USC University of Southern California

Solving the subset sum problem on a quantum computer

Sumeet Shirgure

Motivation

- Implementing two of the most popular and potentially useful algorithms on a quantum computer – the quantum Fourier transform and Grover's algorithm.
- Evaluating the performance of a general purpose NISQ computer on an NP hard problem in combinatorial optimization.

The subset sum problem

Input : given a list of positive integers $[x_1, x_2, \ldots, x_n]$ and a target sum t

Output : a subset (n-bit string $s_1 s_2 \ldots s_n$) with $\sum_i s_i x_i = t$
where one is promised to exist

Let $\sum_i x_i < 2^k$ for some k

In other words, find a bit string s with $f(s) = 1$

$$f(s) = \begin{cases} 1, \sum_i s_i x_i = t \\ 0, otherwise \end{cases}$$

The subset sum problem - classical methods

Known to be NP-complete.

The best known classical algorithms for this problem are superpolynomial in either n or k.

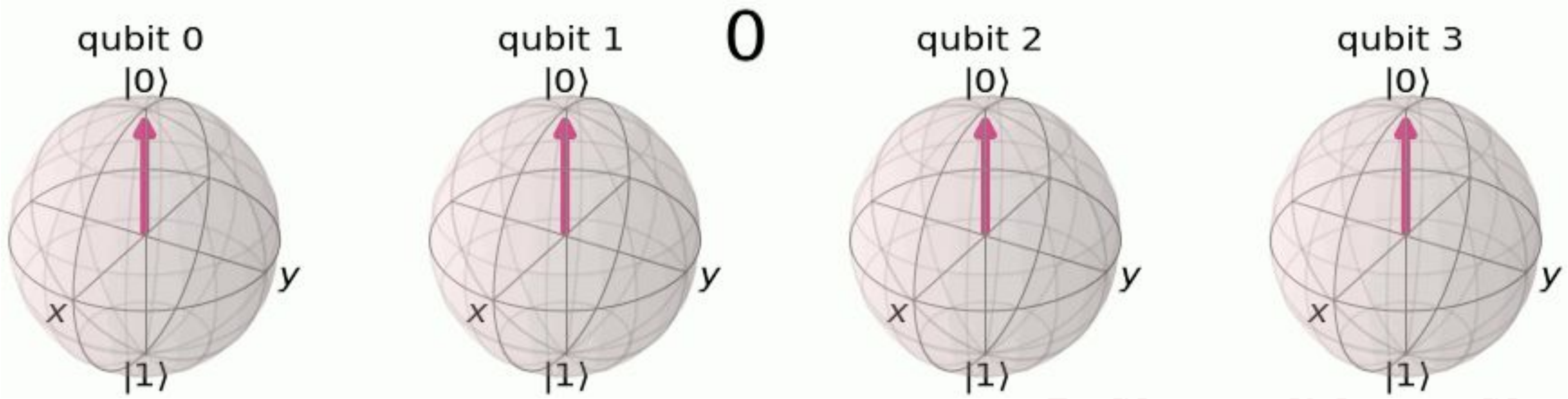
The (promise) subset sum problem - quantum algorithm

We'll discuss a quantum algorithm that runs in $O(nk\sqrt{2^n})$ time while only needing $O(n+k)$ qubits.

Method	Time	Memory*
Brute force	$O(2^n)$	$O(1)$
Dynamic programming	$O(nk2^k)$	$O(nk2^k)$
"Meet in the middle"	$O(\sqrt{2^n})$	$O(\sqrt{2^n})$
Grover search*	$O(nk\sqrt{2^n})$	$O(n+k)$

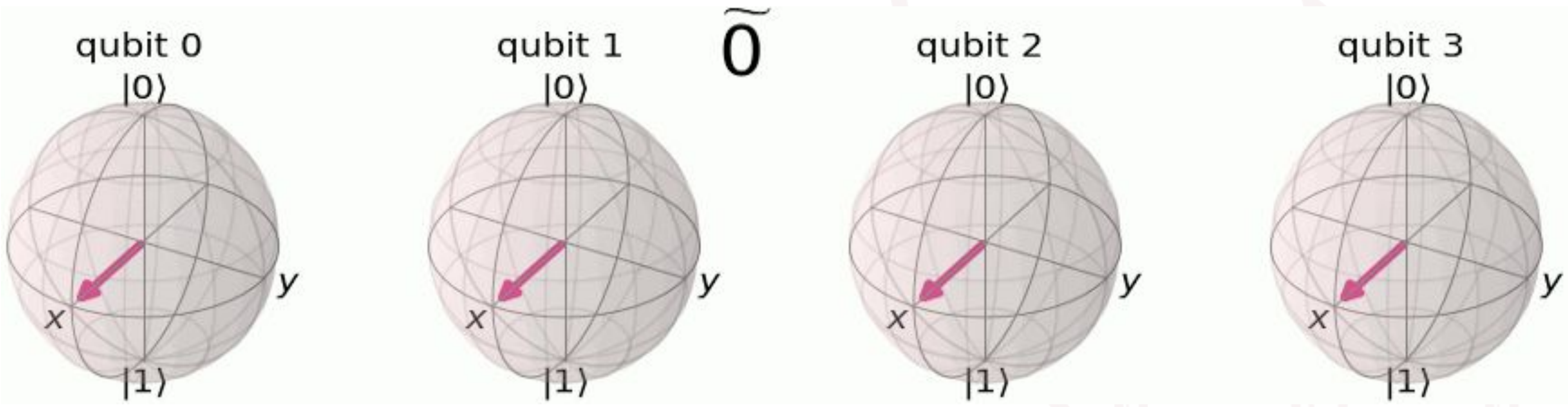
Fourier states

Here is the bloch sphere representation of the Z-basis encoding of 4-bit integers



Fourier states

Here is the bloch sphere representation of the F-basis encoding of 4-bit integers



$$|j_1, j_2, ..., j_n\rangle \rightarrow \frac{(|0\rangle + e^{2\pi i 0.j_n}|1\rangle)(|0\rangle + e^{2\pi i 0.j_{n-1}j_n}|1\rangle)...(|0\rangle + e^{2\pi i 0.j_1 j_2 ... j_n}|1\rangle)}{2^{n/2}}$$

USC

GIFs stolen from Qiskit's online manual

Fourier states

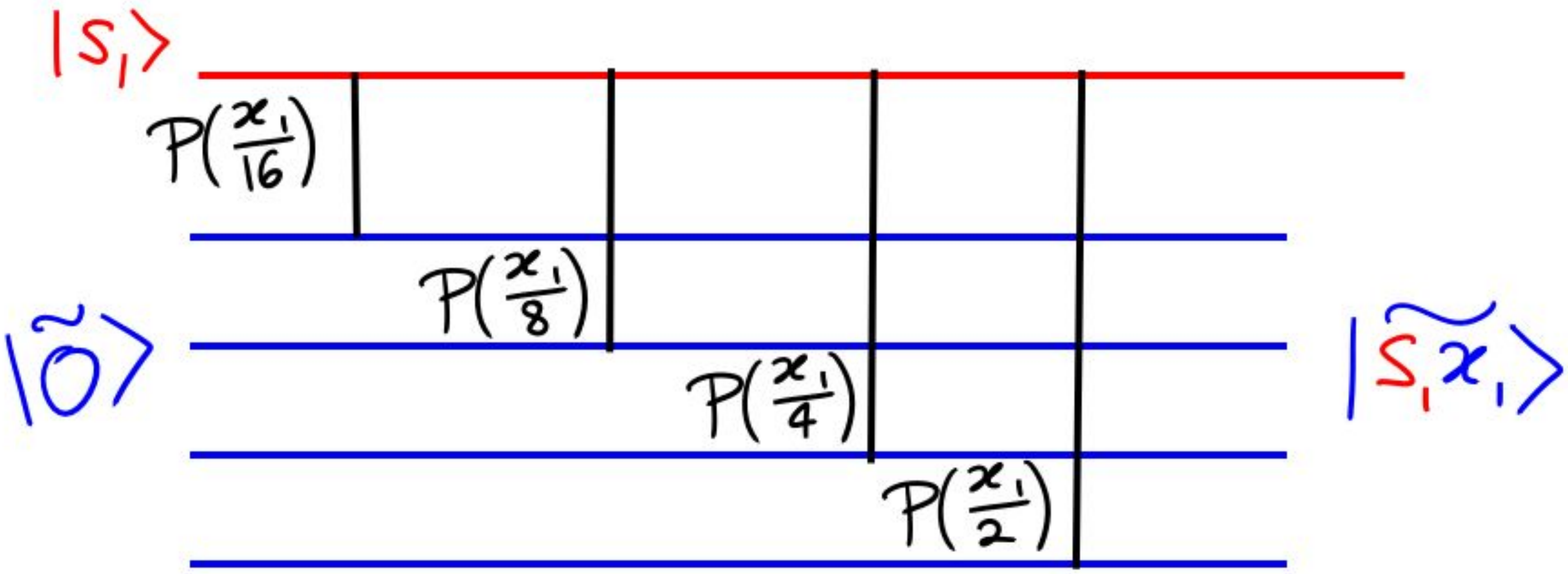
Addition in Z-basis is equivalent to rotation in F-basis.

$$|1\rangle \left(\frac{|0\rangle + e^{2\pi i \phi}|1\rangle}{\sqrt{2}} \right) \to |1\rangle \left(\frac{|0\rangle + e^{2\pi i (\phi+\theta)}|1\rangle}{\sqrt{2}} \right)$$

$$P(\theta) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & e^{2\pi i \theta} \end{pmatrix}$$

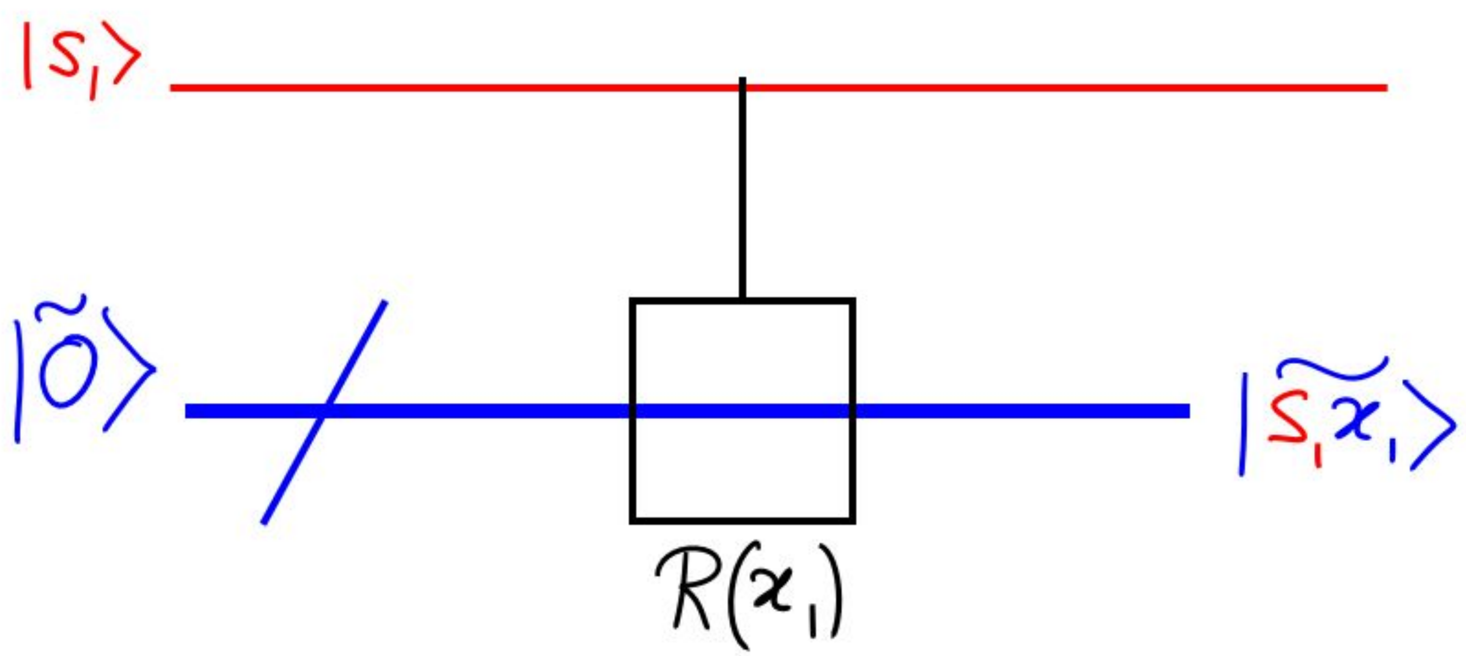
Adder circuit

Idea: Use these controlled rotations to implement addition.

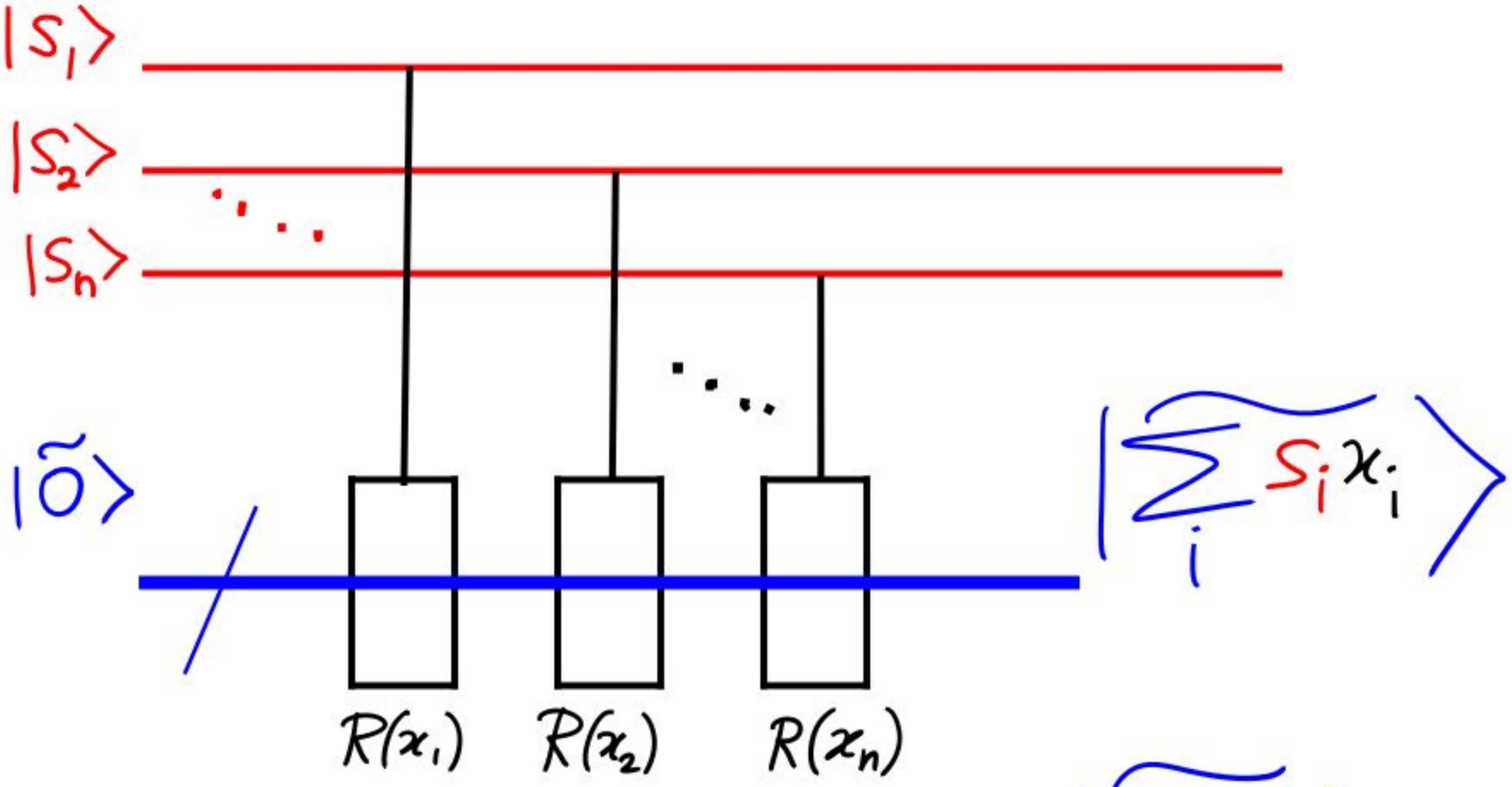


Adder circuit

Idea: Use these controlled rotations to implement addition.



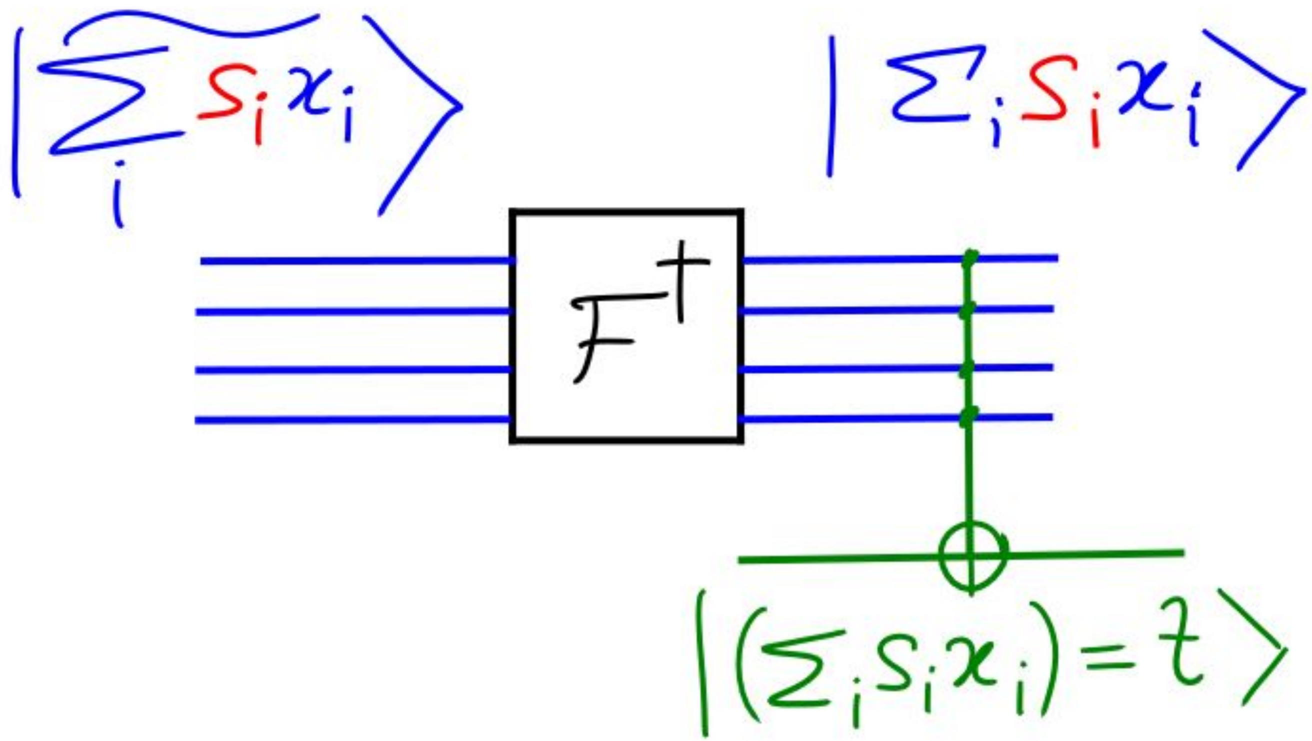
Adder circuit



$$\sum_\sigma |\sigma\rangle|\tilde{0}\rangle \longrightarrow \sum_\sigma |\sigma\rangle\left|\widetilde{\sum_i \sigma_i x_i}\right\rangle$$

Subset sum oracle

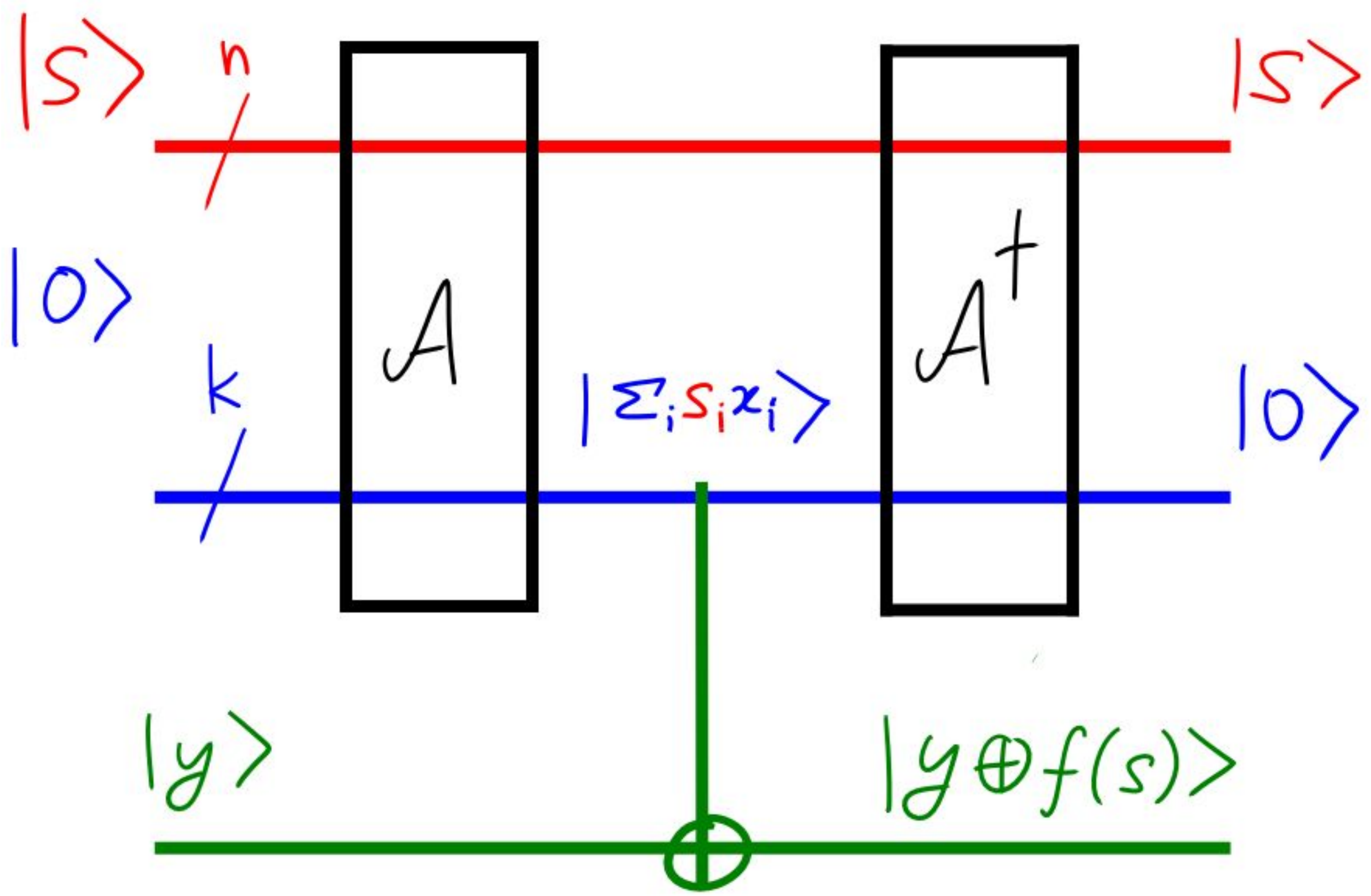
We can check if the subset sum equals the target in Z basis using a multi-controlled NOT gate :



Subset sum oracle

Combining all of the above, we have this circuit.

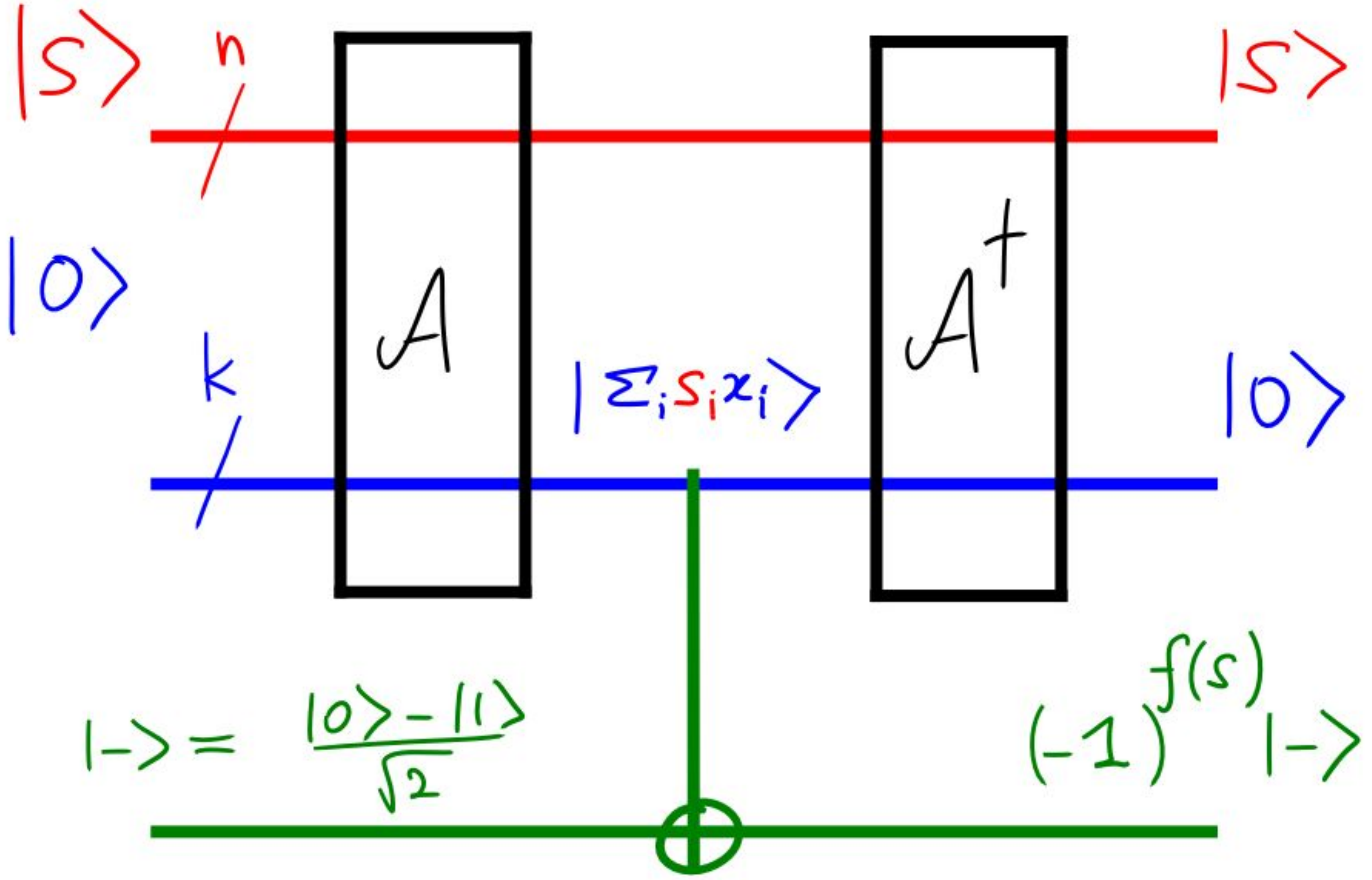
We also need to uncompute the blue register.



Subset sum oracle

The oracle to recognize a valid string can be modified to flip the global phase instead of a bit.

Now we can apply Grover's algorithm.



Executing a small instance on a trapped ion QC

x = [5, 7, 8, 9, 1],

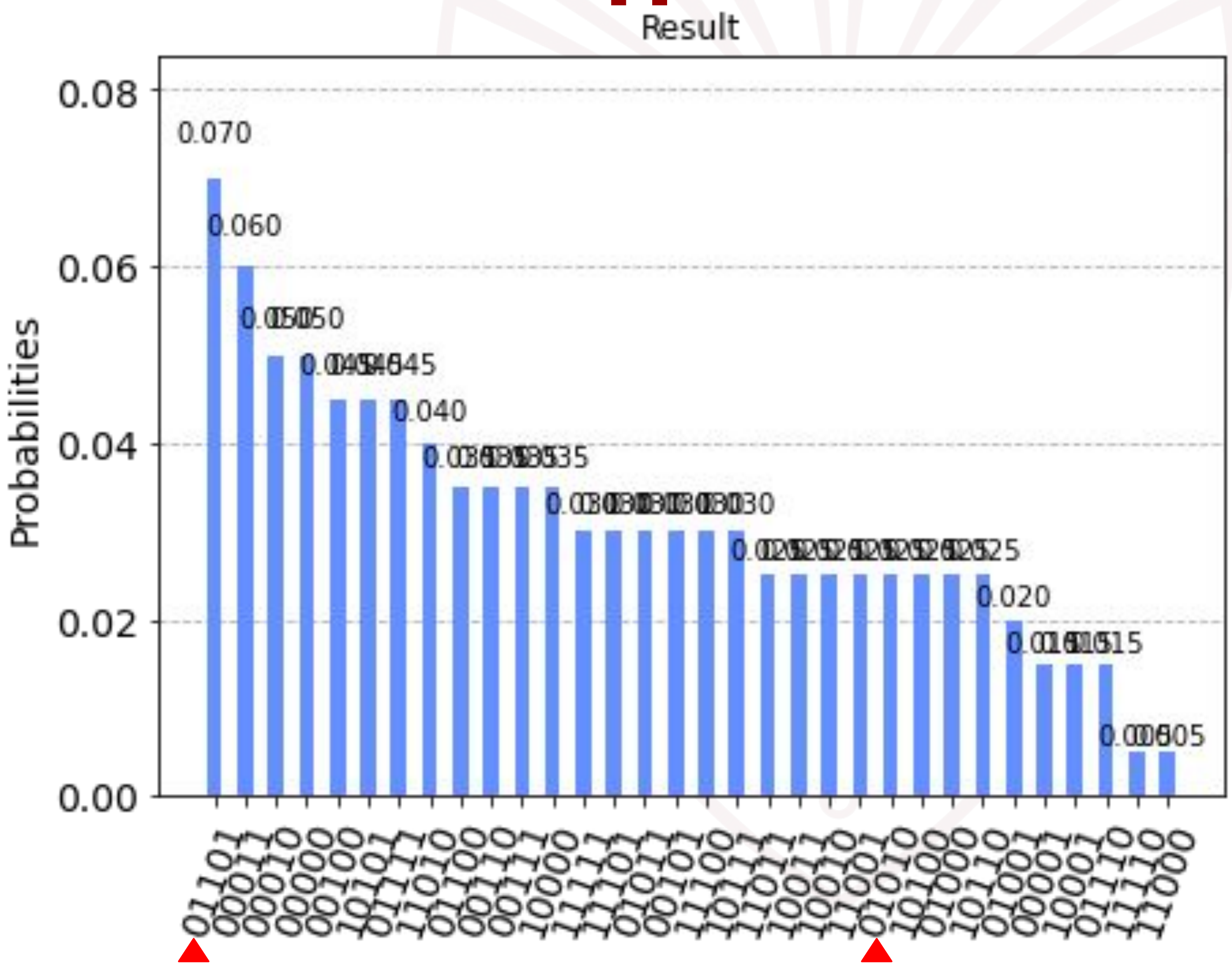
target = 16

feasible strings :

01101, 01010

n = 5, k = 5

After running 3 Grover iterations.



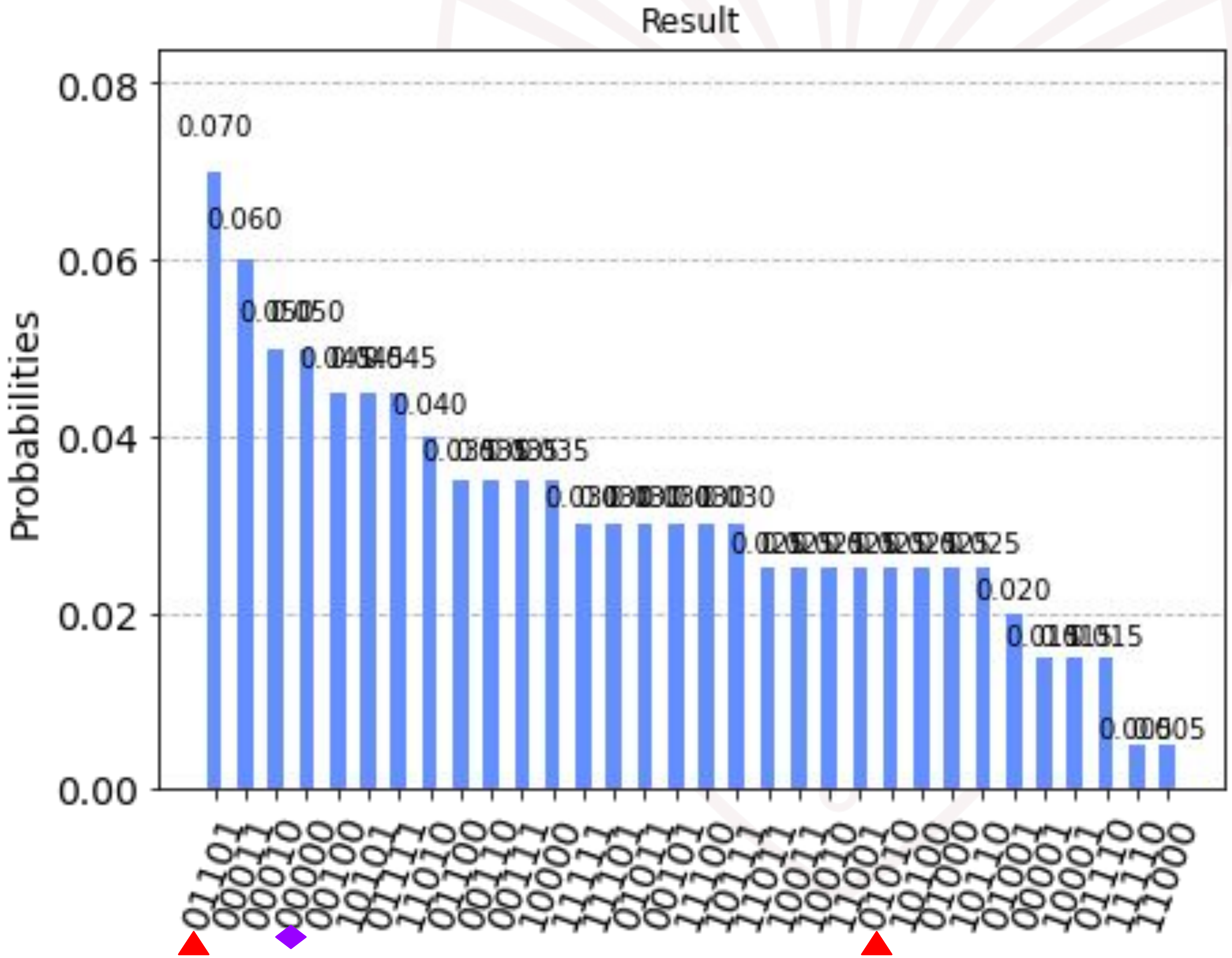
Executing a small instance on a trapped ion QC

Mode is a solution!

Pretty impressive for a circuit without error correction.

Difficult to determine why the second feasible set has low relative frequency.

Erasure errors?



USC

Choices

- Fourier - basis addition to reduce the number of gates. Multi-controlled not gates are costly, while controlled phase gates are much more efficient.

- IonQ’s trapped ion system allows two qubit gates between all pairs. We did need that.

Conclusion

Implemented a customized algorithm for solving an NP complete combinatorial optimization problem on a quantum computer.

Would love to try something similar with primitive error correction techniques, but the hardware necessary to do so is not available in the near future.

https://github.com/sumeetshirgure/qchack2022-microsoft-challenge

USC

Thanks for listening!

USC