

# iQuHack 2026: Title

Adam Godel<sup>1</sup>, Yebin Song<sup>1</sup>, Nico Jackson<sup>1</sup>, Travis Meyer<sup>1</sup>, and Timothy Wright<sup>1</sup>

<sup>1</sup>Boston University, Boston, Massachusetts 02215, USA

February 1, 2026

## Abstract

Placeholder

## 1 Introduction

The development and optimization of fault-tolerant quantum circuits has been at the forefront of quantum research in recent years. This weekend at iQuHack, Superquantum tasked us with implementing efficient approximate quantum circuit implementations of unitary gate operators using the Clifford +  $T$  gate set, where the latter is by far the most computationally difficult gate to execute and functions as a description of the “quantumness” of a circuit.<sup>1</sup>

Our approach to implement these unitaries varied widely depending on the operator. Some have symmetries that can be exploited, while others are completely random and require brute-force optimization. Furthermore, there is a subjective tradeoff between minimizing the number of  $T$  gates and minimizing the distance between the approximate and the exact unitaries. In this writeup, we present our approaches to implement all eleven operators tasked in our challenge, considering how our implementations can be varied in both of these optimization regimes.

We will begin by establishing some background on quantum computing and circuit theory, then continue by discussing each of our implementations in detail, including our experimental results, and conclude with

our reflection and possible future work on these problems.

## 2 Background

A *qubit* is the fundamental bit of information in quantum computing. A *two-qubit quantum state* is represented as

$$|\psi\rangle = \alpha|00\rangle + \beta|01\rangle + \gamma|10\rangle + \delta|11\rangle, \quad (1)$$

where  $|\alpha|^2 + |\beta|^2 + |\gamma|^2 + |\delta|^2 = 1$ . A two-qubit quantum state can be acted upon with *quantum gates*, which are represented as unitary matrices  $A \in U(4)$  such that  $|\psi\rangle \mapsto A|\psi\rangle$ , which is a valid two-qubit quantum state. Similarly, one-qubit quantum gates are unitary matrices in  $U(2)$ .

The following quantum gates are known as the *Clifford gates* [1]:

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad S = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix} \quad CX = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (2)$$

When combined with the  $T$  gate, defined as

$$T = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{4}} \end{bmatrix}, \quad (3)$$

they form a *universal gate set* [2], meaning that any unitary operation can be approximated to arbitrary

<sup>1</sup>Our code is freely available on GitHub at the following link: <https://github.com/adam-godel/2026-Superquantum>.

precision using circuits built with just these four gates.

Moreover, the *Gottesman-Knill theorem* [3] shows that the Clifford gates can be simulated in polynomial time on a classical computer. This reduces the “hardness” of a quantum circuit to be described in the number of  $T$  gates it contains.

### 3 Our Implementations

We now go through each of the eleven unitaries we implemented during the weekend, and the optimizations we made to leverage both low  $T$  gate counts and higher accuracy. We describe the implementations in order, although the difficulty in constructing each implementation varied widely.

#### 3.1 Controlled- $Y$ Gate

The *controlled- $Y$  gate* is defined as

$$CY = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -i \\ 0 & 0 & i & 0 \end{bmatrix}. \quad (4)$$

To implement this gate, we consider that  $(I \otimes S)CX(I \otimes S^\dagger) = CY$ , where the  $\dagger$  operator denotes the conjugate transpose. Since we can implement this gate using only Clifford gates, there is no tradeoff—we get high accuracy and no  $T$  gate cost.

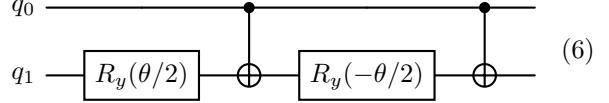
#### 3.2 Controlled- $R_y(\pi/7)$ Gate

The *controlled- $R_y$  gate* is defined as

$$CR_y(\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \cos(\theta/2) & -\sin(\theta/2) \\ 0 & 0 & \sin(\theta/2) & \cos(\theta/2) \end{bmatrix}. \quad (5)$$

We want to consider the case where  $\theta = \pi/7$ , which does not map cleanly onto a Clifford +  $T$  gate set. Therefore, we have to consider approximations. This is when we developed our central strategy for creating these implementations: **modularity** of gates.

Suppose we had a  $R_y$  gate in our toolbox. If this were the case, we could represent  $CR_y(\theta)$  with the following circuit:



We know that this is true by considering the state of  $q_1$  based on the state of  $q_0$ . We know that  $CR_y(\theta)$  should apply  $R_y(\theta)$  if  $q_0 = |1\rangle$  and do nothing if  $q_0 = |0\rangle$ . In Eq. 7, we can see that if  $q_0 = |0\rangle$ , nothing is done to  $q_1$ , while if  $q_0 = |1\rangle$ , we can see that a  $R_y(\theta)$  gate is applied to  $q_1$ .

The problem now reduces to an implementation of  $R_y(\theta)$ . We can observe that since  $SXS^\dagger = Y$  and  $HZH = X$ , we have  $R_y(\theta) = SHR_z(\theta)HS^\dagger$ , i.e. we can represent  $R_y(\theta)$  in terms of just  $R_z(\theta)$  and Clifford gates. It is easy to see that the same is true for  $R_x(\theta)$ .

We implement a Clifford +  $T$  gate approximation of  $R_z(\theta)$  using the `gridsynth` package [4], which takes in an angle  $\theta$  and a precision  $\epsilon$  and computes a decomposition for  $R_z(\theta)$  with a  $T$ -gate count within  $O(\log(\log(1/\epsilon)))$  of optimal.

This allows us to modularly use the  $R_x$ ,  $R_y$ , and  $R_z$  gates, noting that they are the most expensive operation, so we should aim to minimize their usage.

#### 3.3 Exponential of a Pauli String

We next tackle the implementation of  $\exp(i\frac{\pi}{7}ZZ)$ . We can observe that

$$\exp(i\theta ZZ) = CX(I \otimes \exp(i\theta Z))CX. \quad (7)$$

Consider the  $R_z$  gate, which is defined as

$$R_z(\theta) = \begin{bmatrix} e^{-i\frac{\theta}{2}} & 0 \\ 0 & e^{i\frac{\theta}{2}} \end{bmatrix}. \quad (8)$$

It is not hard to see that  $\exp(i\theta Z) = R_z(-2\theta)$ . Therefore, we can implement this unitary similarly to the previous one but using only one  $R_z$  gate.

### 3.4 Exponential of a Hamiltonian

Consider the unitary operator  $\exp(i\frac{\pi}{7}H_1)$ , where  $H_1 = XX + YY$ . We implement each summand separately in a construction similar to above. We implement  $\exp(i\frac{\pi}{7}XX)$  by taking the previous circuit and conjugating with  $H$ , and we implement  $\exp(i\frac{\pi}{7}YY)$  by conjugating with  $HS$  and  $S^\dagger H$  on both qubits. This allows us to implement the exponential of this Hamiltonian using only two  $R_z$  gates.

### 3.5 A Curiously Simple Hamiltonian

Suppose we consider a similar Hamiltonian to the one above, but with a slightly different construction. We consider  $\exp(i\frac{\pi}{4}H_2)$ , where  $H_2 = XX + YY + ZZ$ . Can we implement this operator more efficiently than the previous ones?

It turns out that we can, using the SWAP gate, which is defined as

$$\text{SWAP} = \frac{1}{2}(II + XX + YY + ZZ). \quad (9)$$

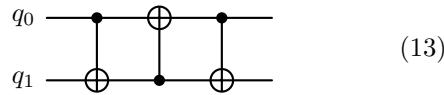
Using the SWAP gate, we can rewrite our exponential as

$$\exp\left(i\frac{\pi}{4}H_2\right) = \exp\left(i\frac{\pi}{4}(2\text{SWAP} - I)\right) \quad (10)$$

$$= e^{-i\frac{\pi}{4}} \exp\left(i\frac{\pi}{2}\text{SWAP}\right) \quad (11)$$

$$= e^{i\frac{\pi}{4}} \text{SWAP} \quad (12)$$

where the coefficient is a global phase that we can discard. Therefore, the Hamiltonian exponential is equivalent to the SWAP gate, which we can easily write using three CNOTs by the XOR swap algorithm:



This form makes the Hamiltonian trivial to simulate.

### 3.6 Two-Qubit Transverse Field Ising Model

We construct this exponential operator, defined as  $\exp(i\frac{\pi}{7}H_3)$ , where  $H_3 = XX + ZI + IZ$ . We construct the  $XX$  term in the same way as described in

Sec. 3.4. For the other two terms, we simply apply  $R_z(-2\theta)$  to each qubit.

### 3.7 State Preparation

Placeholder

### 3.8 A Structured Unitary

We aim to apply the unitary

$$U_1 = \frac{1}{2} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & i & -1 & -i \\ 1 & -1 & 1 & -1 \\ 1 & -i & -1 & i \end{bmatrix}. \quad (14)$$

Notice that this unitary is the two-qubit quantum Fourier transform. Since it only requires  $z$  rotations that are multiples of  $\frac{\pi}{4}$ , we do not need any rotation gates. Our implementation uses only three  $T$  gates.

### 3.9 Another Structured Unitary

## References

- [1] Daniel Gottesman. Theory of fault-tolerant quantum computation. *Phys. Rev. A*, 57:127–137, Jan 1998.
- [2] Adam Sawicki, Lorenzo Mattioli, and Zoltán Zimborás. Universality verification for a set of quantum gates. *Phys. Rev. A*, 105:052602, May 2022.
- [3] Scott Aaronson and Daniel Gottesman. Improved simulation of stabilizer circuits. *Phys. Rev. A*, 70:052328, Nov 2004.
- [4] Neil J. Ross and Peter Selinger. Optimal ancilla-free clifford+ $t$  approximation of  $z$ -rotations. *Quantum Info. Comput.*, 16(11–12):901–953, September 2016.