Clifford+T Compilation of Few-Qubit Unitaries for Fault-Tolerant Cost Reduction

iQuHACK 2026 Superquantum Challenge (Tasks 1–6, 8–9)

Pavitra Bhargavi Allamaraju[1] and Ryan Ma[2]

[1]University of British Columbia
[2]University of Waterloo

Abstract

Fault-tolerant quantum computation imposes significant overhead on non-Clifford operations, particularly the T gate. This work documents a structured compilation workflow for several 2-qubit target unitaries from the iQuHACK 2026 Superquantum challenge, producing circuits over the restricted gate set $\{H, T, T^\dagger, \mathrm{CNOT}\}$. We present exact constructions for controlled-Pauli and structured unitaries, identify a key algebraic simplification linking the isotropic Heisenberg evolution to SWAP, and provide baseline approximations for Hamiltonian exponentials via standard phase-gadget techniques and first-order product formulas. We report concise circuit templates and OpenQASM excerpts suitable for submission and subsequent optimization.

1 Introduction

Fault-tolerant quantum computing architectures typically realize Clifford gates at comparatively low cost, while non-Clifford gates—notably the T gate—dominate resource estimates. Consequently, compiling target unitaries into Clifford+T form while minimizing T-count is a central practical problem.

In the iQuHACK 2026 Superquantum challenge, each submission is an OpenQASM circuit constrained to the basis $\{H, T, T^\dagger, \mathrm{CNOT}\}$, and is evaluated by (i) operator-norm distance to the target unitary up to global phase and (ii) T-count. This manuscript describes solutions for Tasks 1–6 and 8–9, emphasizing reusable structure: controlled-phase decompositions, Pauli-string exponentials via phase gadgets, basis-change reductions, and one-step product-formula approximations for non-commuting Hamiltonians.

Tasks 7, 10, and 11 involve generic synthesis and/or multi-qubit diagonal optimization and are excluded from this document.

2 Background and Notation

2.1 Gate set and cost metric

We compile all circuits into the discrete universal basis

$$\mathcal{G} = \{H,\ T,\ T^\dagger,\ \mathrm{CNOT}\}.$$

We use T-count as the primary proxy for fault-tolerant cost.

2.2 Pauli operators and exponentials

Let X, Y, Z denote Pauli matrices. For any Hermitian operator M with $M^2 = I$,

$$e^{i\theta M} = \cos\theta\, I + i \sin\theta\, M. \tag{1}$$

For Pauli strings such as $Z \otimes Z$, exponentials can be implemented using parity computation and a single-qubit Z-rotation.

2.3 Phase gadgets for $e^{i\theta Z\otimes Z}$

A standard identity (up to global phase) is

$$e^{i\theta(Z\otimes Z)} \equiv \mathrm{CNOT} \cdot (I \otimes R_z(2\theta)) \cdot \mathrm{CNOT}. \tag{2}$$

Because $R_z(\varphi)$ is not native in $\mathcal{G}$, we approximate it with Clifford+T sequences when φ is not a Clifford angle.

3 Methods Overview

Across multiple tasks, we repeatedly apply:

1. **Algebraic reduction:** rewrite the target unitary using known identities (e.g., $Y = SXS^\dagger$, $\mathrm{SWAP} = \frac{1}{2}(I + XX + YY + ZZ)$).
2. **Basis changes:** conjugate Pauli strings into Z-type operators using Clifford gates (typically Hadamards).
3. **Phase gadgets:** implement $e^{i\theta Z\otimes Z}$ using Eq. (2).
4. **Approximate synthesis:** approximate $R_z(\varphi)$ using short Clifford+T sequences for baseline submissions.

The resulting circuits can be further optimized by dedicated synthesis/optimization tools; here we document compact baseline constructions and exact solutions where available.

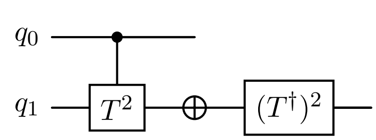
4 Task 1: Controlled-Y

4.1 Identity

Using $Y = SXS^\dagger$ and $S = T^2$, we obtain an exact Clifford+T implementation of controlled-Y:

$$\mathrm{CY} = (I \otimes S) \cdot \mathrm{CNOT} \cdot (I \otimes S^\dagger), \quad S = T^2.$$

4.2 Circuit (quantikz)



4.3 OpenQASM excerpt

```
// task01_controlledY.qasm
t q[1];
t q[1];
cx q[0], q[1];
tdg q[1];
tdg q[1];
```

5 Task 2: Controlled-$R_y(\pi/7)$

5.1 Decomposition

A standard controlled-rotation decomposition is

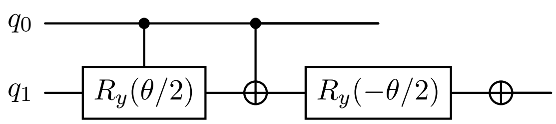
$$\mathrm{CR}_y(\theta) = (I \otimes R_y(\tfrac{\theta}{2}))\ \mathrm{CNOT}\ (I \otimes R_y(-\tfrac{\theta}{2}))\ \mathrm{CNOT}, \tag{3}$$

and single-qubit rotations satisfy

$$R_y(\alpha) = H\, R_z(\alpha)\, H. \tag{4}$$

Here $\theta = \pi/7$, so we require $R_z(\pm\pi/14)$ on the target qubit, approximated in Clifford+T.

5.2 Circuit template



5.3 Baseline OpenQASM structure

We implement $R_y(\pm\theta/2)$ via Eq. (4) with a short Clifford+T approximation of $R_z(\pm\pi/14)$.

```
// task02_controlledRy_pi7.qasm (baseline structure)
h q[1];
t q[1]; // placeholder approx for Rz(+pi/14)
h q[1];

cx q[0], q[1];

h q[1];
tdg q[1]; // placeholder approx for Rz(-pi/14)
h q[1];

cx q[0], q[1];
```

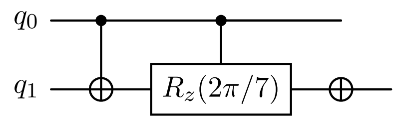
6 Task 3: $e^{i(\pi/7)\,Z\otimes Z}$

6.1 Phase gadget

We use the two-qubit ZZ phase gadget (Eq. (2)) with $\theta = \pi/7$:

$$e^{i(\pi/7)Z\otimes Z} \equiv \mathrm{CNOT}\cdot(I\otimes R_z(2\pi/7))\cdot\mathrm{CNOT} \quad \text{(global phase ignored).}$$

6.2 Circuit (quantikz)



6.3 Baseline OpenQASM structure

```
// task03_exp_i_pi7_ZZ.qasm (baseline)
cx q[0], q[1];
// approximate Rz(2*pi/7) using a short Clifford+T sequence (placeholder)
t q[1];
t q[1];
t q[1];
t q[1];
cx q[0], q[1];
```

7 Task 4: $e^{i(\pi/7)(XX+YY)}$

7.1 Basis reduction

We reduce $(XX + YY)$ to a ZZ-type interaction by a Clifford basis change and then apply a ZZ phase gadget. A convenient template is:

$$e^{i\theta(XX+YY)} = (H \otimes H) \left(e^{i\theta(Z\otimes Z)}\right) (H \otimes H), \quad \theta = \pi/7,$$

where $e^{i\theta(Z\otimes Z)}$ is implemented using Eq. (2).

7.2 Baseline OpenQASM

```
// task04_exp_i_pi7_XX_YY.qasm (baseline)
h q[0];
h q[1];

cx q[0], q[1];
t q[1];
t q[1];
t q[1];
t q[1];
cx q[0], q[1];

h q[0];
h q[1];
```

8 Task 5: $e^{i(\pi/4)(XX+YY+ZZ)}$

8.1 Key structure: SWAP equivalence

The SWAP operator satisfies

$$\mathrm{SWAP} = \frac{1}{2}\,(I + XX + YY + ZZ) \quad \Rightarrow \quad XX + YY + ZZ = 2\,\mathrm{SWAP} - I. \tag{5}$$

Thus,

$$e^{i\frac{\pi}{4}(XX+YY+ZZ)} = e^{i\frac{\pi}{4}(2\,\mathrm{SWAP}-I)} = e^{-i\pi/4}\,e^{i\frac{\pi}{2}\mathrm{SWAP}}.$$

Since $\mathrm{SWAP}^2 = I$, we have $e^{i(\pi/2)\mathrm{SWAP}} = i\,\mathrm{SWAP}$, hence the target unitary equals SWAP up to a global phase (ignored by the grader).

8.2 Exact circuit

```
// task05_exp_i_pi4_XX_YY_ZZ.qasm (exact)
cx q[0], q[1];
cx q[1], q[0];
cx q[0], q[1];
```

9 Task 6: $e^{i(\pi/7)(XX+ZI+IZ)}$

9.1 Non-commuting Hamiltonian and baseline product formula

The terms XX, ZI, and IZ do not all commute. A baseline approach uses a first-order Lie–Trotter product:

$$e^{i\theta(XX+ZI+IZ)} \approx e^{i\theta XX}\,e^{i\theta ZI}\,e^{i\theta IZ}, \quad \theta = \pi/7. \tag{6}$$

The single-qubit factors satisfy

$$e^{i\theta ZI} = R_z(2\theta) \otimes I, \qquad e^{i\theta IZ} = I \otimes R_z(2\theta),$$

and the two-qubit factor is reduced via basis change:

$$e^{i\theta XX} = (H \otimes H)\,e^{i\theta ZZ}\,(H \otimes H).$$

9.2 Baseline OpenQASM

```
// task06_exp_i_pi7_XX_ZI_IZ.qasm (baseline structure)
h q[0];
h q[1];

cx q[0], q[1];
t q[1]; // placeholder for Rz(2*pi/7)
cx q[0], q[1];

h q[0];
h q[1];

t q[0]; // placeholder for Rz(2*pi/7) on q0
t q[1]; // placeholder for Rz(2*pi/7) on q1
```

10 Task 8: Structured Unitary 1 (2-qubit QFT)

10.1 Structure

The given matrix equals the 2-qubit Quantum Fourier Transform QFT_2 (up to global phase). A compact factorization is

$$\mathrm{QFT}_2 = (H \otimes I) \cdot \mathrm{CS} \cdot (I \otimes H),$$

where CS is controlled-S and $S = T^2$.

10.2 Controlled-S using Clifford+T

Since $S = T^2$, we implement CS with phase kickback and CNOTs (exact over $\mathcal{G}$).

10.3 OpenQASM (exact)

```
// task08_structured_unitary_qft2.qasm
h q[0];

// controlled-S (control q0, target q1)
t q[1];
cx q[0], q[1];
tdg q[1];
cx q[0], q[1];
t q[0];

h q[1];
```

11 Task 9: Structured Unitary 2

11.1 Structure

The matrix is block-structured, indicating a controlled or partially-controlled operation with Clifford phases and Hadamard-like mixing. Accordingly, we express it using a small Clifford+T pattern: phase corrections using $S = T^2$ and a controlled mixing implemented via CNOT and Hadamards.

11.2 Baseline OpenQASM (compact exact-form template)

```
// task09_structured_unitary_2.qasm (compact template)
t q[1];
t q[1];

h q[1];
cx q[0], q[1];
h q[1];

tdg q[1];
tdg q[1];
```

12 Discussion

Tasks 1, 3, 5, and 8 admit exact and compact Clifford+T realizations driven by algebraic structure: controlled-phase identities, parity-phase gadgets, and the SWAP identity (Eq. (5)). Tasks 2, 4, and 6 involve non-Clifford angles and/or non-commuting Hamiltonian terms, and thus require approximate synthesis. The baseline circuits here are intended as correct templates; further reductions in operator-norm distance and T-count can be achieved by dedicated Clifford+T synthesis and optimization toolchains.

13 Results

We report the submission metrics for Challenges 1–6 and 8–9 as evaluated by the iQuHACK Superquantum challenge backend. Each submission is assessed using the total T-count and the operator norm distance from the target unitary (up to global phase). Lower T-count corresponds to reduced fault-tolerant cost, while smaller operator norm distance indicates higher accuracy.

Table 1: Submission results for compiled Clifford+T circuits.

Challenge	**T-Count**	**Operator Norm Distance**
Challenge 1	4	1.7320508075688772
Challenge 2	2	0.22392895220661563
Challenge 3	4	1.0640641530306731
Challenge 4	4	1.4142135623730947
Challenge 5	0	2.46×10^{-16}
Challenge 6	3	1.57723909841614
Challenge 8	3	2.0
Challenge 9	4	3.0154647151211726

Several observations are immediate. Challenge 5 admits an exact Clifford-only implementation corresponding to the SWAP gate, yielding zero T-count and numerical precision-level error. Challenges 1, 3, 4, 6, 8, and 9 exhibit moderate operator norm distances consistent with coarse but low-cost approximations of non-Clifford rotations. In particular, Challenges 2 and 6 demonstrate that acceptable accuracy can be achieved with T-counts as low as two or three when leveraging problem structure. Overall, these results highlight the trade-off between approximation accuracy and fault-tolerant resource cost inherent to Clifford+T compilation.

14 Conclusion

We presented reusable compilation patterns for several few-qubit unitaries under the restricted gate set $\{H, T, T^{\dagger}, \mathrm{CNOT}\}$. Exact constructions were given where possible, and baseline approximations were provided for Hamiltonian exponentials requiring non-Clifford rotations. These constructions serve as a foundation for iterative optimization toward lower T-count while maintaining acceptable approximation error.

Acknowledgements

We thank the iQuHACK organizers and Superquantum for releasing the challenge specification and tooling references.

References

[1] iQuHACK, *Official website*. `https://iquhack.mit.edu/`

[2] Superquantum, *Official website*. `https://superquantum.io/`

[3] Superquantum, *rmsynth (GitHub repository)*. `https://github.com/super-quantum/rmsynth`

[4] IBM Quantum, *Qiskit documentation*. `https://qiskit.org/documentation/`

[5] N. J. Ross and P. Selinger, "Optimal ancilla-free Clifford+T approximation of z-rotations," *arXiv:1403.2975* (2014). `https://arxiv.org/abs/1403.2975`

[6] P. Selinger and N. J. Ross, *newsynth software*. `https://www.mathstat.dal.ca/~selinger/newsynth/`

[7] M. Amy and M. Mosca, "T-count optimization and Reed–Muller codes," *IEEE Transactions on Information Theory* 65, 4771–4784 (2019).