QP-3 : Bernstein-Vazrani Algorithm

- Name : **Jayant Som**
- Contact : **jsom@buffalo.edu | 716-348-7708**

---

1. Quantum Circuit creation

i) Imports :

In [1]:
```
# Importing the QuantumCircuit class from Qiskit
# The QuantumCircuit class is used to create quantum circuits
from qiskit import QuantumCircuit

# Importing the numpy library
# Numpy is used for working with arrays and perform numerical operations
import numpy as np
```

ii) Quantum circuit creation and adding different components:

In [2]:
```
# Number of qubits
# It should be greater than 5, so I am taking 6
n = 6
```

In [3]:
```
# Generate the bit string "s"
# Random integer in [0,.....2^nqubits - 1]
s = np.random.randint(2**n)
# Converting to binary string
s_str = format(s, '0' + str(n) + 'b')
print("Random integer:", s)
print("Secret string:", s_str)
```

```
Random integer: 26
Secret string: 011010
```

In [4]:
```
# Creating a Quantum Circuit
# n+1 qubits in x-register
# n classical bits in y-register for measurement
circuit = QuantumCircuit(n + 1, n)

# Initializing the y-register to |-> state
# So, applying X and H to |0>
circuit.x(n)
circuit.h(n)

# Applying H to all x-register qubits
```

```
# H on qubit i one by one
# Used to create superposition of all binary states |0> to |+>.
for i in range(n):
    circuit.h(i)

# Barrier : to separate the oracle from rest of the circuit
circuit.barrier()

# Building an Oracle
# with CNOT's from bit-'s'(if s=1) in x-register to y-register in |->
for i in range(n):
    # if ith bit of 's' is not 0
    if s & (1 << i):
        # CNOT from qubit i (x-register) to qubit n (y-register)
        # Control is i, Target is n
        circuit.cx(i, n)

# Barrier : to separate the oracle from rest of the circuit
circuit.barrier()

# Apply H to all x-register qubits
# H on qubit i one by one
# Used to convert the superposition state to computational basis state s
for i in range(n):
    circuit.h(i)

# Barrier : to separate the rest of the circuit
circuit.barrier()

# Measuring all x-register qubits
for i in range(n):
    # Measuring qubit i
    # Storing result in classical bit i
    circuit.measure(i, i)
```

iii) Circuit diagram :

In [5]:

```
# The draw method is used to visualize the quantum circuit.
# I am drawing the circuit using the 'mpl' output and 'iqp' style
circuit.draw(output='mpl', style='iqp')
```

Out[5]:

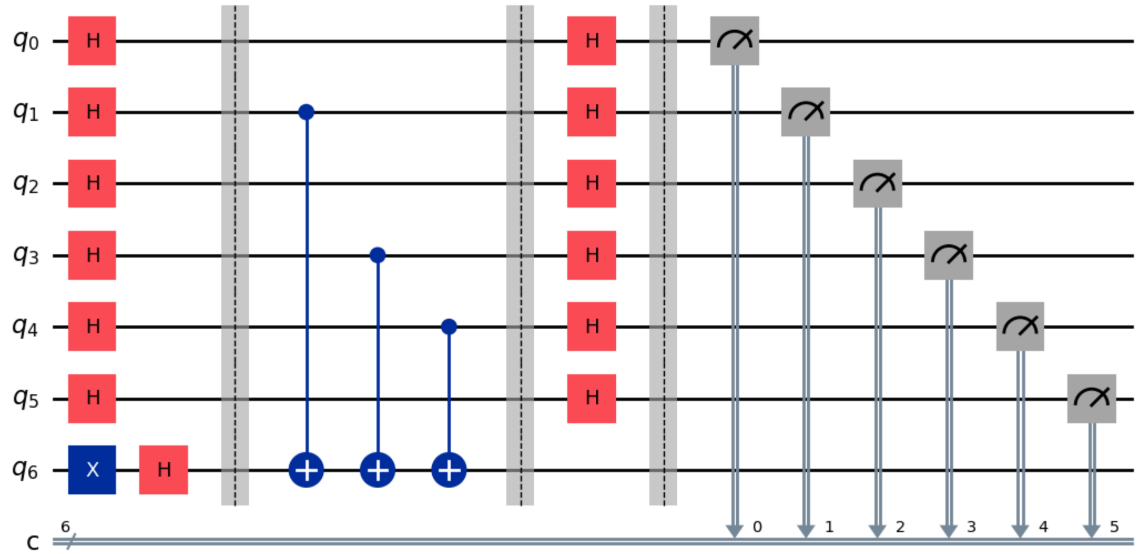


Fig 01 : Quantum Circuit for the Bernstein-Vazrani Algorithm

The above circuit diagram comprises of following notations and components :

Circuit Notations :

- **qo, q1, q2, q3, q4, q5** are the 6 qubits represented by the solid horizontal line. (x-registers)
- **q6** is the 7th qubit represented by the solid horizontal line. (y-register)
- **c** is the classical bit after measurement which is represented by the double lines.
- **6/** above the double lines represents the no. of classical data bits measurements (0,1,2,3,4,5).

Note : In Qiskit, the initial state is |q5 q4 q3 q2 q1 q0> where q5 the most significant bit and q0 is the least significant bit.

Circuit Components (left to right according to time step) :

- **H on q0, q1, q2, q3, q4 and q5** in orange box represents the Hadamard gate. It converts initial state |0> to |+>, so it produces superposition on all x-registers.

- **X on q6** in blue box represents the X gate. Here, it inverts intital state |0> to |1>.

- **H on q6** in orange box represents the Hadamard gate. Here, it converts |1> to |-> on the y-register.

- **CNOT** denoted by dark-blue line where **.** is the control and **+** is the target. There can be multiple CNOT gates in the circuit depending upon the secret string. This group of CNOT gates is called the **Oracle**.

***Note** : The condition in the program **if s & (1 << i)**: ensures that a CNOT gate is applied only if the ith bit of s is 1. If the ith bit is 1, then a CNOT gate is added with qubit i (from the x-register) as the control and qubit n (the y-register) as the target.*

- **Barrier** denoted by dotted lines separates the oracle from rest of the circuit.

- **H on q0, q1, q2, q3, q4 and q5** in orange box represents the Hadamard gate. It converts the superposition state to computational basis state. This is used to obtain the secret state s again.

- **Meters** in gray boxes represents the Measurement operation on q0, q1, q2, q3, q4 and q5.

2. Simulation code and output

i) Imports :

In [6]:
```
# The qiskit_aer library provides backend quantum simulators
# I am importing the Aer module which contains various type of simulators.
from qiskit_aer import Aer

# I am importing the transpile function from the qiskit library
# Transpile function is required to ensure that my circuit
# is able to run on the simulator.
from qiskit import transpile

# Importing the plot_histogram function from qiskit
# It used to visualize the simulation result.
from qiskit.visualization import plot_histogram
```

ii) Getting the Simulator and running it

In [7]:
```
# The qasm simulator runs the circuit and its result is classical bits.
simulator = Aer.get_backend("qasm_simulator")

# Transpile transforms the circuit to something appropriate for the machine.
# I am transpiling my circuit for the backend qasm simulator
sim_circuit = transpile(circuit, backend = simulator)

# The run method in the simulator executes the transpiled circuit.
# I am running the trial 4096 times.
job_sim = simulator.run(sim_circuit, shots = 4096)
```

iii) Fetching the result and plotting histogram :

In [8]:
```
# I am fetching the results of the simulation job execution.
# This result contains the counts of each measurement outcome.
```

```
result_sim = job_sim.result()

# result.get_counts() method is used to find the count of different outcomes
# I am generating and displaying a histogram of the simulation outcomes.
plot_histogram(result_sim.get_counts(circuit))
```

Out[8]:

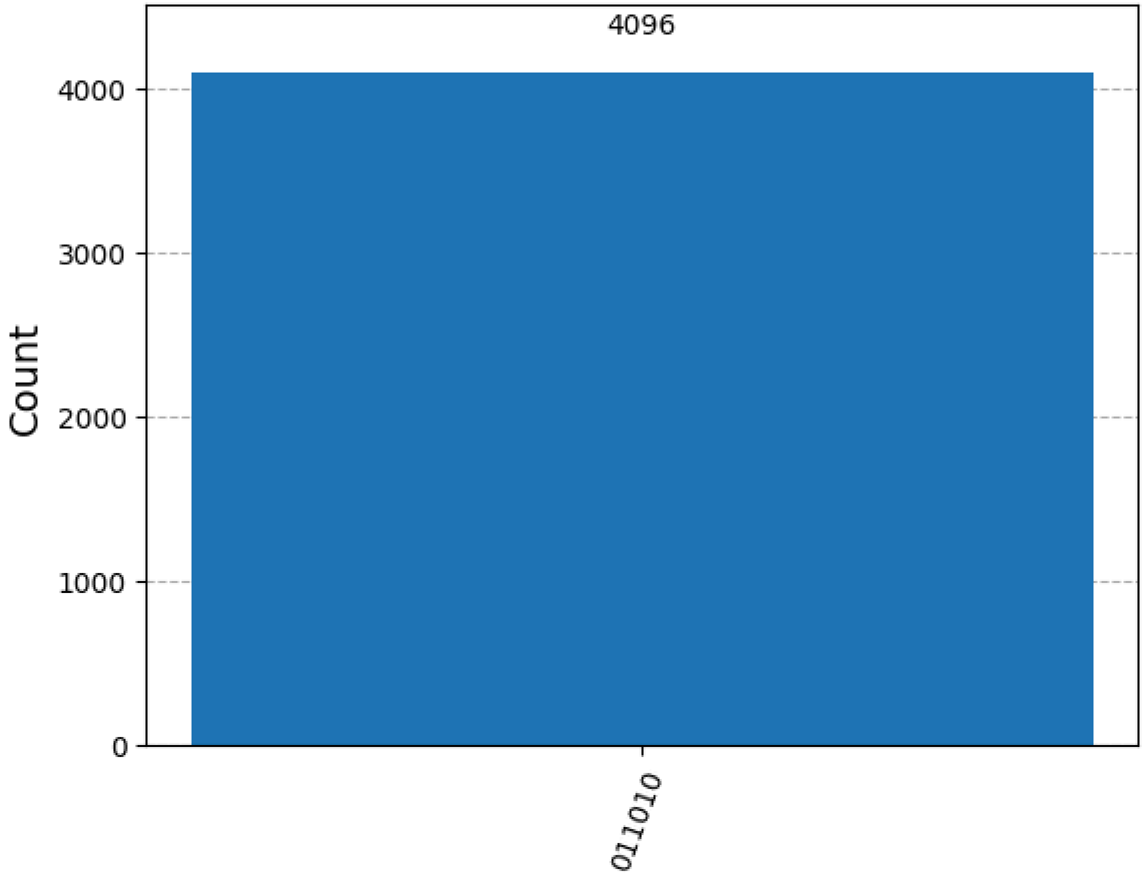


Fig 02 : Measured state Vs Counts

The histogram represents the probability of measuring the output states when my quantum circuit runs on the simulator.

In the above histogram :

- The **x-axis** represents the measured states.
- The **y-axis** represents the number of times each state was measured.

I ran my circuit for 4096 trials.

Here, on the x-axis, there is 011010 with count 4096, which means that the measurement found the qubits q5, q4, q3, q2, q1, q0 in 0, 1, 1, 0, 1, 0 state with 100% probability.

So, I get back my secret string s = 011010 with 100% probability after the measurement.

3. IBM QC Hardware calculation

i) Imports :

In [9]:
```
# Importing the QiskitRuntimeService class from qiskit_ibm_runtime module
# The QiskitRuntimeService class is used to connect to IBMQ Services
# and run actual IBM QC hardware
from qiskit_ibm_runtime import QiskitRuntimeService

# Importing the SamplerV2 class from qiskit_ibm_runtime module
# The SamplerV2 class is used to find the probabilities of output states
from qiskit_ibm_runtime import SamplerV2 as Sampler
```

ii) Getting the Hardware and running it

In [10]:
```
# I am creating a new object of QiskitRuntimeService
# It is used to connect with my IBMQ account and use the sevices
service = QiskitRuntimeService()



# backends method is used to fetch list of all available quantum backends
mybackends = service.backends(operational = True, simulator = False,
                              min_num_qubits = 5)
mybackends
```

Out[10]:
```
[<IBMBackend('ibm_brisbane')>,
 <IBMBackend('ibm_sherbrooke')>,
 <IBMBackend('ibm_kyiv')>]
```

In [11]:
```
# least_busy method is used to pick the best available backend
device = service.least_busy(operational = True, simulator = False,
                            min_num_qubits = 5)
device
```

Out[11]:
```
<IBMBackend('ibm_brisbane')>
```

In [12]:
```
# Transpile transforms the circuit to something appropriate for the hardware
# seed is used to get the same transpiled circuit every time I run
transpiled_circuit = transpile(circuit, device, seed_transpiler = 13)

# SamplerV2 is used to find the probabilities of output states
# mode = device is used to select the least busy hardware I got above
sampler = Sampler(mode = device)

# The run method in the sampler executes the transpiled circuit
job_hardware = sampler.run([transpiled_circuit])
```

iii) Fetching the result and plotting histogram :

In [13]:
```
# I am fetching the results of the sampler job execution.
# This result contains the counts of each measurement outcome.
```

```
result_hardware = job_hardware.result()

# the 1st element at 0th index is the public result
pub_result = result_hardware[0]

# I am extracting the classical data part from the public result
# the values of c tells about the count of each outcome
classical_data = pub_result.data.c

# .get_counts() is used to measure the data in the classical bit 'c'
# I am generating and displaying a histogram of the execution outcomes
plot_histogram(classical_data.get_counts())
```

Out[13]:

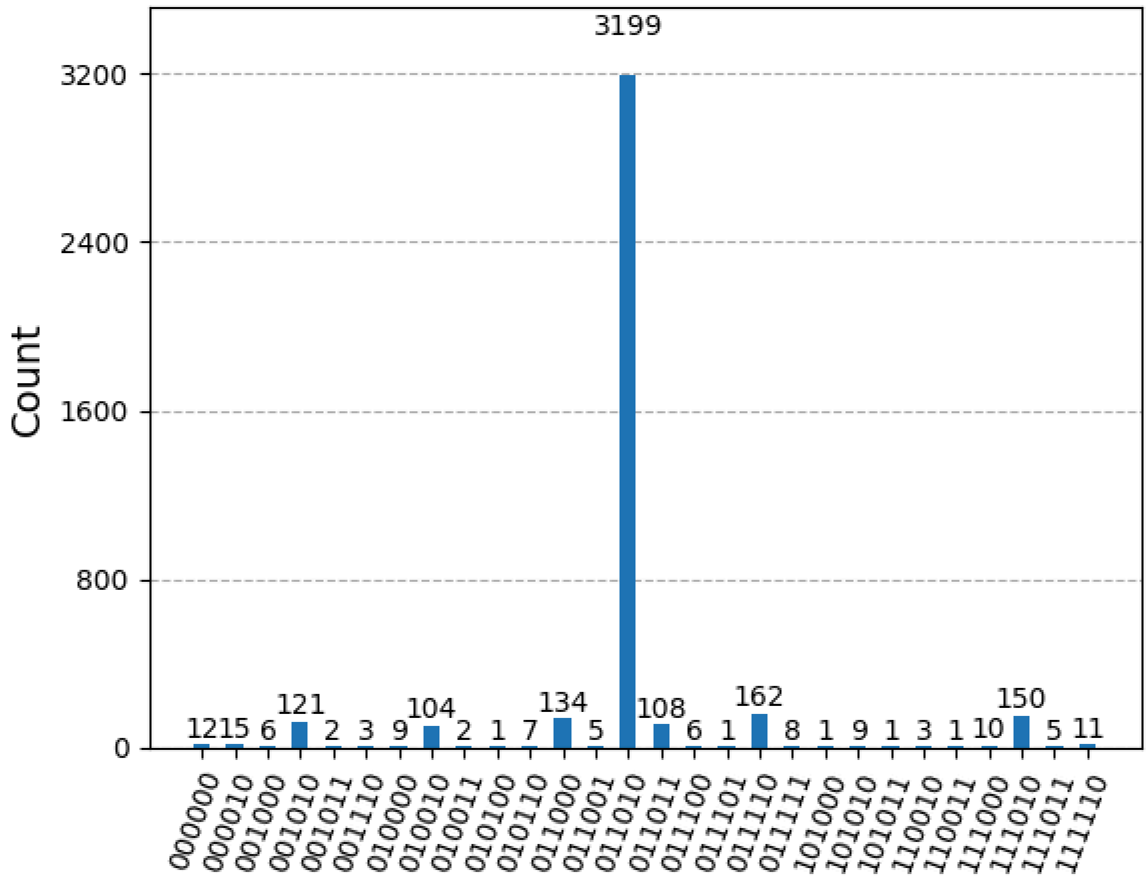


Fig 03 : Measured state Vs Counts

The histogram represents the probability of measuring the output states when my quantum circuit runs on the IBM QC Hardware.

In the above histogram :

- The **x-axis** represents the measured states.
- The **y-axis** represents the number of times each state was measured.

The circuit ran for 4096 trials on the IBM QC Hardware.

Here, on the x-axis, there are multiple different measured states with smaller counts and 011010 with count 3199.

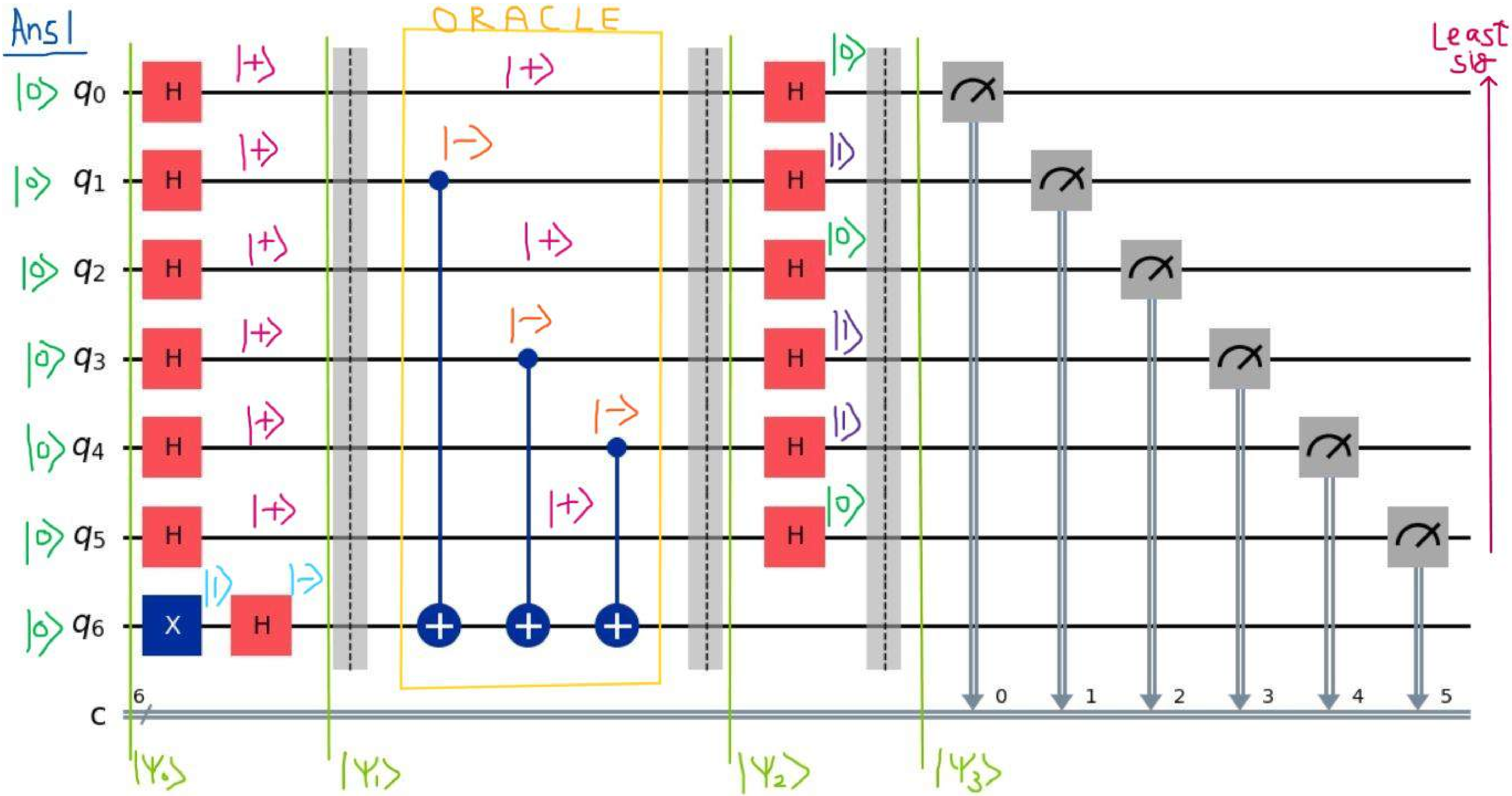
3199/4096 = 0.7810

This means that the measurement found the qubits q5, q4, q3, q2, q1, q0 in 0, 1, 1, 0, 1, 0 state with 78.1 % probability.

So, I get back my secret string s = 011010 with 78.1 % probability after the measurement. This proves that I successfully got back my secret string with a high probability.

Also the simulator and the IBM QC Hardware produced very similar results, which validates that the Bernstein-Vazrani Algorithm worked fine.

Ans 1



ORACLE
Least sig
|0⟩ q0
|0⟩ q1
|0⟩ q2
|0⟩ q3
|0⟩ q4
|0⟩ q5
|0⟩ q6
c
6
H
X
|+⟩
|−⟩
|1⟩
|0⟩
0
1
2
3
4
5
|ψ0⟩
|ψ1⟩
|ψ2⟩
|ψ3⟩

$n = 6$

x-register with $n=6$ qubits and a y-register with 1 qubit.

i) The initial state of all qubits is $|0\rangle \rightarrow$

so, $|\psi_0\rangle = |0\rangle^{\otimes n} |0\rangle_y$

$n=6$, $\boxed{|\psi_0\rangle = |0\rangle^{\otimes 6} |0\rangle = |000000\rangle \otimes |0\rangle}$

ii) Converting the y-register to $|-\rangle$ state $\rightarrow$

$X|0\rangle = |1\rangle$

$H|1\rangle = |-\rangle$

so, $|\psi_0'\rangle = |0\rangle^{\otimes n} |-\rangle_y$

$n=6$, $\boxed{|\psi_0'\rangle = |0\rangle^{\otimes 6} |-\rangle = |000000\rangle \otimes |-\rangle}$

iii) Applying H to all x-registers $\rightarrow$

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

For all n, $H^{\otimes n} |0\rangle^{\otimes n} = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n - 1} |x\rangle$

$$\therefore |\psi_1\rangle = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n - 1} |x\rangle |-\rangle_y$$

$n=6$

$$H^{\otimes 6} |000000\rangle = \frac{1}{\sqrt{2^6}} \sum_{x=0}^{2^6 - 1} |x\rangle$$

$n=6$.

$$\boxed{|\psi_1\rangle = \frac{1}{\sqrt{64}} \sum_{x=0}^{63} |x\rangle \otimes |-\rangle}$$

iv) Applying oracle -

$$U_f |x\rangle |-\rangle_y = (-1)^{f(x)} |x\rangle |-\rangle_y$$

$$f(x) = s \cdot x$$

$$\therefore |\psi_2\rangle = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} (-1)^{s \cdot x} |x\rangle |-\rangle_y$$

$n = 6$,
$$\boxed{|\psi_2\rangle = \frac{1}{\sqrt{64}} \sum_{x=0}^{63} (-1)^{s \cdot x} |x\rangle \otimes |-\rangle}$$

v) Applying H again on n-registers →

$$H^{\otimes n} (-1)^{s \cdot x} |x\rangle = |s\rangle$$

$$\therefore |\psi_3\rangle = |s\rangle |-\rangle_y$$

$n = 6$,
$$H^{\otimes 6} \sum_{x=0}^{63} (-1)^{s \cdot x} |x\rangle = \frac{1}{\sqrt{64}} \sum_{z=0}^{63} \left(\frac{1}{\sqrt{64}} \sum_{x=0}^{63} (-1)^{s \cdot x} (-1)^{x \cdot z} \right) |z\rangle$$

$$= \underbrace{\frac{1}{64} \sum_{x=0}^{63} (-1)^{x \cdot (s \oplus z)} |z\rangle} = |s\rangle$$

when, $z = s$, $s \oplus z = 0$, summation $= 64$

$z \neq s$, $s \oplus z \neq 0$, summation $= 0$

so, $\boxed{|\psi_3\rangle = |s\rangle |-\rangle}$

4) Given x- summation $\sum_{x=0}^{2^n-1} (-1)^{f(x)+x\cdot z} = K$.

To prove it yields a non-zero value only for $|z\rangle = |s\rangle$ and zero for all other $|z\rangle \neq |s\rangle$.

$$f(x) = s\cdot x \quad \text{and} \quad s\cdot x = \sum_{i=0}^{n-1} s_i x_i \bmod 2$$

$$\therefore f(x) = s_0 x_0 + s_1 x_1 + \text{----} + s_{n-1} x_{n-1} \bmod 2$$

$\therefore$ x-summation $K = \sum_{x=0}^{2^n-1} (-1)^{s\cdot x + x\cdot z}$

$$= \sum_{x=0}^{2^n-1} (-1)^{x\cdot(s\oplus z)}$$

Doing binary summation $\rightarrow$ $x = \underbrace{(x_{n-1}, x_{n-2}, \text{---} x_0)}_{n \text{ bit binary string}}$

$K =$

$$\sum_{x=0}^{2^n-1} = \sum_{x_0\in\{0,1\}} \sum_{x_1\in\{0,1\}} \text{---} \sum_{x_{n-1}\in\{0,1\}}$$

$$\therefore (-1)^{x\cdot(s\oplus z)} = (-1)^{\sum_{i=0}^{n-1} x_i(s_i\oplus z_i)}$$

$$= \prod_{i=0}^{n-1} (-1)^{x_i(s_i\oplus z_i)}$$

$$\therefore K = \sum_{x_0\in\{0,1\}} \sum_{x_1\in\{0,1\}} \text{---} \sum_{x_n\in\{0,1\}} \prod_{i=0}^{n-1} (-1)^{x_i(s_i\oplus z_i)}$$

$$\therefore K = \prod_{i=0}^{n-1} \sum_{x_i \in \{0,1\}} (-1)^{x_i (s_i \oplus z_i)}$$

Binary, so only 2 possibilities

If $s_i \oplus z_i = 0$, $\sum_{x_i \in \{0,1\}} (-1)^{x_i \cdot 0} = 2$.

If $s_i \oplus z_i = 1$, $\sum_{x_i \in \{0,1\}} (-1)^{x_i \cdot 1} = (-1)^0 + (-1)^1 = 0$.

$$\therefore K = \prod_{i=0}^{n-1} \sum_{x_i \in \{0,1\}} (-1)^{x_i (s_i \oplus z_i)}$$

$$K = \prod_{i=0}^{n-1} 2 = 2^n \qquad\qquad K = \prod_{i=0}^{n-1} 0 = 0$$

$\therefore$ For $|z\rangle = |s\rangle \longrightarrow$ "x-summation" $= 2^n$

For $|z\rangle \neq |s\rangle \longrightarrow$ "x-summation" $= 0$

For My $|s\rangle = |011010\rangle$,

$\longrightarrow$ if $|z\rangle = |011010\rangle \Rightarrow \dfrac{x\text{-summation}}{(2)^6 = 64}$

$\longrightarrow$ if $|z\rangle \neq |011010\rangle \Rightarrow \dfrac{x\text{-summation}}{0}$

For example, $s = |011010\rangle$, Let $z = |000000\rangle$

$$s \oplus z = 011010 \oplus 000000 = 011010$$

$$K = \prod_{i=0}^{5} \sum_{x_i \in \{0,1\}} (-1)^{x_i (s_i \oplus z_i)}$$

$$= 2 \times 0 \times 0 \times 2 \times 0 \times 2 = 0.$$

$i=0$, $s \oplus z = 0 \Rightarrow \sum = 2$
$i=1$, $s \oplus z = 1 \Rightarrow 0$
$i=2$, $s \oplus z = 1 \Rightarrow 0$
$i=3$, $s \oplus z = 0 \Rightarrow 2$
$i=4$, $s \oplus z = 1 \Rightarrow 0$
$i=5$, $s \oplus z = 0 \Rightarrow 2$

For $|z\rangle \neq |s\rangle$, $K = 0$

Ans. 5).

a) Show $CNOT\, |x\rangle |-\rangle = (-1)^x |x\rangle |-\rangle$

Target

control

$x=0$, $|x\rangle |-\rangle = |0\rangle \dfrac{|0\rangle - |1\rangle}{\sqrt{2}}$

control Target

$CNOT\, |0\rangle |-\rangle = |0\rangle |-\rangle = (-1)^0 |0\rangle |-\rangle$ ---- (i)

control is $|0\rangle$, so target will not flip.

$x=1$, $|x\rangle |-\rangle = |1\rangle \dfrac{|0\rangle - |1\rangle}{\sqrt{2}}$

control Target

$CNOT\, |1\rangle |-\rangle$

control is $|1\rangle$, so target will flip

$\therefore CNOT\, |1\rangle |-\rangle = |1\rangle \dfrac{-|0\rangle + |1\rangle}{\sqrt{2}} = -|1\rangle |-\rangle$

$\therefore CNOT\, |1\rangle |-\rangle = (-1)^1 |1\rangle |-\rangle$ ---- (ii)

From, (i) and (ii), it is proved that

$$CNOT\, |x\rangle |-\rangle = (-1)^x |x\rangle |-\rangle$$

b). CNOT $|+\rangle|-\rangle$

$$|+\rangle|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

or $$|+\rangle|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes |-\rangle$$

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

Applying CNOT -

$$\text{CNOT } |+\rangle|-\rangle = \frac{1}{\sqrt{2}}(\text{CNOT } |0\rangle|-\rangle + \text{CNOT } |1\rangle|-\rangle)$$

We know,
$$\text{CNOT } |x\rangle|-\rangle = (-1)^x |x\rangle|-\rangle.$$

$$\therefore \text{CNOT } |0\rangle|-\rangle = (-1)^0 |0\rangle|-\rangle = |0\rangle|-\rangle$$

$$\therefore \text{CNOT } |1\rangle|-\rangle = (-1)^1 |1\rangle|-\rangle = -|1\rangle|-\rangle$$

$$\therefore \text{CNOT } |+\rangle|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle|-\rangle - |1\rangle|-\rangle)$$

$$= \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)|-\rangle$$

$$= |-\rangle|-\rangle$$

CNOT $|-\rangle|-\rangle$

$$|-\rangle|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)\otimes|-\rangle$$

$$= \frac{1}{\sqrt{2}}(|0\rangle|-\rangle - |1\rangle|-\rangle)$$

Applying CNOT $\rightarrow$

$$\text{CNOT}\,|-\rangle|-\rangle = \frac{1}{\sqrt{2}}(\text{CNOT}\,|0\rangle|-\rangle - \text{CNOT}\,|1\rangle|-\rangle)$$

$$\text{CNOT}\,|0\rangle|-\rangle = (-1)^0\,|0\rangle|-\rangle = |0\rangle|-\rangle$$

$$\text{CNOT}\,|1\rangle|-\rangle = (-1)^1\,|1\rangle|-\rangle = -|1\rangle|-\rangle$$

$$\text{CNOT}\,|-\rangle|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle|-\rangle - (-|1\rangle|-\rangle))$$

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

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

$$= |+\rangle|-\rangle$$

c) Show that $\Rightarrow$ $U_f |x\rangle|-\rangle = (-1)^{f(x)} |x\rangle|-\rangle$

$$U_f |x\rangle|-\rangle = U_f \left(|x\rangle \otimes \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)\right)$$

$$= \frac{1}{\sqrt{2}}\left(U_f |x\rangle|0\rangle - U_f |x\rangle|1\rangle \right)$$

$$U_f |x\rangle|0\rangle = |x\rangle|0 \oplus f(x)\rangle \qquad U_f |x\rangle|1\rangle = |x\rangle|1 \oplus f(x)\rangle$$

$$= |x\rangle|f(x)\rangle$$

$$\therefore U_f |x\rangle|-\rangle = \frac{1}{\sqrt{2}}\left(|x\rangle|f(x)\rangle - |x\rangle|1 \oplus f(x)\rangle \right)$$

$$= |x\rangle \otimes \frac{1}{\sqrt{2}}\left(|f(x)\rangle - |1 \oplus f(x)\rangle \right)$$

i) $f(x) = 0 \Rightarrow |f(x)\rangle - |1 \oplus f(x)\rangle = |0\rangle - |1\rangle \times \frac{\sqrt{2}}{\sqrt{2}}$

$$= \sqrt{2}\,|-\rangle$$

ii) $f(x) = 1 \Rightarrow |f(x)\rangle - |1 \oplus f(x)\rangle = |1\rangle - |0\rangle \times \frac{\sqrt{2}}{\sqrt{2}}$

$$= -\sqrt{2}\,|-\rangle$$

$$\therefore U_f |x\rangle|-\rangle = |x\rangle \otimes \frac{1}{\sqrt{2}}\left((-1)^{f(x)} \sqrt{2}\,|-\rangle \right)$$

$$= (-1)^{f(x)} \left(|x\rangle \otimes |-\rangle \right)$$

$$= (-1)^{f(x)} |x\rangle|-\rangle$$

d). $U_f |+\rangle|-\rangle$ for $f(0)=f(1)$

$$|+\rangle|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle+|1\rangle) \otimes \frac{1}{\sqrt{2}}(|0\rangle-|1\rangle)$$

$$= \frac{1}{2}(|0\rangle|0\rangle - |0\rangle|1\rangle + |1\rangle|0\rangle - |1\rangle|1\rangle)$$

$$\therefore U_f |+\rangle|-\rangle = \frac{1}{2}\left(U_f(|0\rangle|0\rangle - U_f|0\rangle|1\rangle + U_f|1\rangle|0\rangle - U_f|1\rangle|1\rangle\right)$$

Using formula, $U_f|x\rangle|y\rangle = |x\rangle|y \oplus f(x)\rangle$

$$\therefore U_f|0\rangle|0\rangle = |0\rangle|0\oplus f(0)\rangle = |0\rangle|f(0)\rangle$$

$$U_f|0\rangle|1\rangle = |0\rangle|1\oplus f(0)\rangle$$

$$U_f|1\rangle|0\rangle = |1\rangle|0\oplus f(1)\rangle = |1\rangle|f(1)\rangle$$

$$U_f|1\rangle|1\rangle = |1\rangle|1\oplus f(1)\rangle$$

$$\therefore U_f|+\rangle|-\rangle = \frac{1}{2}\left(|0\rangle|f(0)\rangle - |0\rangle|1\oplus f(0)\rangle + |1\rangle|f(1)\rangle - |1\rangle|1\oplus f(1)\rangle\right)$$

Case 1 $\rightarrow$ i) $f(0) = f(1) = 0$.

$$\therefore U_f|+\rangle|-\rangle = \frac{1}{2}(|0\rangle|0\rangle - |0\rangle|1\rangle + |1\rangle|0\rangle - |1\rangle|1\rangle)$$

$$= \frac{1}{2}(|0\rangle(|0\rangle-|1\rangle) + |1\rangle(|0\rangle-|1\rangle))$$

$$= \frac{1}{2}(|0\rangle+|1\rangle)(|0\rangle-|1\rangle)$$

$$= |+\rangle|-\rangle$$

ii) $f(0) = f(1) = 1$

$$U_f |+\rangle|-\rangle = \frac{1}{2}\left(|0\rangle|1\rangle - |0\rangle|0\rangle + |1\rangle|1\rangle - |1\rangle|0\rangle\right)$$

$$= \frac{1}{2}\left(|0\rangle(|1\rangle - |0\rangle) + |1\rangle(|1\rangle - |0\rangle)\right)$$

$$= \frac{1}{2}\left(|0\rangle + |1\rangle\right)\left(|1\rangle - |0\rangle\right)$$

$$= -\frac{1}{2}\left(|0\rangle + |1\rangle\right)\left(|0\rangle - |1\rangle\right)$$

$$= -|+\rangle|-\rangle$$

For $f(0) = f(1)$,

$$U_f |+\rangle|-\rangle = (-1)^{f(0)} |+\rangle|-\rangle$$

Case-2 → $f(0) \neq f(1)$. Let $f(0) = 0$ and $f(1) = 1$
or $f(0) = 1$ and $f(1) = 0$

$\therefore$ $$U_f |+\rangle|-\rangle = \frac{1}{2}\left(|0\rangle|0\rangle - |0\rangle|1\rangle + |1\rangle|1\rangle - |1\rangle|0\rangle\right)$$

$$= \frac{1}{2}\left(|0\rangle(|0\rangle - |1\rangle) + |1\rangle(|1\rangle - |0\rangle)\right)$$

$$= \frac{1}{2}\left(|0\rangle(|0\rangle - |1\rangle) - |1\rangle(|0\rangle - |1\rangle)\right)$$

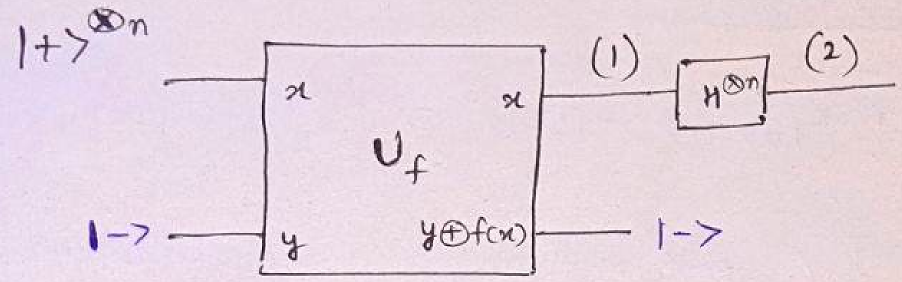
$$= \frac{1}{2}\left(|0\rangle - |1\rangle\right)\left(|0\rangle - |1\rangle\right)$$

$$= |-\rangle|-\rangle$$

$\therefore$ for $f(0) \neq f(1)$

$$U_f |+\rangle|-\rangle = |-\rangle|-\rangle$$

6)



Given, $$|+\rangle^{\otimes n} = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle$$

(a) State vector (1) $\rightarrow$

$$U_f\left(|+\rangle^{\otimes n} |-\rangle \right) = U_f\left(\frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle \otimes |-\rangle \right)$$

$$= \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} U_f\left(|x\rangle |-\rangle \right)$$

Now, $$U_f\left(|x\rangle |-\rangle\right) = U_f\left[|x\rangle \otimes \frac{1}{\sqrt{2}} \left(|0\rangle - |1\rangle \right) \right]$$

$$= \frac{1}{\sqrt{2}} \left(U_f |x\rangle |0\rangle - U_f |x\rangle |1\rangle \right)$$

$$= \frac{1}{\sqrt{2}} \left(|x\rangle |0 \oplus f(x)\rangle - |x\rangle |1 \oplus f(x)\rangle \right)$$

$$= \frac{1}{\sqrt{2}} \left(|x\rangle \otimes \left(\underbrace{|f(x)\rangle - |1 \oplus f(x)\rangle}_{} \right) \right)$$

$$f(x) = 0 \longrightarrow |f(x)\rangle - |1 \oplus f(x)\rangle = \left(|0\rangle - |1\rangle \right) \times \frac{\sqrt{2}}{\sqrt{2}}$$
$$= \sqrt{2}\, |-\rangle$$

$$f(x) = 1 \longrightarrow |f(x)\rangle - |1 \oplus f(x)\rangle = |1\rangle - |0\rangle \times \frac{\sqrt{2}}{\sqrt{2}}$$
$$= -\sqrt{2}\, |-\rangle$$

$$\therefore\ U_f(|x\rangle|-\rangle) = |x\rangle \otimes \frac{1}{\sqrt{2}}\left((-1)^{f(x)}\sqrt{2}\,|-\rangle\right)$$

$$= (-1)^{f(x)}\left(|x\rangle \otimes |-\rangle\right)$$

$$= (-1)^{f(x)}\,|x\rangle|-\rangle$$

$$\therefore\ U_f\left(|+\rangle^{\otimes n}|-\rangle\right) = \frac{1}{\sqrt{2^n}}\sum_{x=0}^{2^n-1} U_f(|x\rangle|-\rangle)$$

$$= \underbrace{\frac{1}{\sqrt{2^n}}\sum_{x=0}^{2^n-1}(-1)^{f(x)}\,|x\rangle}_{\text{state vector (1) after } U_f}|-\rangle$$

(b) State vector (2) $\rightarrow$

We know,
$$H|x\rangle = \frac{\sum_z (-1)^{xz}|z\rangle}{\sqrt{2}}$$

For n qubits,
$$H^{\otimes n}|x_1\cdots x_n\rangle = \frac{\sum_{z_1\cdots z_n}(-1)^{x_1z_1+\cdots+x_nz_n}|z_1\cdots z_n\rangle}{\sqrt{2^n}}$$

$$= \frac{\sum_{z=0}^{2^n-1}(-1)^{xz}|z\rangle}{\sqrt{2^n}}$$

State vector (2) is obtained by applying $H^{\otimes n}$ to the state vector (1)

So, we get →

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

$$= \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} (-1)^{f(x)} H^{\otimes n} |x\rangle$$

$$= \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} (-1)^{f(x)} \left(\frac{1}{\sqrt{2^n}} \sum_{z=0}^{2^n-1} (-1)^{x \cdot z} |z\rangle\right)$$

$$= \frac{1}{\sqrt{2^n}} \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} \sum_{z=0}^{2^n-1} (-1)^{f(x)+x\cdot z} |z\rangle$$

Let $K = \sum_{x=0}^{2^n-1} (-1)^{f(x) + x\cdot z}$

i) If $f(x)$ is a constant function ie. $f(x)=0$ or $f(x)=1$ for all 2^n x-values.

$$K = (-1)^{f(x)} \underbrace{\sum_{x=0}^{2^n-1} (-1)^{x\cdot z}}$$

if $z = 0^n$, $x\cdot z = 0$ so, $\sum_{x=0}^{2^n-1} (-1)^{x\cdot z} = 2^n$

if $z = 1^n$, $x \cdot z = \sum_{i=1}^{n} x_i \cdot 1 = \sum_{i=1}^{n} x_i \bmod 2$

If the no. of 1's is even, $x \cdot z = 0$

If the no. of 1's is odd, $x \cdot z = 1$

so, for $z = 1^n$

$$\sum (-1)^{xz} = \underbrace{\sum \overset{(-1)^0}{\overbrace{1}}}_{\text{even}} + \underbrace{\sum \overset{(-1)^1}{\overbrace{-1}}}_{\text{odd}} = 0$$

so, $K = 0$ (for $z = 1^n$) $\longrightarrow$ zero

$\underbrace{K = (-1)^{f(x)} \cdot 2^n}$ (for $z = 0^n$) $\longrightarrow$ Non-zero

ii) If $f(x)$ is a balanced function, $f(x)=0$ for half of 2^n x values and $f(x) = 1$ for other half of 2^n x values.

$$K = (-1)^{f(x)} \sum_{x=0}^{2^n-1} (-1)^{xz}$$

$$= \sum_{f(x)=0} (-1)^{0+xz} + \sum_{f(x)=1} (-1)^{1+xz}$$

$$= \sum_{f(x)=0} (-1)^{xz} - \sum_{f(x)=1} (-1)^{xz}$$

$= 0$ (always)