MANIPAL INSTITUTE OF TECHNOLOGY
MANIPAL
A Constituent Institution of Manipal University

Project Report
Topic: Bernstein-Vazirani Algorithm

Subject: Quantum Computing CSE 5115

Submitted By :
Uday Yadav : 230913008
Blen Joswin : 230913009
Poorva TM : 230913011

Under the guidance of :
Prof. Dr. Vivekananda Bhat K.
Department of Computer Science and Engineering

Bernstein-Vazirani Algorithm

1. Introduction

The Bernstein-Vazirani algorithm is intended to quickly ascertain an unknown bit string that is concealed in an oracle. It is a prime contender for real-world quantum computing applications due to its comparatively straightforward circuit design and mathematical structure. The hidden shift problem, which has significant uses in error-correcting codes and encryption, is resolved by the BV algorithm. Given a function f(x) that is guaranteed to have a hidden shift a, or that there exists an unknown bit string a such that $f(x) = f(a \oplus x)$ for all inputs x, we have a hidden shift issue. Finding the hidden shift an is the aim of the hidden shift issue.

2. Methodology

Imagine a concealed Boolean function that accepts an n-bit string $x= \{x_0, x_1, \ldots x_{n-1}\}$. It returns 1 for only a unique n-bit string $s= \{s_0, s_1, \ldots s_{n-1}\}$ and 0 for all other inputs.

To discover the secret number s, one might initially consider trying all possible numbers from 0 to 2^n-1 for an n-bit secret number. However, this approach results in an exponential number of attempts as n increases.

A more efficient strategy involves not just obtaining a yes/no result if the number matches, but instead calculating s.x modulo 2. This calculation involves computing the bitwise AND between the numbers s and x, summing the results, and finally returning the sum modulo 2. By providing the function with n different inputs (2^0, 2^1, 2^2, …, 2^{n-1}), each bit of the secret number can be revealed. This method requires only n-attempts to find the secret number.

The Bernstein-Vazirani algorithm offers an even more efficient solution, enabling the discovery of the secret number in just a single attempt, regardless of the size of the secret number.

3. Algorithm

The algorithm consists of four primary steps:

1. Set the initial state of the first n qubits to |0>, and the final qubit to the |1> state.
2. Apply Hadamard gates to all qubits.
3. Construct the “oracle”, which is a box containing the secret number. This is done by creating a function that applies CNOT gates from the first n qubits to the last qubit whenever there is a 1 in the secret number. This is performed in reverse order, meaning a CNOT gate will be applied from the nth qubit to the last qubit if the first bit of the secret number is 1.
4. Measure the first n qubits in the Bell basis, which involves applying Hadamard gates to the first n qubits again before taking measurements.

Quantum Circuit is specifically constructed for each unique secret string.

4. Implementation

```
from qiskit import *
s = '110101'
```

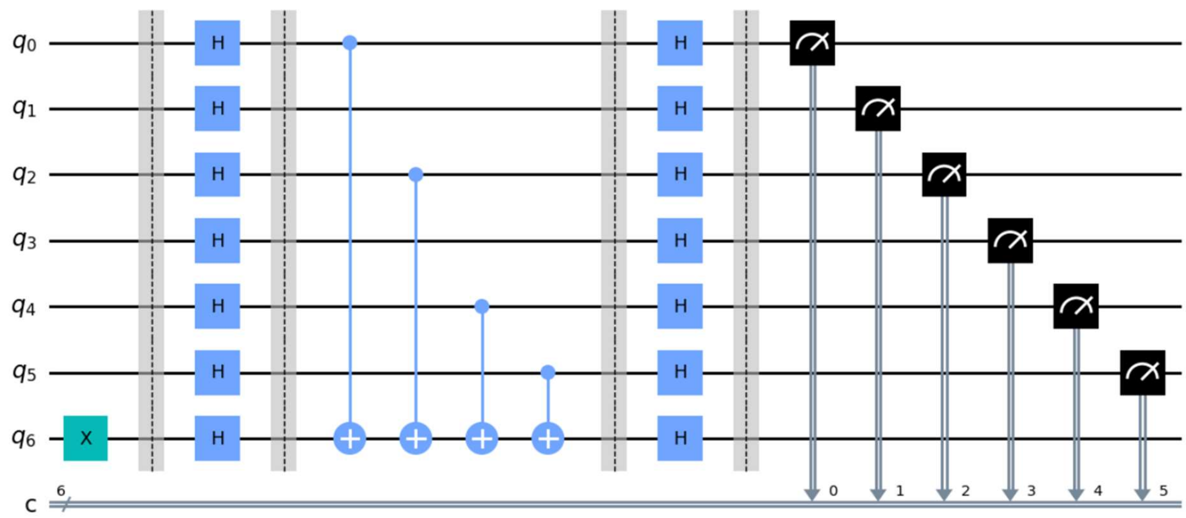
```
n = len(s)
circuit = QuantumCircuit(n+1,n)
```

```
circuit.x(n)
circuit.barrier()
circuit.h(range(n+1))
circuit.barrier()
```

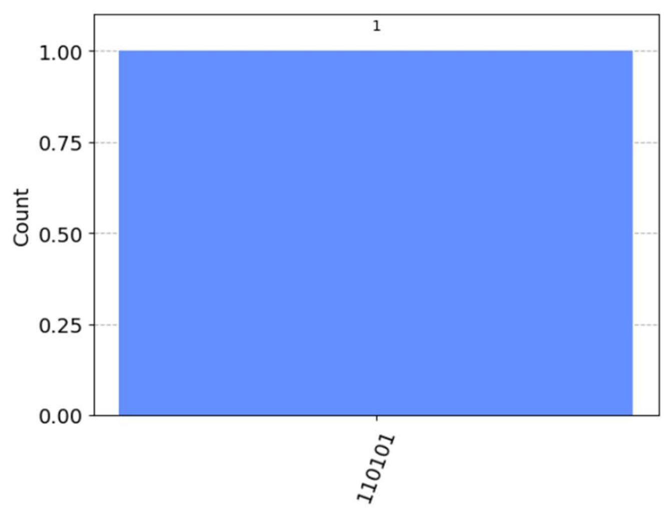
```
for ii, yesno in enumerate(reversed(s)):
    if yesno == '1':
        circuit.cx(ii, n)
```

```
circuit.barrier()
circuit.h(range(n+1))
circuit.barrier()
```

```
circuit.measure(range(n), range(n))
```

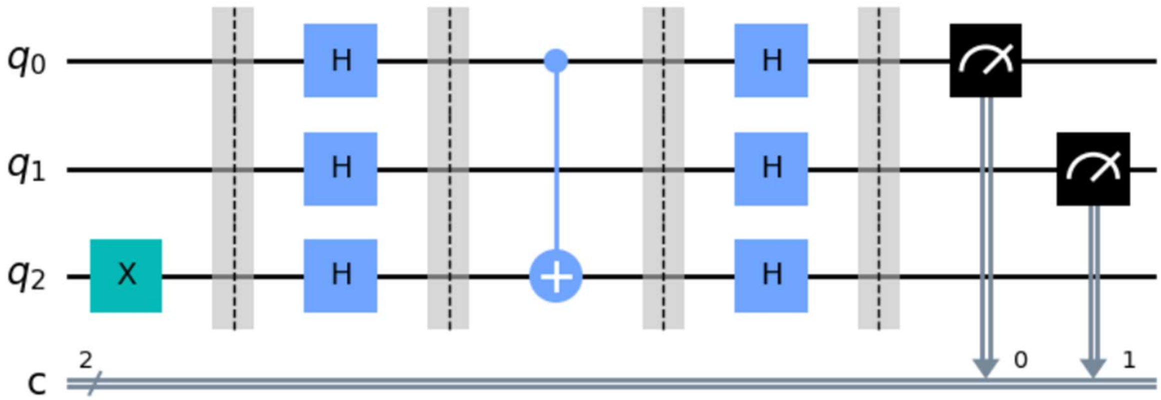



```
simulator = Aer.get_backend('qasm_simulator')
result = execute(circuit, backend=simulator, shots=1).result()
from qiskit.visualization import plot_histogram
plot_histogram(result.get_counts(circuit))
```



5. Applying Bernstein-Vazirani Algorithm

For s = “01”, we the following circuit is generated



We start with $|x_0\rangle = |00\rangle$, we are ignoring the ancillary qubit since its result does not matter, then applying H gate to qubits

$$|\psi_1\rangle = (\frac{|0\rangle + |1\rangle}{\sqrt{2}})(\frac{|0\rangle + |1\rangle}{\sqrt{2}}) \;= \frac{|00\rangle + |01\rangle + |10\rangle + |11\rangle}{2}.$$

Using phase kickback logic that we did in the Deutsch-Josza algorithm where we considered the cases for when the function gave the outputs 0 and 1 to define the action of the oracle as:

$$O_f|x\rangle = (-1)^{a\cdot x}|x\rangle,$$

Using the above equation, we have:

$$\begin{aligned}|\psi_2\rangle &= \frac{1}{2}((-1)^{00.01}\,|00\rangle + (-1)^{01.01}\,|01\rangle + (-1)^{10.01}\,|10\rangle + (-1)^{11.01}\,|11\rangle)\\ &= \frac{1}{2}(|00\rangle - |01\rangle + |10\rangle - |11\rangle).\end{aligned}$$

Now we need to apply the Hadamard gate to both bits,

$$\begin{aligned}|\psi_3\rangle &= (\frac{|0\rangle + |1\rangle}{\sqrt{2}})(\frac{|0\rangle + |1\rangle}{\sqrt{2}}) - (\frac{|0\rangle + |1\rangle}{\sqrt{2}})(\frac{|0\rangle - |1\rangle}{\sqrt{2}})\\ &+ (\frac{|0\rangle - |1\rangle}{\sqrt{2}})(\frac{|0\rangle + |1\rangle}{\sqrt{2}}) - (\frac{|0\rangle - |1\rangle}{\sqrt{2}})(\frac{|0\rangle - |1\rangle}{\sqrt{2}})\\ &= \frac{\begin{array}{c}(|00\rangle + |01\rangle + |10\rangle + |11\rangle - |00\rangle + |01\rangle - |10\rangle + |11\rangle\\ + |00\rangle + |01\rangle - |10\rangle - |11\rangle - |00\rangle + |01\rangle + |10\rangle - |11\rangle)\end{array}}{4}\\ &= |01\rangle\,,\end{aligned}$$

which is the query string and the expected output $|01\rangle$

6. Conclusion

In conclusion, the Bernstein-Vazirani algorithm stands out as a powerful and efficient quantum algorithm, showcasing the significant advantages quantum computing can offer in certain problem domains. By efficiently determining an unknown binary string in a single query to an oracle, the algorithm has demonstrated a marked improvement over its classical counterpart. Its ability to solve the specific problem of querying an oracle and unveiling a hidden binary string in O (1) time complexity has implications for cryptographic protocols and database search algorithms. As quantum computing continues to advance, the Bernstein-Vazirani algorithm exemplifies the transformative potential of quantum algorithms, providing a glimpse into the promising future of quantum information processing and its impact on various computational tasks.

Oracle Compression

- In classical computation, querying an oracle to determine the coefficients of a hidden linear function requires multiple queries. The Bernstein-Vazirani Algorithm, however, allows for the compression of this information into a single query, making it useful for oracles in various applications, particularly in cryptographic protocols.

Boolean Function Evaluation

- The algorithm can be used to evaluate hidden Boolean functions efficiently. This could be useful in scenarios where the Boolean function represents certain conditions or constraints in a problem, and determining the hidden input efficiently is essential.

Cryptography - Key Identification

- The Bernstein-Vazirani Algorithm is often mentioned in the context of cryptography. In a cryptographic setting, the hidden bit string could represent a secret key. Efficiently determining this key is crucial in scenarios such as secure communication or authentication protocols.

Quantum Key Distribution (QKD)

- Quantum Key Distribution protocols aim to establish secure communication channels using quantum properties. The Bernstein-Vazirani Algorithm could be employed in certain aspects of key establishment or verification.