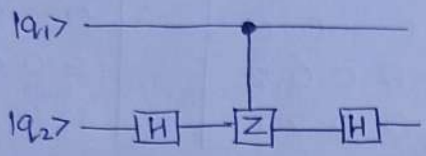
CS5100: Quantum Computing

Problem Set 2

C. Akshay Santoshi
CS21BTECH11012.

Q1.



When the first qubit (control qubit) is 0, applying the first Hadamard gate would not give a $|11\rangle$ basis state. So, controlled-Z gate would not ~~have~~ make any change. Since $H^{-1} = H$, applying the second Hadamard gate would just give back the original two-qubit state.

When the first qubit (control qubit) is 1, after applying the first Hadamard gate, controlled-Z gate would now have an effect on the basis state $|11\rangle$, where it changes it to $-|11\rangle$, just like how CNOT gate would have effect when controlled bit was 1. Applying the second Hadamard gate would rotate the target bit back to the standard basis $(|0\rangle, |1\rangle)$ mimicking the CNOT gate.

$\boxed{|00\rangle}$

1) $\frac{1}{\sqrt{2}}(|00\rangle + |01\rangle)$

2) $\frac{1}{\sqrt{2}}(|00\rangle + |01\rangle)$

3) $\frac{1}{2}(|00\rangle + |\cancel{01}\rangle + |00\rangle - |\cancel{01}\rangle)$
$= |00\rangle$.

$\boxed{|01\rangle}$

1) $\frac{1}{\sqrt{2}}(|00\rangle - |01\rangle)$

2) $\frac{1}{\sqrt{2}}(|00\rangle - |01\rangle)$

3) $\frac{1}{2}(|\cancel{00}\rangle + |01\rangle - |\cancel{00}\rangle + |01\rangle)$
$= |01\rangle$.

$\boxed{|10\rangle}$

1) $\frac{1}{\sqrt{2}}(|10\rangle + |11\rangle)$

2) $\frac{1}{\sqrt{2}}(|10\rangle - |11\rangle)$

3) $\frac{1}{2}(|\cancel{10}\rangle + |11\rangle - |\cancel{10}\rangle + |11\rangle)$
$= |11\rangle$

$\boxed{|11\rangle}$

1) $\frac{1}{\sqrt{2}}(|10\rangle - |11\rangle)$

2) $\frac{1}{\sqrt{2}}(|10\rangle + |11\rangle)$

3) $\frac{1}{2}(|10\rangle + |\cancel{11}\rangle + |10\rangle - |\cancel{11}\rangle)$
$= |10\rangle$

10

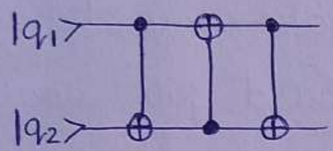
$(I \times H)\, CZ\, (I \times H) = U.$

$$\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & -1 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & -1 \end{bmatrix}$$

$$= \frac{1}{2}\begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & -1 \end{bmatrix}\begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & -1 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 2 \\ 0 & 0 & 2 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$= \text{CNOT gate}$

Q2.



After first CNOT gate (Here $|q_1\rangle$ is control qubit and $|q_2\rangle$ is target qubit):

$|q_1\rangle \rightarrow |q_1\rangle$

$|q_2\rangle \rightarrow |q_1 \oplus q_2\rangle$

After second CNOT gate (Here second qubit is control qubit and ~~third~~ first qubit is target qubit):

$|q_1\rangle \longrightarrow |q_1 \oplus (q_1 \oplus q_2)\rangle = |q_2\rangle.$

$|q_2\rangle \longrightarrow |q_1 \oplus q_2\rangle$

After third CNOT gate (Here first qubit is control qubit and second qubit is target qubit):

$|q_2\rangle \longrightarrow |q_2\rangle$

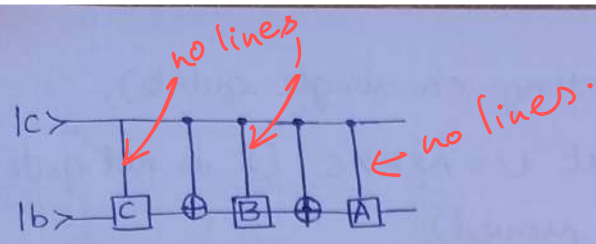
$|q_1 \oplus q_2\rangle \longrightarrow |q_2 \oplus (q_1 \oplus q_2)\rangle = |q_1\rangle.$

So finally, $|q_1\rangle \rightarrow \cancel{|q_2\rangle} \rightarrow\rightarrow |q_2\rangle$

$|q_2\rangle \rightarrow \rightarrow\rightarrow |q_1\rangle.$

10

Q3.



<u>When c=0:</u>

Applying C would give $|c\rangle \otimes C|b\rangle$

Applying CNOT would not ~~have~~ make change since $c=0$.

Applying B would give $|c\rangle \otimes BC|b\rangle$.

Applying CNOT would not make change since $c=0$.

Applying A would give $|c\rangle \otimes ABC|b\rangle$.

We know that $ABC = I$, therefore final state would be $\underline{|cb\rangle}$

<u>When c=1:</u>

Applying C would give $|c\rangle \otimes C|b\rangle$.

Applying CNOT would give $|c\rangle \otimes XC|b\rangle$

Applying B would give $|c\rangle \otimes BXC|b\rangle$

Applying CNOT would give $|c\rangle \otimes XBXC|b\rangle$

Applying A would give $|c\rangle \otimes AXBXC|b\rangle$.

We know that $AXBXC = U$, therefore final state would be

$\underline{|c\rangle \otimes U|b\rangle}$.

Therefore the given circuit implements the controlled U gate $|c\rangle|b\rangle \longrightarrow |c\rangle U^c|b\rangle$.

10

Q4
~~Q5~~.

Given a quantum circuit, C, which has CNOTs and single-qubit gates only.

To implement C in a controlled way, each gate has to be replaced with its controlled part.

For
Every ~~single~~ unitary ~~qubit~~ gate, U (acting on single qubits),
there exists A, B, C such that $U = AXBXC$ (X is not gate)
and $ABC = I$. (Need to be proved).

From question 3, we saw that the controlled U-gate (single qubit) can be implemented using atmost three single qubit gates and two CNOT gates.

So for controlled U-gate, there is a constant overhead. So $O(1)$ gates.

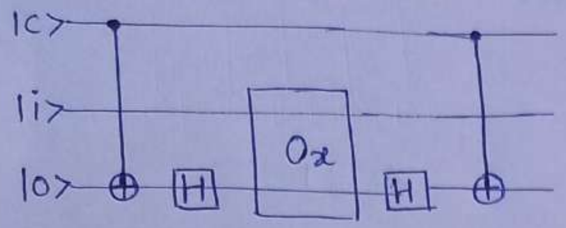
Since C contains $O(T)$ gates, and some of these are single-qubit gates, the overall cost for controlling all single-qubit gates will be $O(T)$ gates.

Control for CNOT gates is Toffoli gates (CCNOT). This also is proportional to $O(1)$ gates for each CNOT in original circuit C.

The total no. of gates required to implement the controlled version of circuit C is $O(T)$ where T is the no. of gates in given circuit, C.

10

Q5.



Initial State : $|c, i, 0\rangle$.

Apply CNOT gate (Control ~~bit~~ qubit as first qubit and target qubit as third qubit):

State becomes $|c, i, 0 \oplus c\rangle = |c, i, c\rangle$.

Apply Hadamard on third qubit :

State becomes:

If $c=0$: $|c, i, \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\rangle$

If $c=1$: $|c, i, \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)\rangle$

~~|c, i, H|c>~~ $|c, i\rangle \otimes H|c\rangle$

Apply the query O_x:

$O_x : |i, b\rangle \longrightarrow |i, b \oplus x_i\rangle$

The state becomes:

	$x_i = 0$	$x_i = 1$
$c=0$	$|c, i, \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\rangle$	$|c, i, \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\rangle$
$c=1$	$|c, i, \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)\rangle$	$|c, i, \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)\rangle$

So if $c=0$, then state is unchanged

if $c=1$, a phase difference of $(-1)^{x_i}$ is added.

Thus combining, a phase difference of $(-1)^{cx_i}$ is added.

Apply Hadamard on third qubit :

State becomes :

	$x_i = 0$	$x_i = 1$
$c=0$	$\|c, i, 0\rangle$	$\|c, i, 0\rangle$
$c=1$	$\|c, i, 1\rangle$	$\|c, i\rangle \otimes -\|1\rangle$

10

We get $(-1)^{cx_i} |c, i, c\rangle$

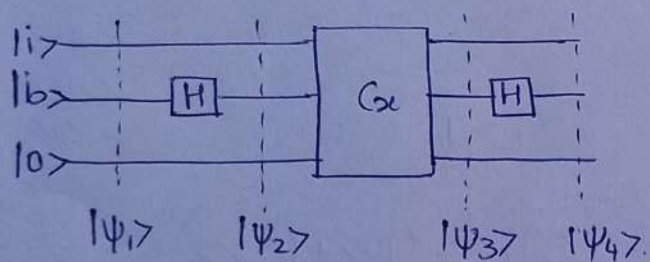
Apply CNOT gate : (with control qubit as first gate and target qubit as third gate) :

Final state would be : $(-1)^{cx_i} |c, i, 0\rangle$.

Q6. We can take the initial state as $|i\rangle |b\rangle |0\rangle$ where $|0\rangle$ is the auxiliary qubit.

$C_x : |i, b, 0\rangle \longrightarrow (-1)^{bx_i} |i, b, 0\rangle$.

$O_x : |i, b\rangle \longrightarrow |i, b \oplus x_i\rangle$.



$|\psi_1\rangle = |i, b, 0\rangle$.

$|\psi_2\rangle = |i, \frac{1}{\sqrt{2}}(|0\rangle + (-1)^b |1\rangle), 0\rangle$.

$|\psi_3\rangle$:

$|i, 0, 0\rangle \longrightarrow |i, 0, 0\rangle$

$|i, 1, 0\rangle \longrightarrow (-1)^{x_i} |i, 1, 0\rangle$.

$$|\psi_3\rangle = |i, \tfrac{1}{\sqrt{2}}(|0\rangle + (-1)^{b}(-1)^{x_i}|1\rangle), 0\rangle$$

$$\cancel{|\psi_4\rangle = \text{If } b} \quad = |i, \tfrac{1}{\sqrt{2}}(|0\rangle + (-1)^{b\oplus x_i}|1\rangle), 0\rangle$$

$$\cancel{|\psi_4\rangle \text{ if } b\oplus x_i = 1, \ |i, \tfrac{1}{\sqrt{2}}(|0\rangle}$$

$|\psi_4\rangle$

$$|i, \tfrac{1}{\sqrt{2}}\left(\tfrac{1}{\sqrt{2}}(|0\rangle+|1\rangle) + (-1)^{b\oplus x_i} \times \tfrac{1}{\sqrt{2}}(|0\rangle-|1\rangle)\right), 0\rangle$$

$$= |i, \tfrac{1}{2}(|0\rangle+|1\rangle + (-1)^{b\oplus x_i}|0\rangle - (-1)^{b\oplus x_i}|1\rangle), 0\rangle$$

If $b\oplus x_i = 1$,

$$|i, \tfrac{1}{2}(|0\rangle+|1\rangle - |0\rangle + |1\rangle), 0\rangle$$

$$= |i, 1, 0\rangle = |i, b\oplus x_i, 0\rangle$$

If $b\oplus x_i = 0$,

$$|i, \tfrac{1}{2}(|0\rangle+|1\rangle+|0\rangle-|1\rangle), 0\rangle$$

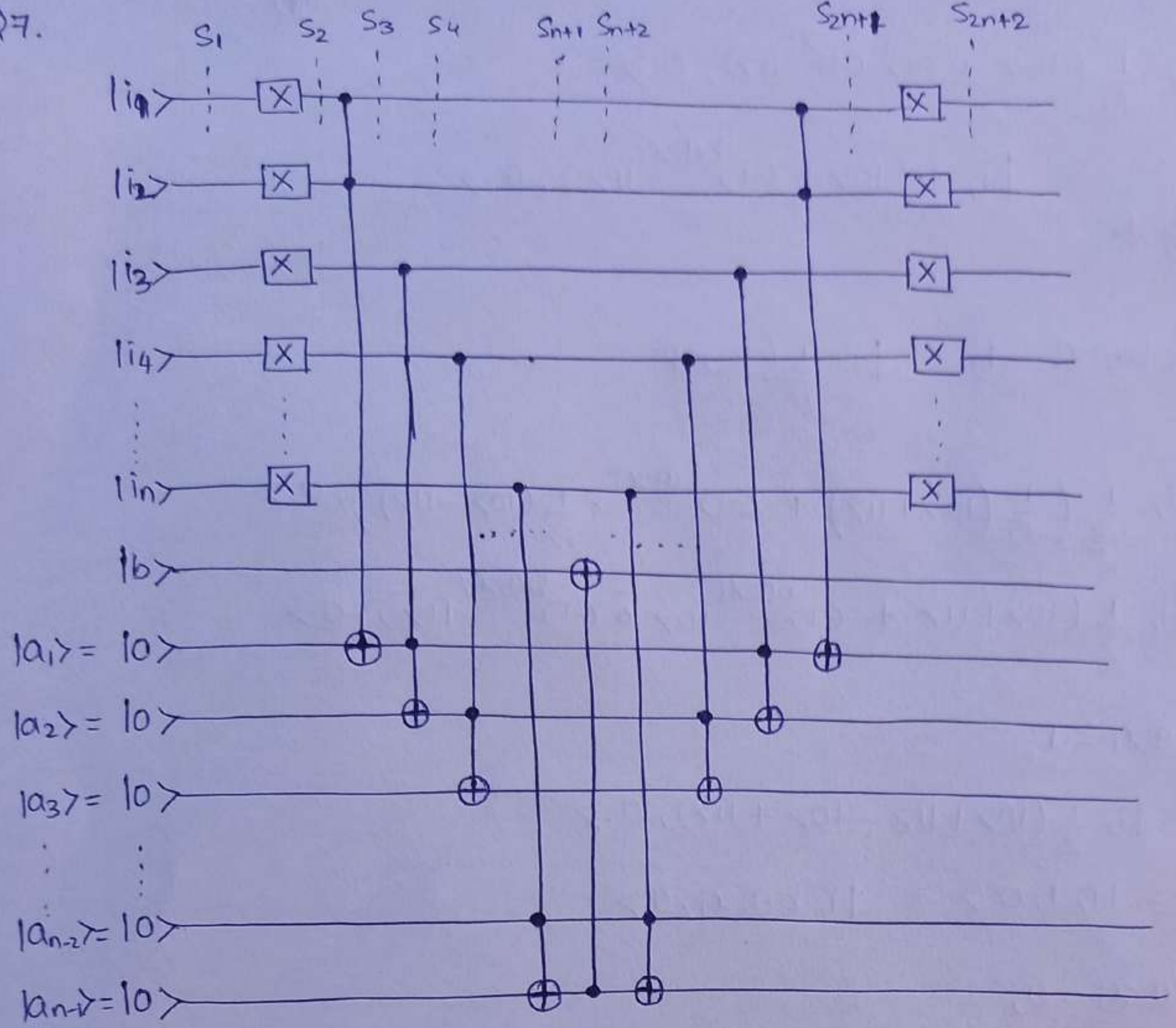
$$= |i, 0, 0\rangle = |i, b\oplus x_i, 0\rangle.$$

State would be $|i, b\oplus x_i, 0\rangle$.

Therefore this circuit implements the standard query O_x $|i, b, 0\rangle \to |i, b\oplus x_i, 0\rangle$ using one controlled-phase query to x.

10

Q7.



→ We have used $n-1$ $|0\rangle$ auxiliary qubits as shown.

→ First, apply NOT gate (X) to the first n qubits ($|i\rangle$).
State of these qubits would be changed as follows.
$|\bar{i}_1, \bar{i}_2, \ldots, \bar{i}_n\rangle$.

→ Next, we use the auxiliary qubits as target qubits for Tofolli gates to check if the string i was initially 0^n or not. It implements the AND functionality.

First Tofolli-gate: Control: ~~First~~ $|i_1\rangle, |i_2\rangle$
Target: First auxiliary qubit.

Second Toffoli-gate: Control: Third address qubit and first auxiliary qubit
Target: Second auxiliary qubit.

⋮

$(n-1)^{th}$ Toffoli gate: Control: $|i_n\rangle$ and $(n-2)^{th}$ auxiliary qubit.

Target : $(n-1)^{th}$ auxiliary qubit (last ~~qubit~~ one) 5

→ Apply CNOT gate with control bit as $(n-1)^{th}$ auxiliary qubit and target as $|b\rangle$.

If $|i\rangle$ had been $|0^n\rangle$, the last auxiliary qubit would have become $|1\rangle$ by now.

If $|i\rangle$ had been anything other than $|0^n\rangle$, the auxiliary qubit would remain unchanged.

Applying CNOT would change $|b\rangle$ to $|1 \oplus b\rangle$ if last auxiliary qubit is 1. (i.e. $|i\rangle$ was initially $|0^n\rangle$)

Else $|b\rangle$ would remain unchanged.

→ Now, to put back the auxiliary qubits to $|0\rangle$ and get the initial $|i\rangle$, reverse the circuit, i.e, undo all of the Toffoli gates and then apply X gates to the address qubits corresponding to $|i\rangle$.

From circuit diagram, we can write

$S_1 = |i_1, i_2, i_3, \ldots, i_n, b, 0^{n-1}\rangle$

$S_2 = |\bar{i}_1, \bar{i}_2, \bar{i}_3, \ldots, \bar{i}_n, b, 0^{n-1}\rangle$

$S_3 =$ if $|\bar{i}_1\rangle$ and $|\bar{i}_2\rangle$ are $|1\rangle$, then $|\bar{i}_1, \bar{i}_2, \bar{i}_3, \ldots, \bar{i}_n, b, 1, 0^{n-2}\rangle$

else $|\bar{i}_1, \bar{i}_2, \bar{i}_3, \ldots, \bar{i}_n, b, 0^{n-1}\rangle$.

$S_4 =$ if $|\bar{i}_3\rangle = |1\rangle$ and $|a_1\rangle = |1\rangle$, then $|\bar{i}_1, \bar{i}_2, \bar{i}_3, \ldots, \bar{i}_n, b, 1, 1, 0^{n-3}\rangle$

if $|\bar{i}_3\rangle = |1\rangle$ and $|a_1\rangle = |0\rangle$, then $|\bar{i}_1, \bar{i}_2, \bar{i}_3, \ldots \bar{i}_n, b, 0^{n-1}\rangle$

if $|\bar{i}_3\rangle = |0\rangle$ and $|a_1\rangle = |1\rangle$, then $|\bar{i}_1, \bar{i}_2, \bar{i}_3, \ldots, \bar{i}_n, b, 1, 0^{n-2}\rangle$.

else $|\bar{i}_1, \bar{i}_2, \bar{i}_3, \ldots, \bar{i}_n, b, 0^{n-1}\rangle$.

$S_{n+1} =$ if $i = 0^n$, then $|\bar{i}_1, \bar{i}_2, \bar{i}_3, \ldots, \bar{i}_n, b, 1^{n-1}\rangle$.

$S_{n+2} =$ if $i = 0^n$, then $|\bar{i}_1, \bar{i}_2, \bar{i}_3, \ldots, \bar{i}_n, 1, 1^{n-1}\rangle$

else $|\bar{i}_1, \bar{i}_2, \bar{i}_3, \ldots, \bar{i}_n, 0, 1^k, 1^{n-k-1}\rangle$ where k depends on $|i\rangle$.

S_{2n+1} = if $i = 0^n$, then $|\bar{i}_1, \bar{i}_2, \ldots, \bar{i}_n, 1, 0^{n-1}\rangle$
else $|\bar{i}_1, \bar{i}_2, \ldots, \bar{i}_n, 0, 0^{n-1}\rangle$.

S_{2n+2} = if $i = 0^n$, then $|i_1, i_2, \ldots, i_n, 1, 0^{n-1}\rangle$
else $|i_1, i_2, \ldots, i_n, 0, 0^{n-1}\rangle$.

10

Q8. Circuit for this would be

Circuit from Q7 and at the end you add a U_f query gate (query to x) which takes as input $|i_1\rangle, |i_2\rangle, \ldots, |i_n\rangle$ as well as $|b'\rangle$ (total n+1 qubits).

$|b'\rangle$ ↓ the $(n+1)^{th}$ qubit in the circuit diagram.

$|i_1\rangle, |i_2\rangle, \ldots |i_n\rangle$ are used to get the value of x_i from query and then then x_i is XORed with $|b'\rangle$.

$$|i, b'\rangle \longrightarrow |i, b' \oplus x_i\rangle.$$

Notice ~~Notice~~ that when $|i\rangle$ is $|0^n\rangle$, $|b'\rangle$ is $|1 \oplus b\rangle$ (from prev question implementation) and ~~in any~~ if $|i\rangle$ is any other string ($\neq 0^n$), $|b'\rangle$ is $|b\rangle$.

So, in this circuit, when $|i\rangle$ is $|0^n\rangle$, the output would be $|i\rangle|1 \oplus b \oplus x_i\rangle$.

Else, it would be $|i\rangle|b \oplus x_i\rangle$.

Since, x' is the input x with its first bit flipped (from question) we want that when you query for any bit, it give

$$|i\rangle|b \oplus x'_i\rangle.$$

This, is nothing but as follows:

⑥

When you query for the first bit, it should give

$|0^n, b \oplus x'_i\rangle = |0^n, b \oplus x_i \oplus 1\rangle$ (since first bit is flipped in x')

$\vdots$

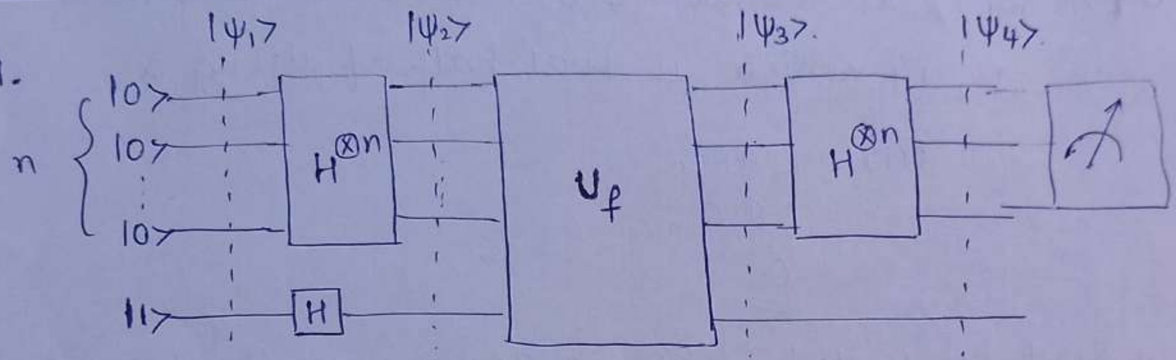
When you query for any other bit, it should give

$|i, b \oplus x'_i\rangle = |i, b \oplus x_i\rangle$ (Rest all bits are same as in x_i for x'_i).

Our circuit is implementing the query to x' by using one query to x.

10

Q9.



U_f is the oracle for the function $f(i) = x_i \oplus i_0$ where x_i is the value of the bit from x (input string) at index i and i_0 is the first bit of the binary string.

from left or right?

$$|\psi_1\rangle = |0^n\rangle|1\rangle$$

$$|\psi_2\rangle = \frac{1}{\sqrt{2^n}} \sum_{i \in \{0,1\}^n} |i\rangle \otimes \left(\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)\right)$$

$$|\psi_3\rangle = \frac{1}{\sqrt{2^n}} \sum_{i \in \{0,1\}^n} |i\rangle \otimes \left(\frac{1}{\sqrt{2}}(|0 \oplus f(i)\rangle - |1 \oplus f(i)\rangle)\right)$$

$$= \frac{1}{\sqrt{2^n}} \sum_{i \in \{0,1\}^n} (-1)^{f(i)} |i\rangle \otimes \left(\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)\right)$$

$$|\psi_4\rangle = \frac{1}{2^n} \left(\sum_{j \in \{0,1\}^n} |j\rangle \sum_{i \in \{0,1\}^n} (-1)^{f(i) + i \cdot j} \right) \otimes \left(\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)\right)$$

Ignore the second register for now.

Case 1 : If first $N/2$ bits are all zero and ~~sec~~ second $N/2$ bits are all 1 ~~$x=0$~~

$$x = 0^{N/2} 1^{N/2}$$

$$x_i = \begin{cases} 0, & \text{if } i \in \{0,1,\dots N/2-1\} \\ 1, & \text{if } i \in \{N/2,\dots,N-1\} \end{cases}$$

When first bit i_0 is 0 (which means x is of the form $0 i_1 i_2 \dots i_{n-1}$), x_i would be 0, because in this case $i \in \{0,1,\dots,N/2-1\}$ which is first half of string x.

$$\begin{aligned} f(i) &= x_i \oplus i_0 \\ &= 0 \oplus 0 \\ &= 0 \end{aligned}$$

When first bit i_0 is 1 (which means x is of the form $1 i_1 i_2, \dots i_{n-1}$), x_i would be 1, because in this case $i \in \{\cancel{0,1,}\ N/2, \dots N-1\}$ which is second half of string x.

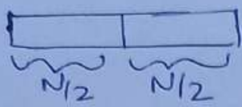
$$f(i) = x_i \oplus i_0 = 1 \oplus 1 = 0$$

In case 1, first register would look like

$$\frac{1}{2^n} \sum_{i \in \{0,1\}^n} \sum_{j \in \{0,1\}^n} (-1)^{i \cdot j} |j\rangle.$$

This is similar to the constant x_i case of Deutsch-Jozsa algorithm and hence when we measure $|j\rangle$, it would give $|0^n\rangle$ with 100% probability.
Rest all states would have destructive interference which gets balanced and hence would have zero-amplitude.

Case 2: When the no. of 1s in the first half of x plus the no. of 0s in the second half equals $N/2$.



Let us say, there are k 1's in the first half, then there would be $\left(\frac{N}{2}-k\right)$ 0's in the first half.

From the condition, there would be $\left(\frac{N}{2}-k\right)$ 0's in the second half, and so, k 1's in the second half.

$$\frac{1}{2^n}\left(\sum_{j\in\{0,1\}^n} |j\rangle \sum_{i\in\{0,1\}^n} (-1)^{f(i)+i\cdot j}\right).$$

Let us see see what the amplitude of $|0^n\rangle$ might be in this case

$$\frac{1}{2^n}\left(\sum_{i\in\{0,1\}^n} (-1)^{f(i)} |0^n\rangle\right).$$

When i is in first half ($i_0=0$)

$f(i) = x_i \oplus 0$

$x_i = 1$ k times

$x_i = 0$ $\left(\frac{N}{2}-k\right)$ times.

$f(i) = 1$ k times

0 $\frac{N}{2}-k$ times

When i is in second half ($i_0=1$)

$f(i) = x_i \oplus 1$.

$x_i = 1$ k times

$x_i = 0$ $\left(\frac{N}{2}-k\right)$ times

$f(i) = 0$ k times

1 $\left(\frac{N}{2}-k\right)$ times.

3

$\sum_{i\in\{0,1\}^n} (-1)^{f(i)}$ would become

$$(-1)^1 k + (-1)^0 \cdot \left(\frac{N}{2}-k\right) + (-1)^0 k + (-1)^1 \left(\frac{N}{2}-k\right)$$

$$= -k + \frac{N}{2} - k + k - \frac{N}{2} + k$$

$$= 0$$

How do you implement the oracle for $f(i) = x_i + i_0$?

Therefore $|0^n\rangle$ would never occur in second case.

Hence, when we measure first register, if we get $|0^n\rangle$, then it is case 1, any other state ($\neq |0^n\rangle$) would indicate case 2.

Q10. Let us write how the states would look like at each ~~step~~ of step of Simon's Algorithm.

Initial : $|0^n\rangle|0^n\rangle$.

Hadamard to first n qubits : $\frac{1}{\sqrt{2^n}} \sum_{i\in\{0,1\}^n} |i\rangle |0^n\rangle$.

Query would turn it into : $\frac{1}{\sqrt{2^n}} \sum_{i\in\{0,1\}^n} |i\rangle |x_i\rangle$.

Again apply Hadamard on first n qubits :

$$\frac{1}{2^n} \sum_{i\in\{0,1\}^n} \sum_{j\in\{0,1\}^n} (-1)^{i\cdot j} |j\rangle |x_i\rangle \quad \text{—— (1).}$$

Let the unknown subspace be $V \subseteq \{0,1\}^n$

If we take a vector $v_1 \in V$, notice that if R is the value of

$$R = \frac{1}{2^n} \sum_{i\in\{0,1\}^n} \sum_{j\in\{0,1\}^n} (-1)^{i\cdot j}$$

$i \oplus v_1$ means i is shifted by v_1.

So, we can also write

$$R = \frac{1}{2^n} \sum_{i\in\{0,1\}^n} \sum_{j\in\{0,1\}^n} (-1)^{(i\oplus v_1)\cdot j}.$$

Since v_1 is arbitrary, this holds true for any $v \in V$.

So, when we sum up for all $v \in V$, we would get

$$|V| R = \frac{1}{2^n} \sum_{i\in\{0,1\}^n} \sum_{j\in\{0,1\}^n} \sum_{v\in V} (-1)^{(i\oplus v)\cdot j}$$

$$R = \frac{1}{2^n \cdot |V|} \sum_{i\in\{0,1\}^n} \sum_{j\in\{0,1\}^n} \sum_{v\in V} (-1)^{(i\oplus v)\cdot j}$$

So, (1) can be written as

$$\frac{1}{2^n} \cdot \frac{1}{|V|} \sum_{i\in\{0,1\}^n} \sum_{j\in\{0,1\}^n} \sum_{v\in V} (-1)^{(i\oplus v)\cdot j} |j\rangle |x_i\rangle.$$

$$= \frac{1}{2^n} \cdot \frac{1}{|V|} \sum_{i\in\{0,1\}^n} \sum_{j\in\{0,1\}^n} (-1)^{i\cdot j} \left(\sum_{v\in V} (-1)^{v\cdot j} \right) |j\rangle |x_i\rangle.$$

If there exists $v_1 \in V$, such that $v_1 \cdot j = 1$, then

$$\sum_{v\in V} (-1)^{v\cdot j} = \frac{1}{2} \sum_{v\in V} \left((-1)^{v\cdot j} + (-1)^{(v\oplus v_1)\cdot j} \right)$$

$$= \frac{1}{2} \sum_{v\in V} \left((-1)^{v\cdot j} \cancel{} + (-1)^{v\cdot j} (-1)^{v_1\cdot j} \right).$$

$$\sum_{v \in V} (-1)^{v.j} = \frac{1}{2} \sum_{v \in V} (-1)^{v.j} (1 + (-1)^{v_1 . j})$$

$$= \underline{\underline{0}} \quad (\text{since } v_1 . j = 1)$$

If $j \in V^{\perp}$, $j.v = 0 \bmod 2$ for all $v \in V$.

$$\sum_{v \in V} (-1)^{v.j} = |V|$$

So, the state would be $\frac{1}{2^n} \cdot \frac{1}{|V|} \sum_{i,j \in \{0,1\}^n} (-1)^{i.j} |j\rangle |x_i\rangle$.

Thus, after the final measure of the first n qubits, (after 1 run of Simon's algorithm), we would get a $j \in \{0,1\}^n$, $j \in V^{\perp}$, i.e. it is orthogonal to the whole subspace ($j.v = 0 \bmod 2$ for every $v \in V$).

10