QC “Qptimizer”

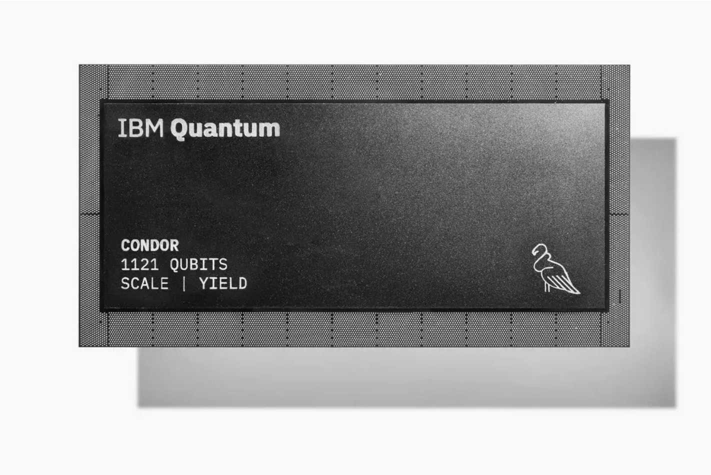
Hardware-aware Optimal Quantum Circuit Cutting and Knitting

Thang Tran
Advisors: Emmanouil Giortamis, Francisco Romão
Chair of Computer Systems
https://dse.in.tum.de/

15.09.2023 – 15.03.2024

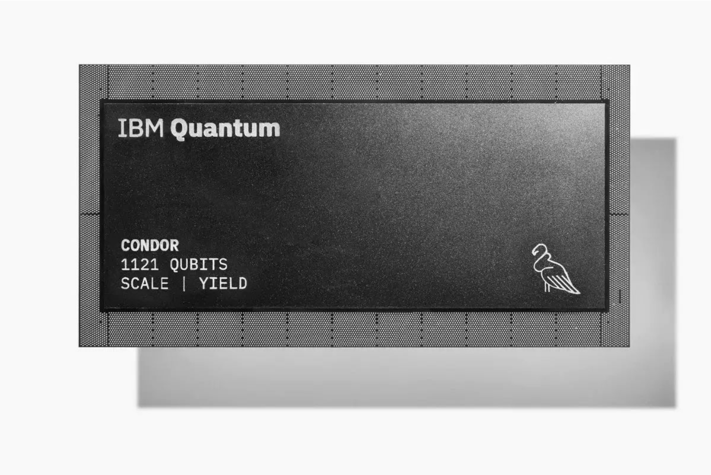
Motivation: Quantum Computing

- Potentially more powerful at solving problems compared to classical computers
  - Superposition
  - Entanglement
- high potential in: machine learning, factorization, optimization, and other



Challenges

- Size: small
    - Useful applications: millions of qubits
    - Largest QPU: IBM Condor (1121 qubits)
- noisy (NISQ)
    - Limited connectivity
    - Not scalable / fault-tolerance



Potential Solutions

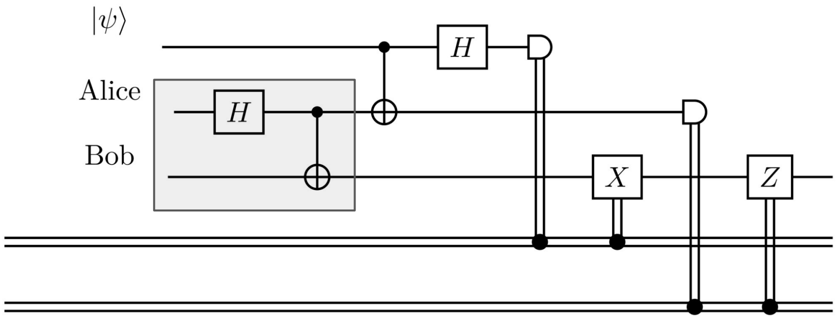
- Teleportation
  - latency
  - not yet realizable
- Circuit cutting and knitting
  - exponential overheads => How to cut optimally

Outline

- ~~Motivation~~
- Background
- Design
- Implementation
- Evaluation

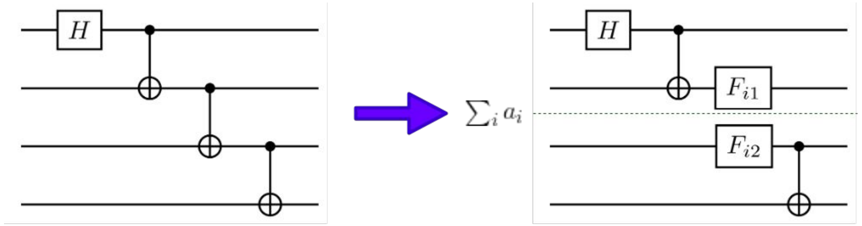
Background: Teleportation

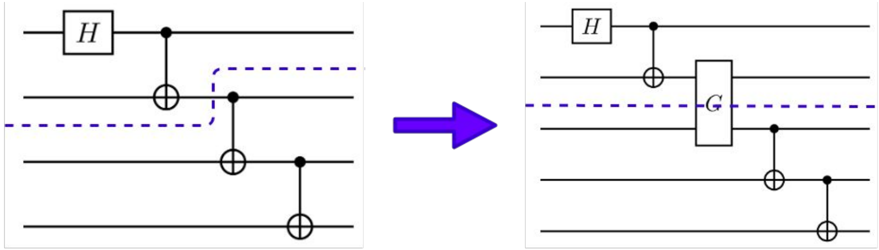
- Transfer unknown state $|\psi\rangle$
- Required a pre-shared EPR pairs
  - Two ancilla qubits
  - Classical communication (latency)
  - Can be geographically distributed



Background: Circuit cutting and knitting

- Two type of dependencies:
  - Wire dependencies
  - Gate dependencies
- Two steps:
  - Partitioning
  - Knitting
- Gate knitting: $\mathcal{U} = \sum_i a_i F_i$

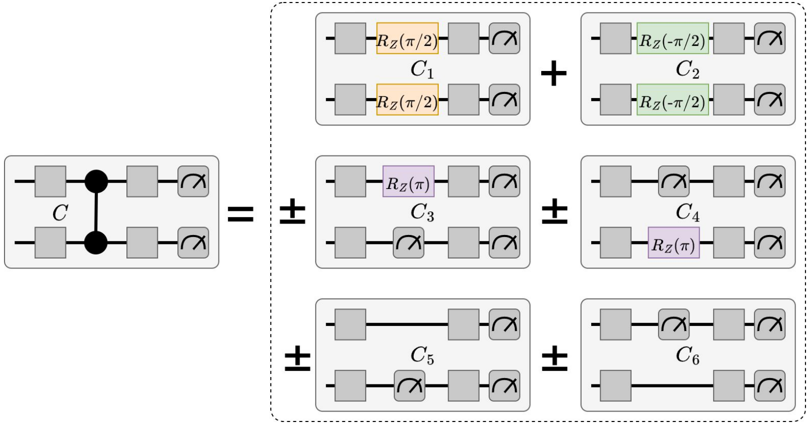




Background: Gate Virtualization

- Virtualize any binary gates using only unary gates
- Original probability distribution obtained using knitting formula
  - Rules of stochastically independent events

$$P_i(|x\rangle) = P_i^1(|x\rangle) \cdot P_i^2(|x\rangle)$$

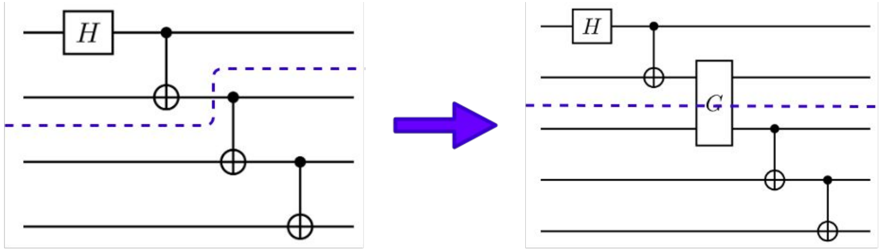


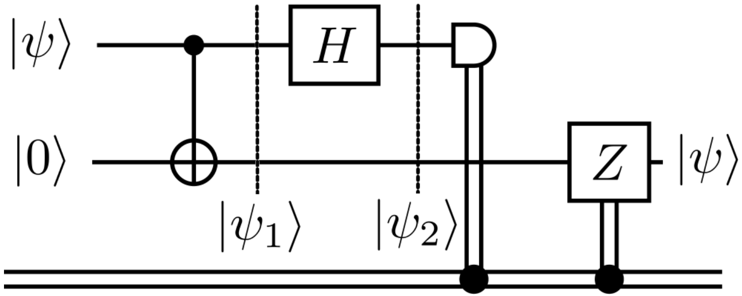
[1]

[1] N. Tornow. "DQS: A Framework for Efficient Distributed Simulation of Large Quantum Circuits." In: (2022).

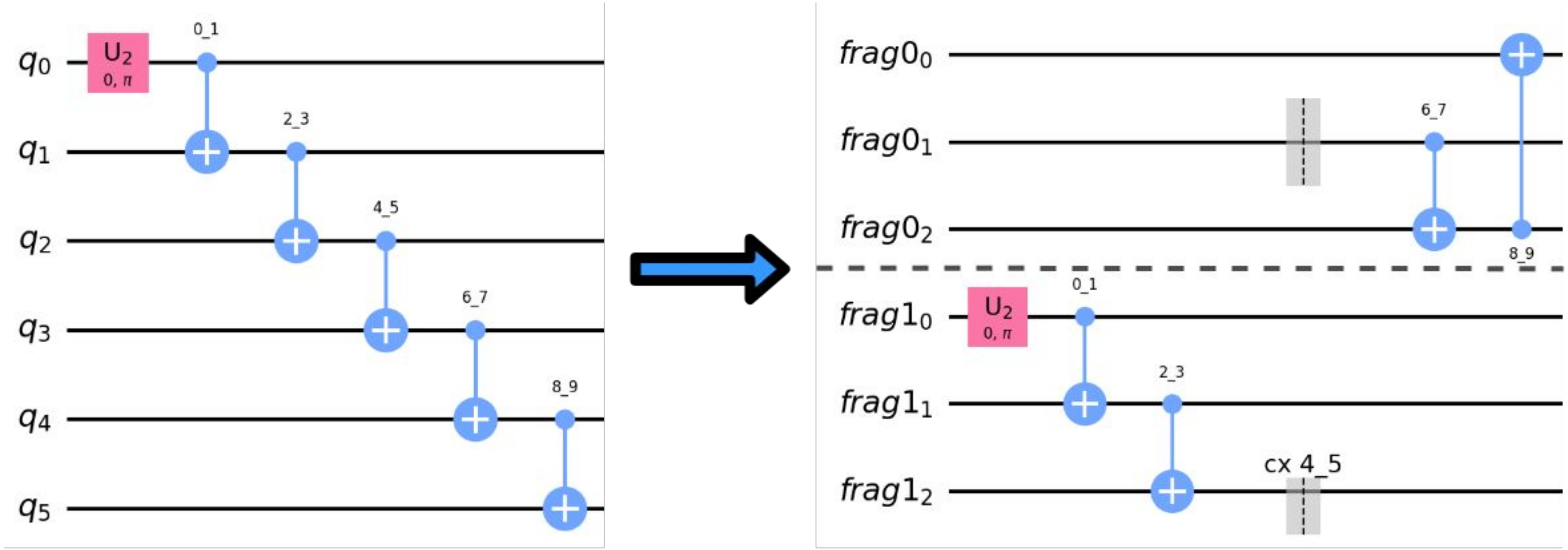
Background: Circuit cutting and knitting

- Multiple options:
    - Swap gate
    - Teleportation
    - Move Circuit

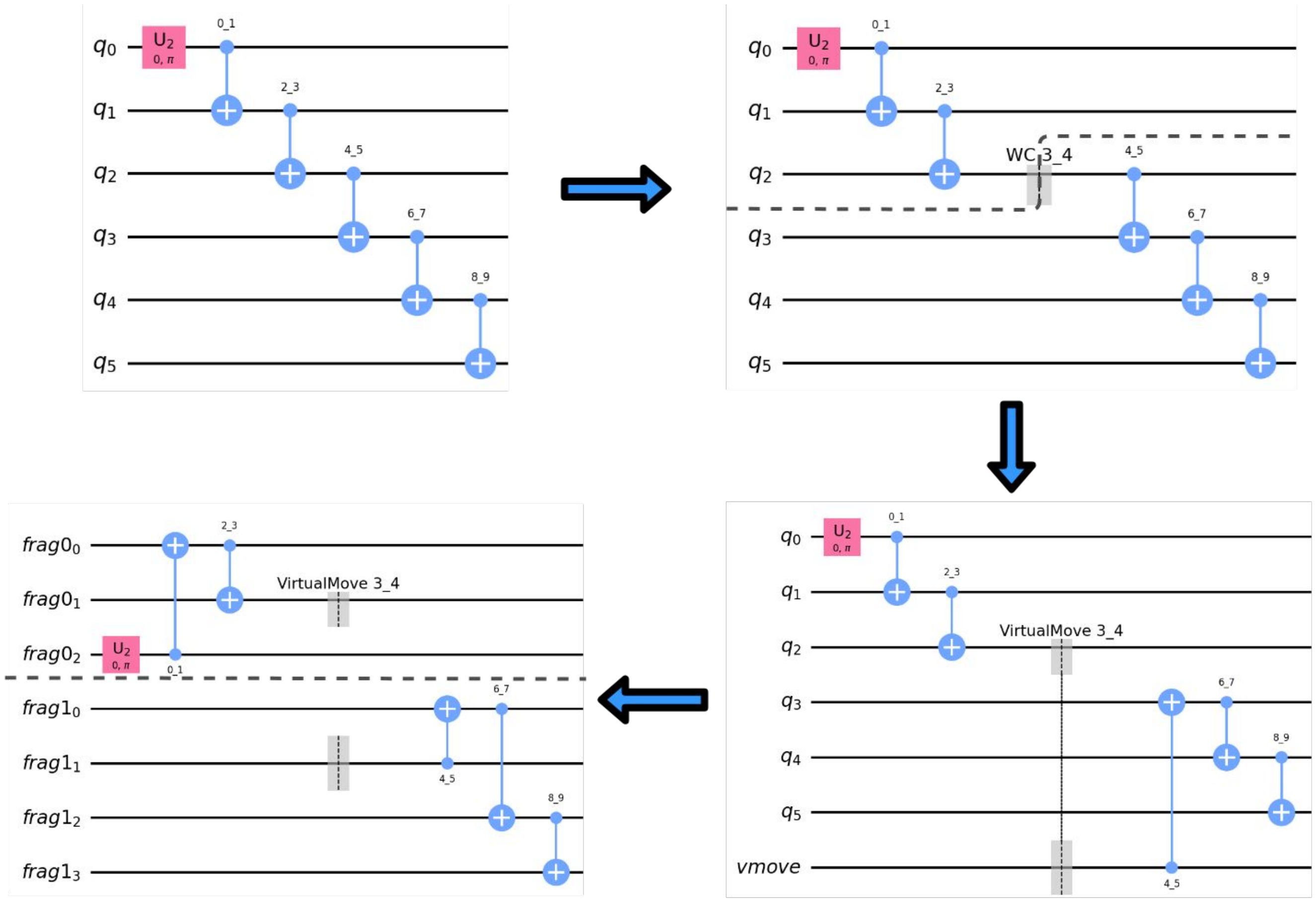




Background: Demonstration of Gate Cut



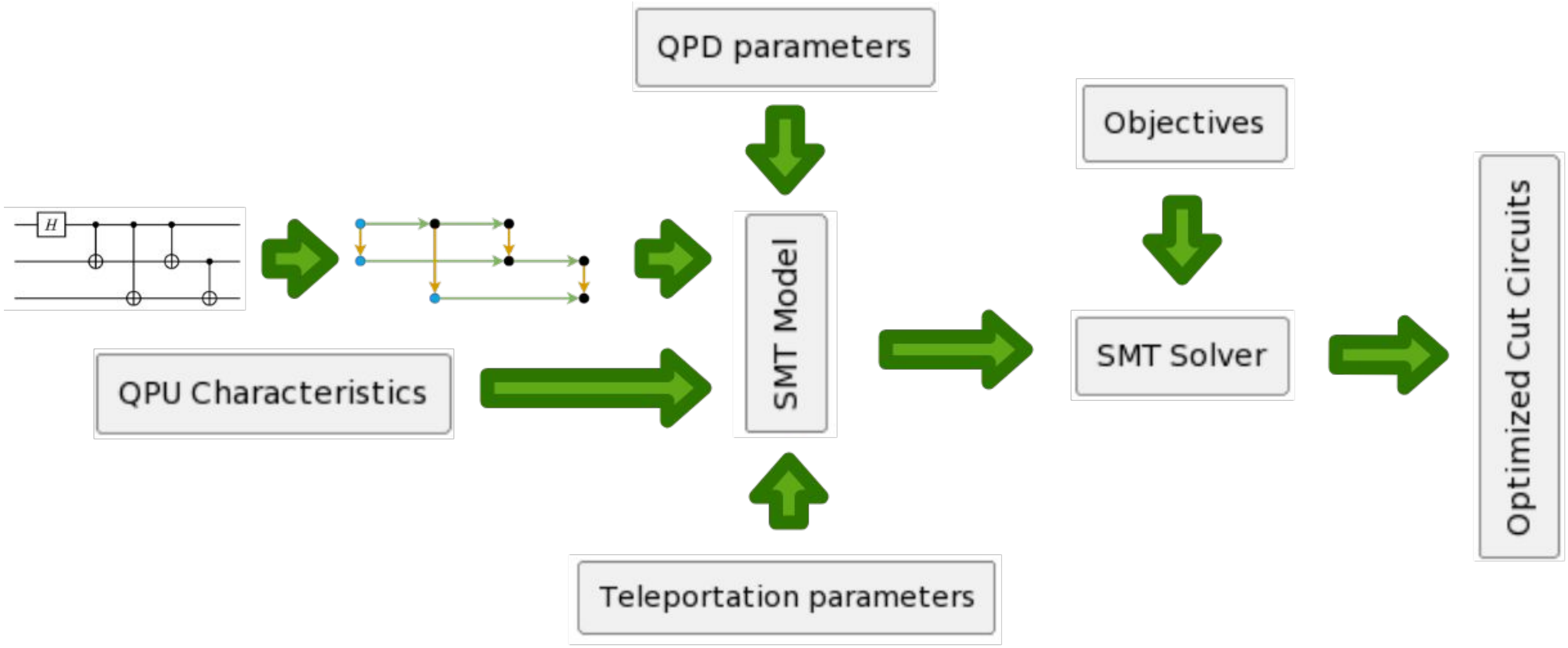
Background: Demonstration of Wire Cut



Outline

- ~~Motivation~~
- ~~Background~~
- Design
- Implementation
- Evaluation

Design: System Overview



Design: Details

- Unary gates are ignored
- Transpile gates into binary gates
- A gate creates a gate-cut-edge
  - creates 2 vertices
    - blue vertices: first vertex of a qubit
    - black vertices: normal vertices
- wire-cut-edge: 2 adjacent vertices on a qubit

Design: Details

- qubit topology
- noise model

QPU Characteristics

Design: Details

QPD parameters

- sampling overhead
- latency
- …

Teleportation parameters

Design: Details

- minimize total number of cuts
- minimize total sampling overhead
- minimize number of qubits per partitions
- …

Objectives

Outline

- ~~Motivation~~
- ~~Background~~
- ~~Design~~
- Implementation
- Evaluation

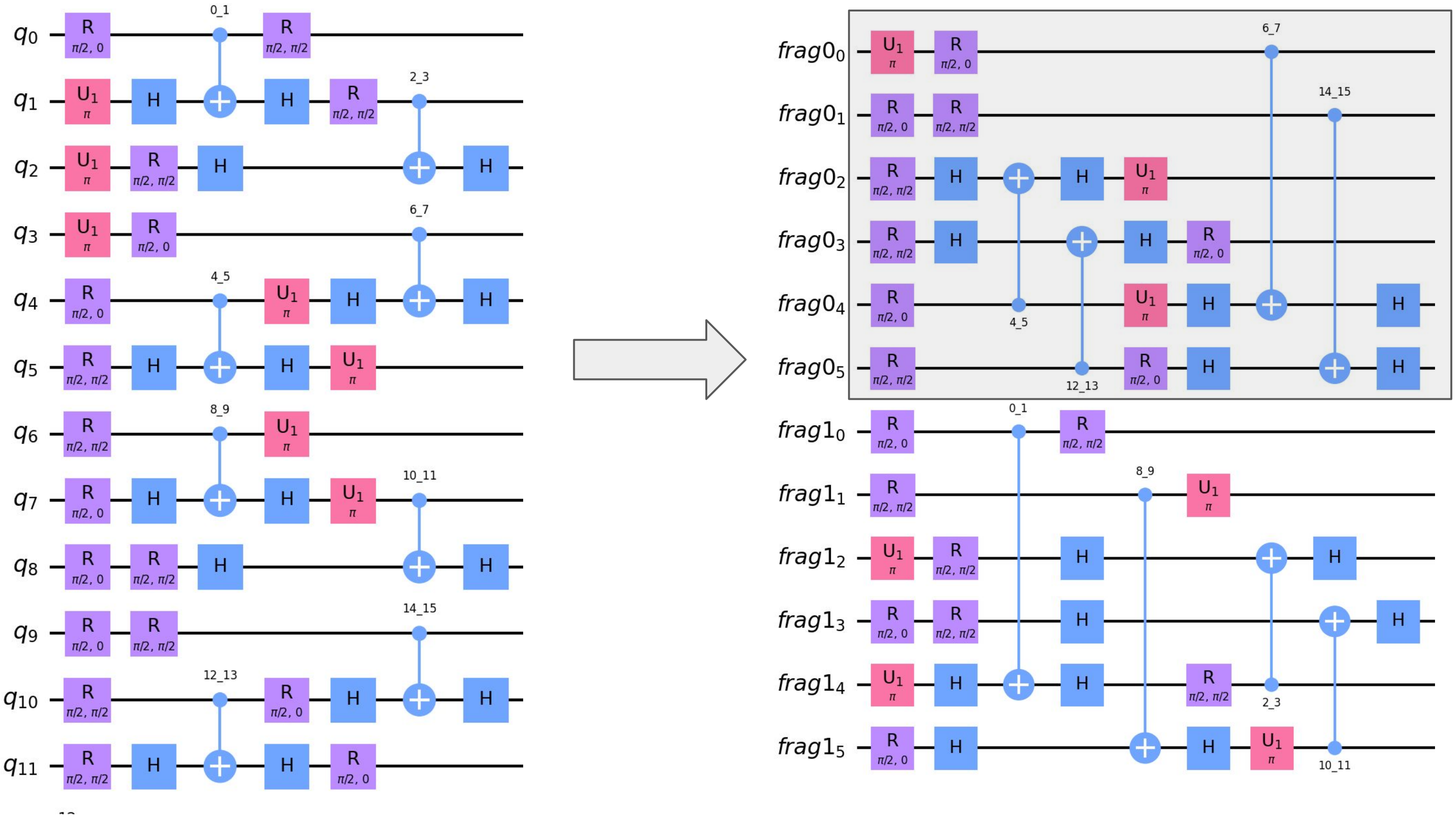
Implementation

- python
- Z3-solver (SAT solver)
- QOS::QVM framework
- clingo grounder (ASP)

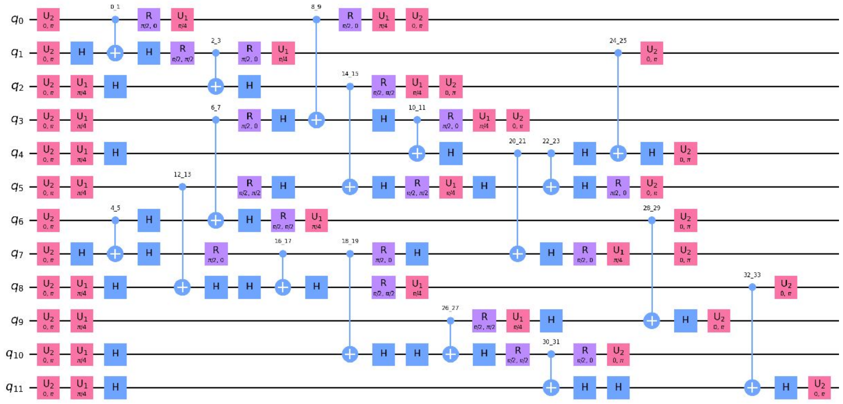
Outline

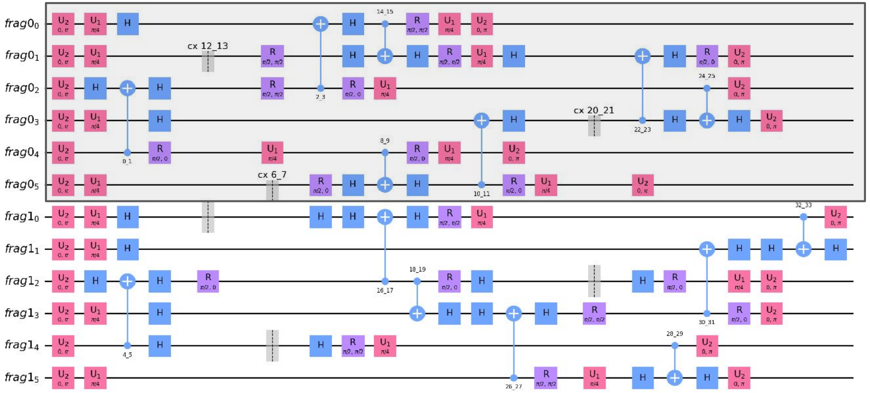
- ~~Motivation~~
- ~~Background~~
- ~~Design~~
- ~~Implementation~~
- Evaluation

Evaluation: Sycamore 12 qubits

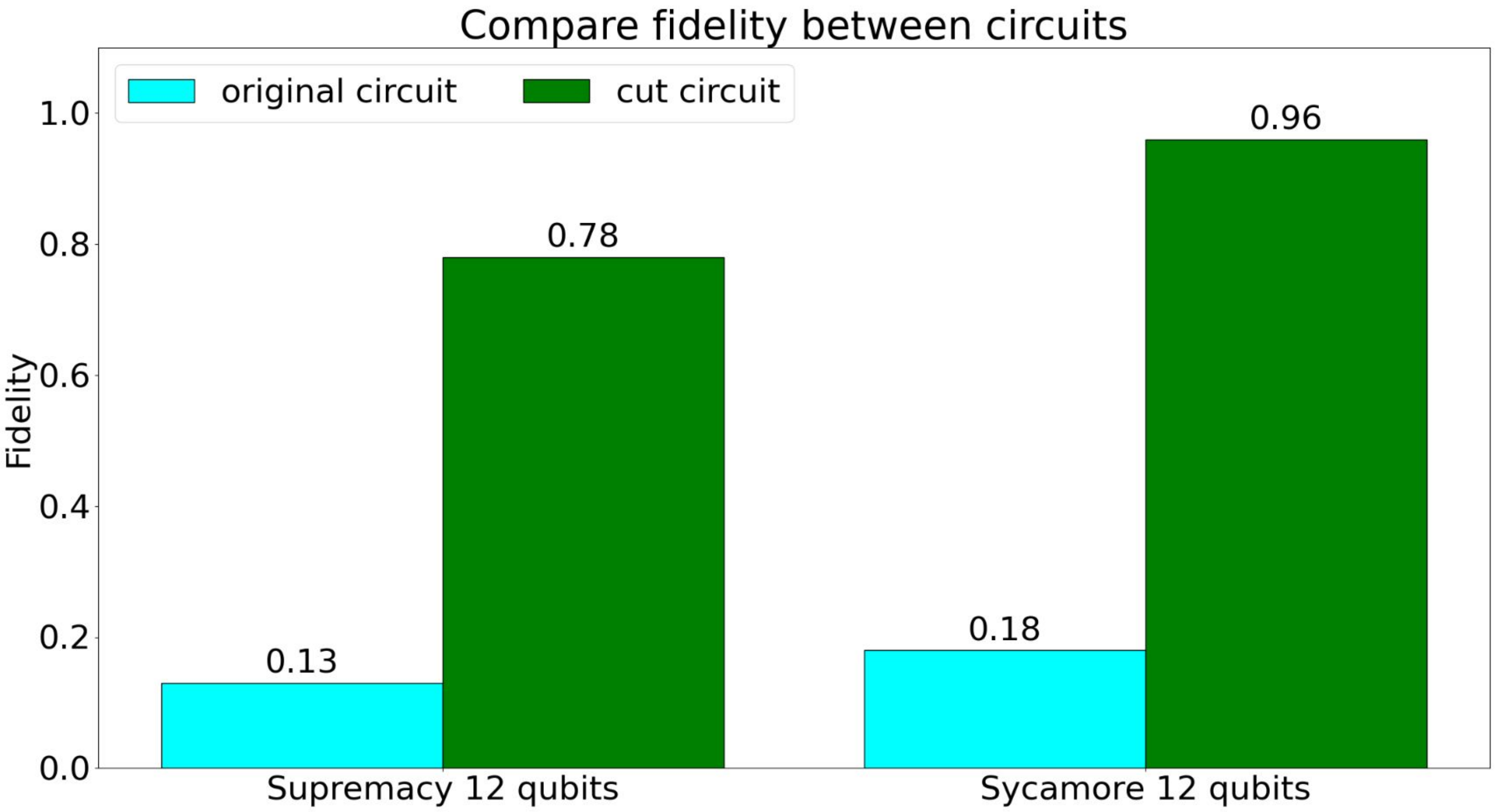


Evaluation: Supremacy 12 qubits

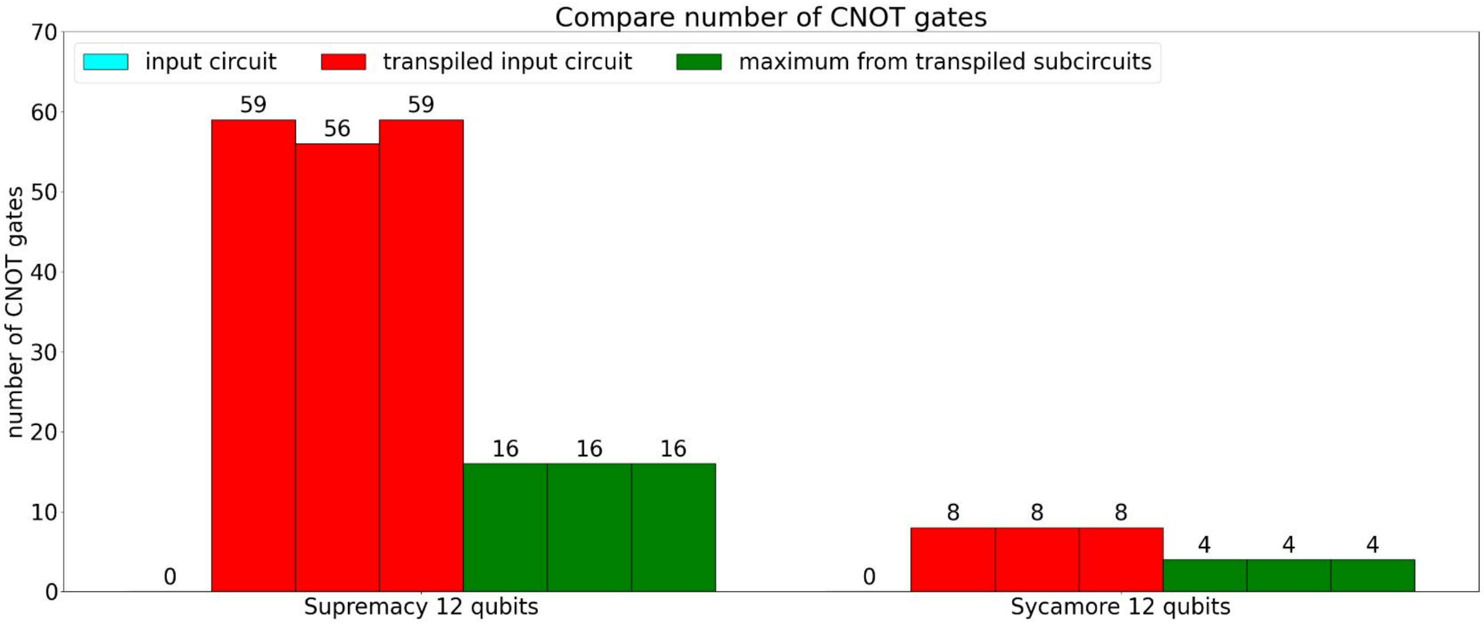




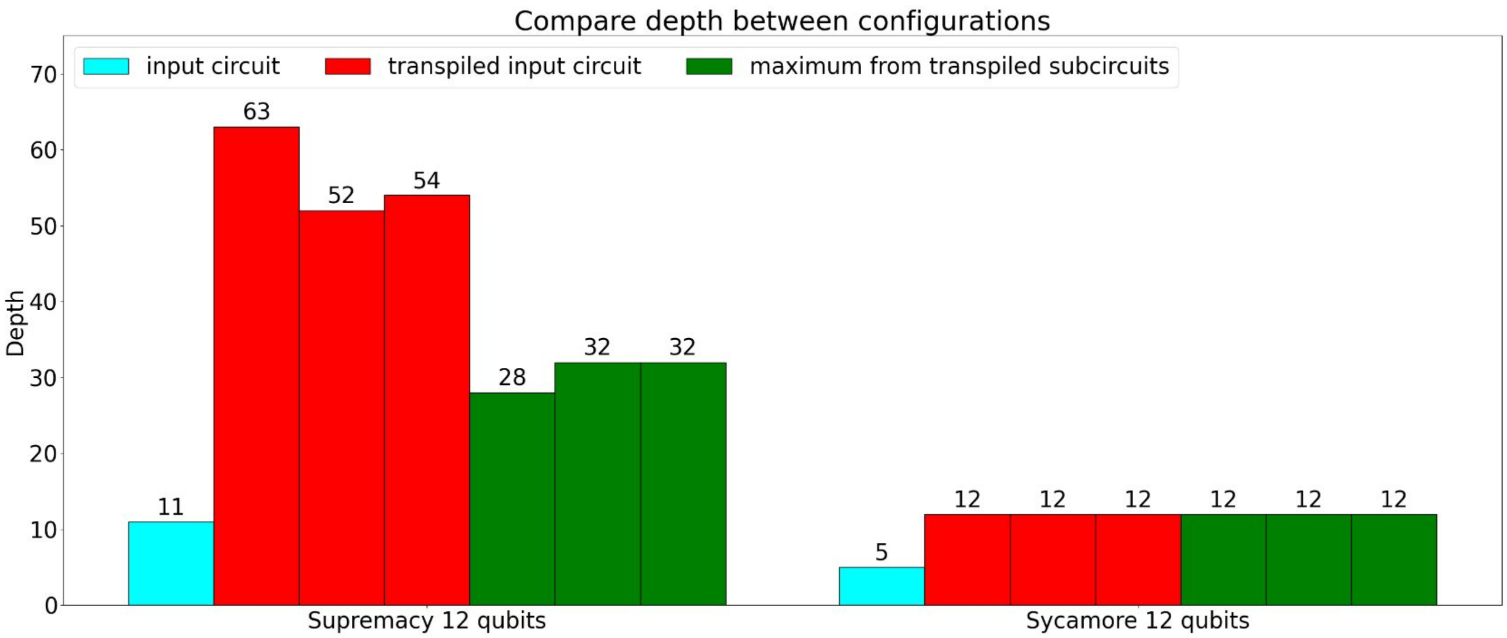
Evaluation: Fidelity



Evaluation: Number of CNOTs



Evaluation: Circuit depth



Summary

- Larger circuits:
  - harder to execute
  - produce less accurate results
- Our solution:
  - fidelity is improved
  - Cutting could take long
  - Limited to small circuits (<= 25 qubits)
  - Increased overhead is significant
- Future works:
  - pre-processing steps fragment the circuits (in the case of sycamore 12 qubits)
  - Improve solver efficiency
  - Implement teleport