This is also a uniform superposition of basis states.

Extending it to n number of $|0\rangle$ s

$$|0\rangle^{\otimes n} \xmapsto{H^{\otimes n}} \frac{1}{\sqrt{2^n}} \sum_x |x\rangle$$

Uniform superposition of n-qubit basis ~~to~~ states.

The Last ~~process in~~ component in Quantum circuit is the process of measurement. Measurement is always done, unless specified, in the computational basis. It is represented by the symbol