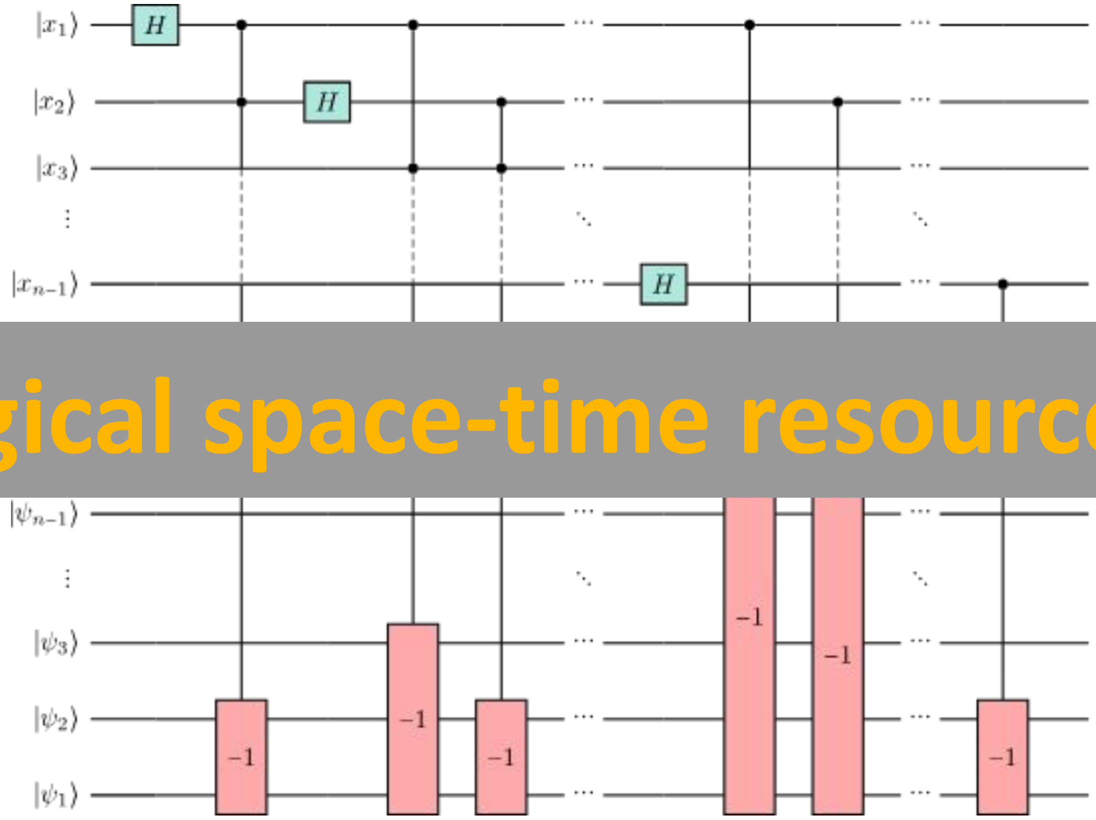
CMPT 478/981 Spring 2025
Quantum Circuits & Compilation

Matt Amy

Today’s agenda

- Surface codes & their space-time requirements
- Quantum algorithms
- Single-qubit approximation
- Classical logic synthesis

Logical space-time resources

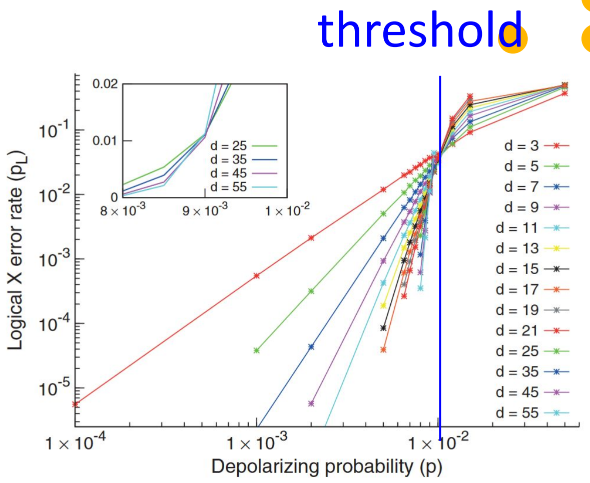


Recall: stabilizer codes

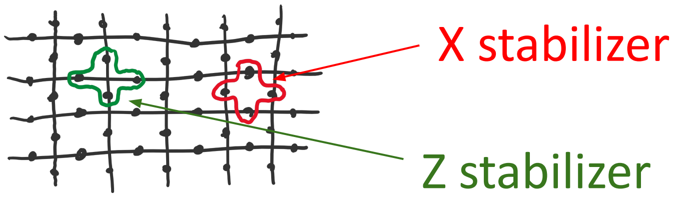
- Last class we talked (briefly) about stabilizer codes
- Given an Abelian (commuting) subgroup S of n-qubit Pauli operators
  - The stabilizer code defined by S is the +1 eigenspace of all P in S
  - If S has k generators, it encodes n-k logical qubits
- Example: 3-bit repetition code has stabilizer $\langle Z\otimes Z\otimes I, I\otimes Z\otimes Z\rangle$

The surface code

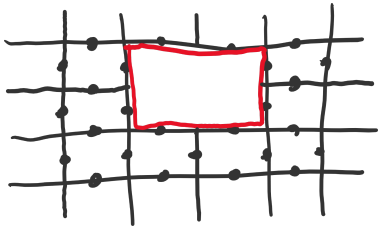
- Based on Kitaev's toric code
- Since 2010's, most promising candidate for FTQEC
  - Threshold around 10^{-2} vs 10^{-5} for Steane code
  - Can be implemented on a 2D lattice ("low density")



- Define two types of stabilizers on a 2D lattice



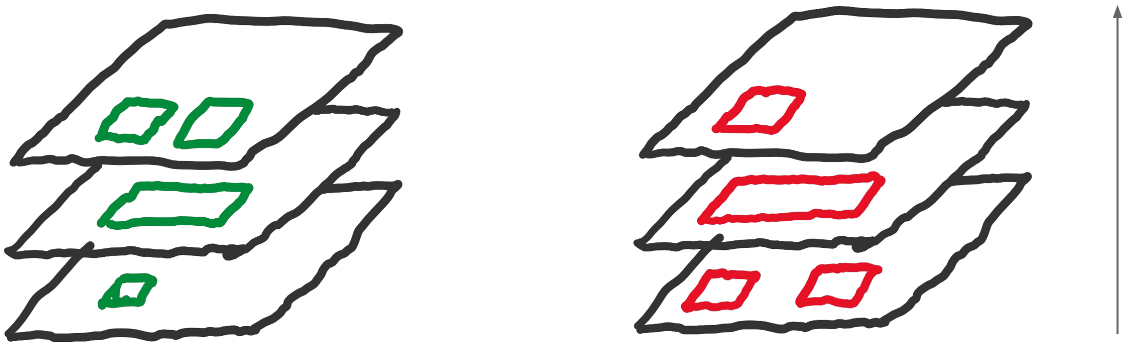
- "Turn off" stabilizers in a section (a defect) to add qubits:



Fault tolerant (Clifford) gates in the surface code

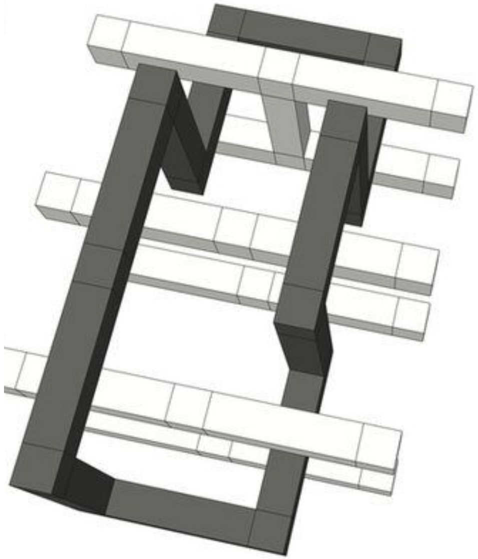
- Circa 2010's: Braiding

- Now: Lattice surgery

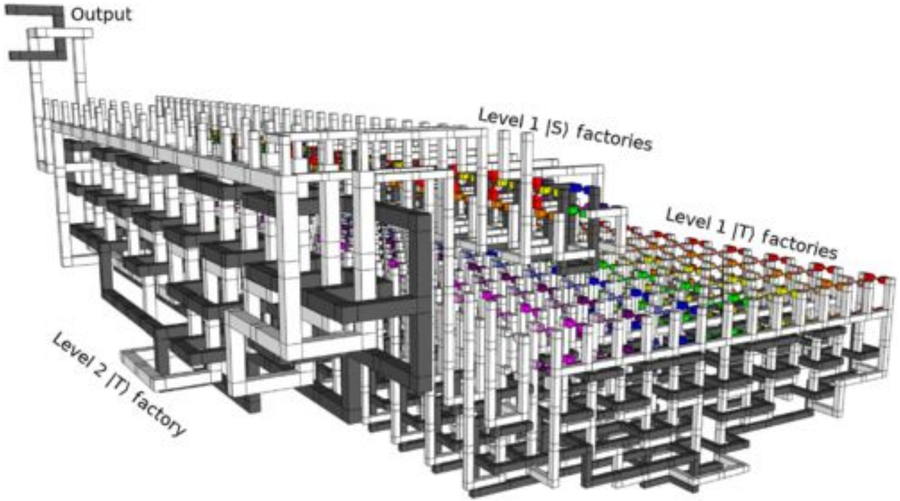


Relative space-time volumes

CNOT:

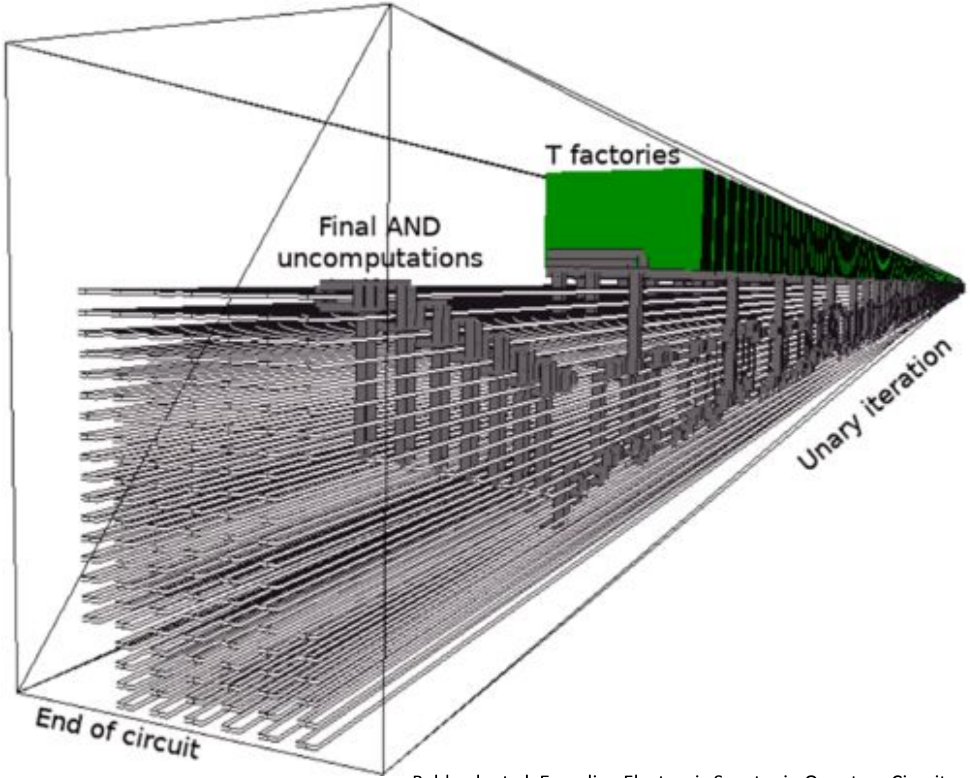


T distillation factory:



Babbush et al. Encoding Electronic Spectra in Quantum Circuits with Linear T Complexity. Phys. Rev. X 2018.

A compiled FTQEC computation



Babbush et al. Encoding Electronic Spectra in Quantum Circuits with Linear T Complexity. Phys. Rev. X 2018.

Lattice surgery

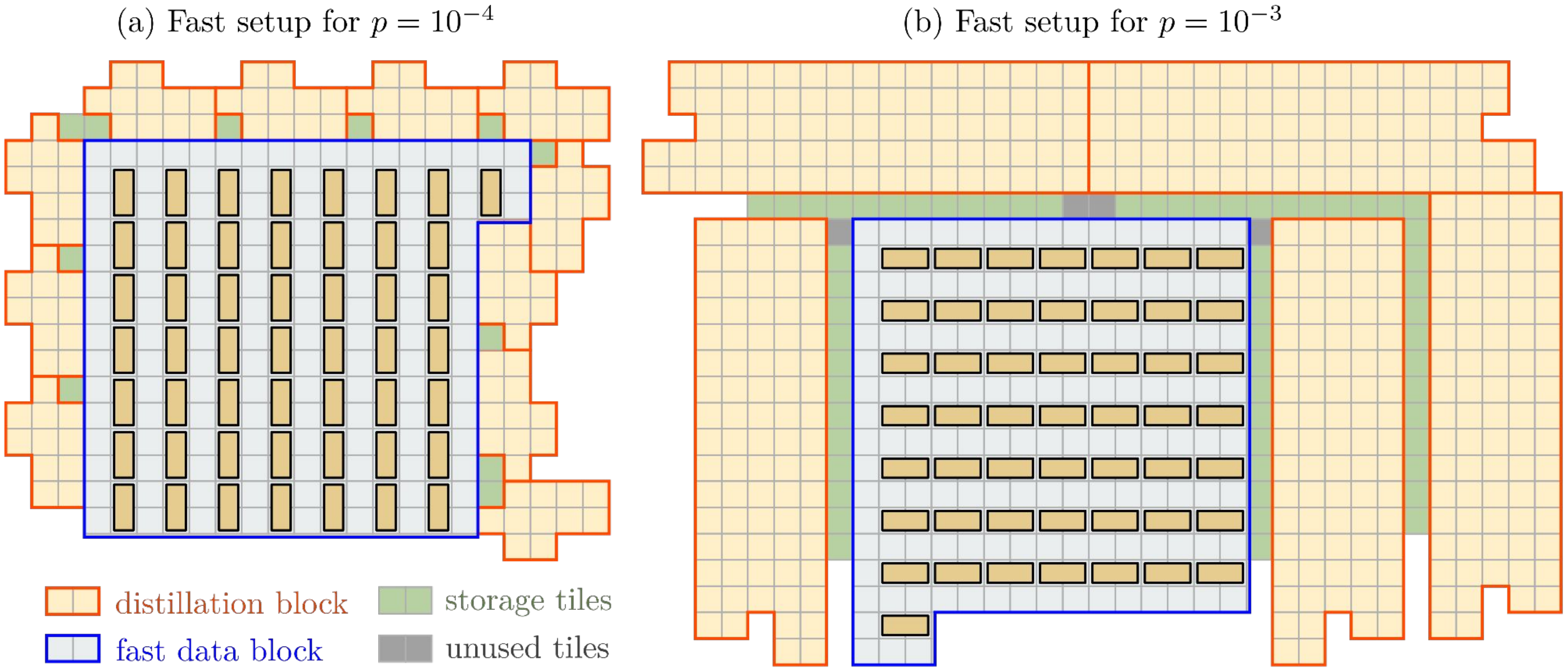


Figure 23: Fast setups using fast data blocks and 11 15-to-1 distillation blocks for $p = 10^{-4}$ or 5 116-to-12 distillation block for $p = 10^{-3}$.

Litinski, A Game of Surface Codes, Quantum 2019.

Maybe not…

arXiv > quant-ph > arXiv:1905.06903

Quantum Physics

[Submitted on 16 May 2019 (v1), last revised 6 Nov 2019 (this version, v3)]

Magic State Distillation: Not as Costly as You Think

Daniel Litinski

arXiv > quant-ph > arXiv:2409.17595v1

Quantum Physics

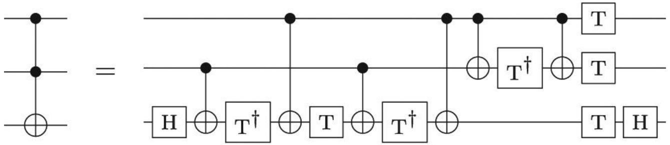
[Submitted on 26 Sep 2024]

Magic state cultivation: growing T states as cheap as CNOT gates

Craig Gidney, Noah Shutty, Cody Jones

What about other non-Clifford gates?

- Toffoli+Hadamard is also universal
  - …but the Toffoli gate is best implemented by using 7 T gates (**optimal**) in most cases



- What about gates from higher levels?
  - …relies on |T> states to implement via gate teleportation
  - …but can result in more efficient implementations in some regimes


arXiv > quant-ph > arXiv:1603.04230

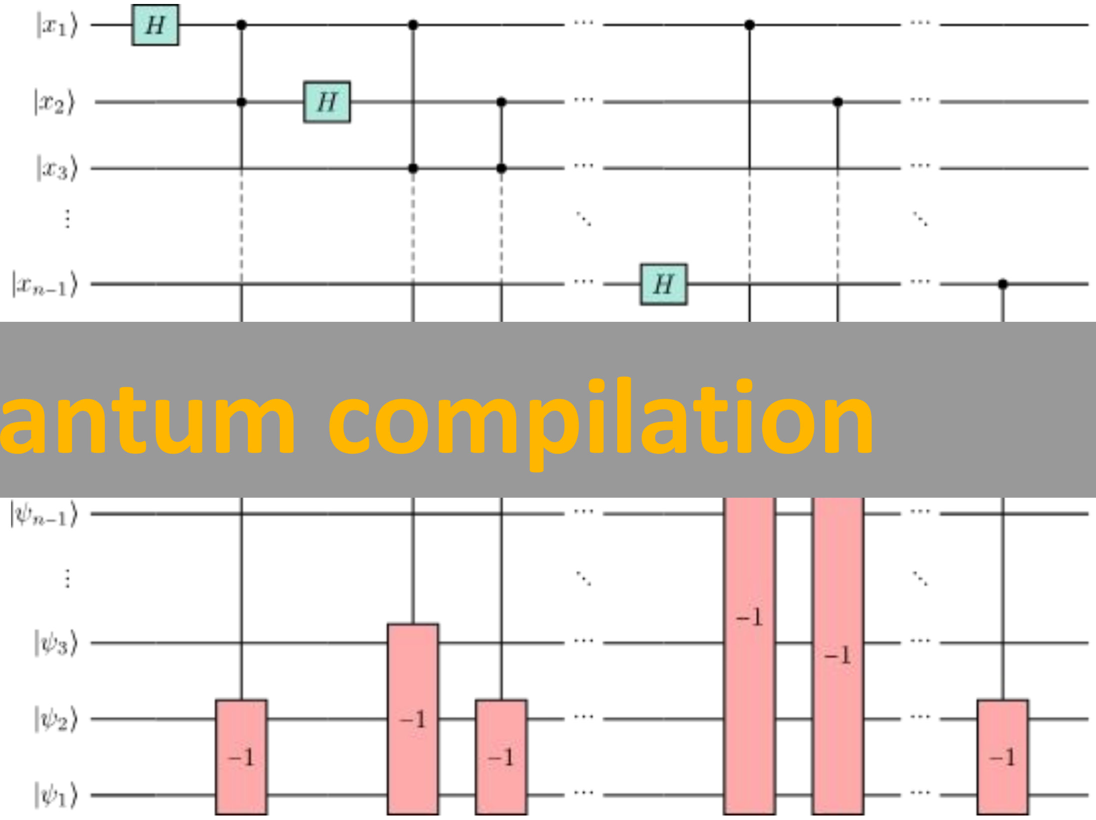
Quantum Physics

[Submitted on 14 Mar 2016 (v1), last revised 14 Oct 2016 (this version, v2)]

An efficient magic state approach to small angle rotations

Earl T. Campbell, Joe O'Gorman

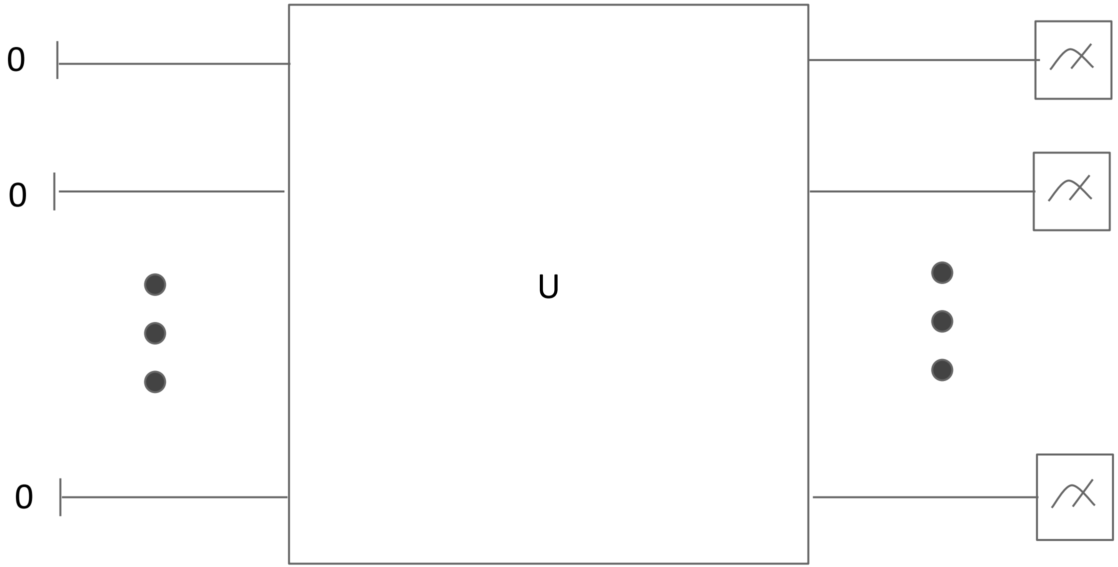
Quantum compilation



Compilation constraints & needs

- So far we've been looking at the physical capabilities of QC
  - What we can do physically
  - What we can do logically
  - Constraints (connectivity, exact vs approximate) and relative costs (fidelity, MSD)
- Next we'll consider the algorithmic needs and meet-in-the-middle!
  - What algorithms exist
  - How we can compile them
  - What their computational bottlenecks are

Recall: the quantum circuit model

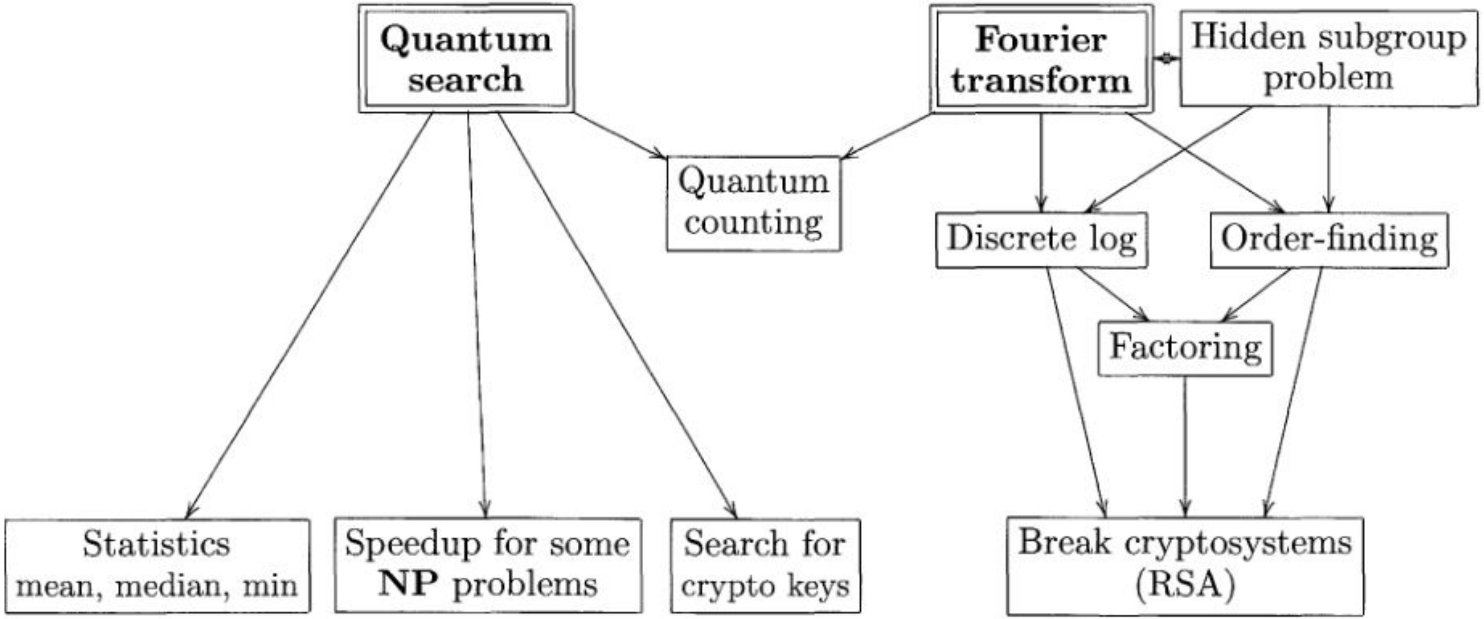


Quantum algorithms

- Not really a sensible algorithmic model

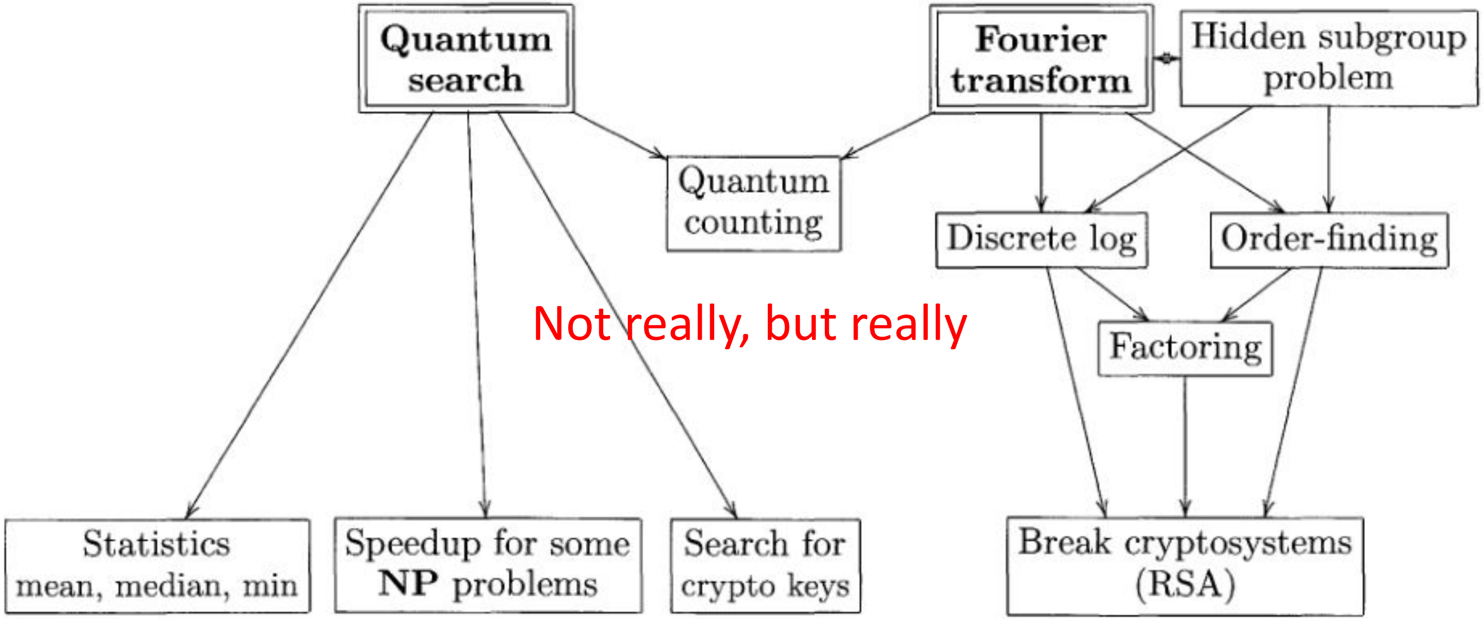
- An algorithm shows that the gate is implementable over a particular gate set with a certain complexity
  - In particular, already have a decomposition/proof of an efficient decomposition
  - **Most unitaries are not efficiently implementable over CNOT+U(2)**
    - Classical: $O(2^n/n)$ (Shannon 1949)
    - Quantum: $O(4^n)$ (Shende, Markov & Bullock 2004)

Quantum algorithms circa 2000



Nielsen & Chuang, QCQI

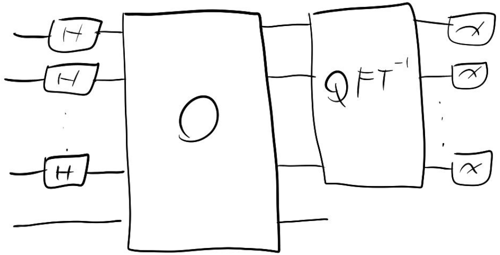
Quantum algorithms circa 2020



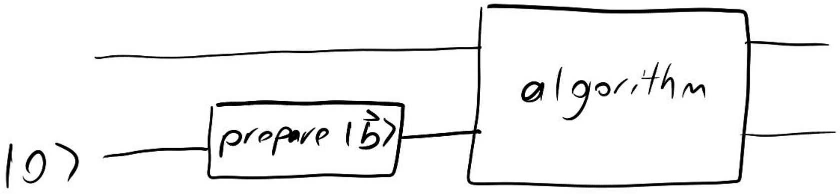
Nielsen & Chuang, QCQI

Quantum algorithms

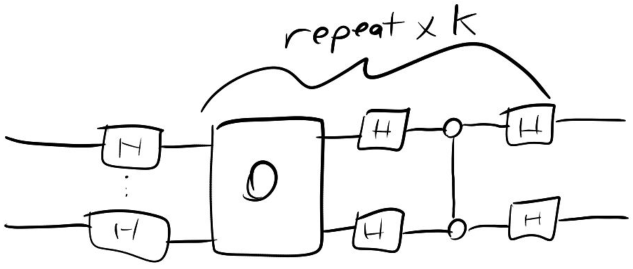
- Fourier-transform based:



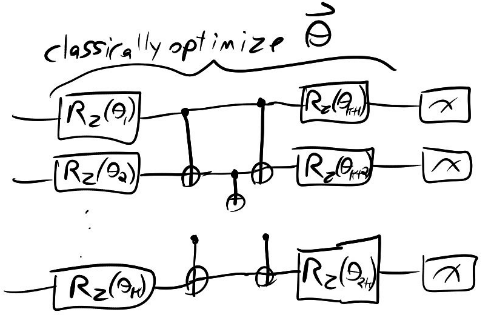
- QRAM based:



- Search-based:



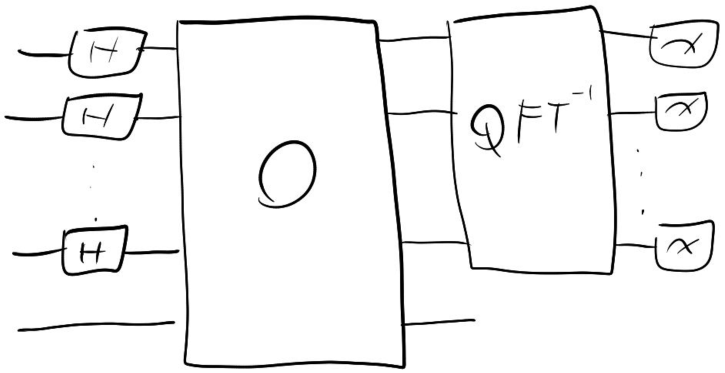
- NISQ/hybrid algorithms:



QFT-based algorithms

Incl. Shor, Dlog, ground-state estimation, linear systems, etc.

1. Create superposition
2. Apply oracle O
3. Apply QFT
4. Measure



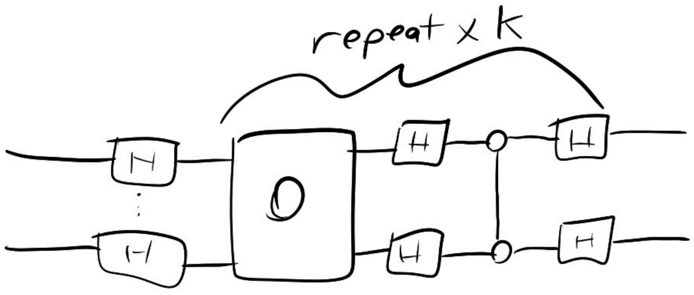
Compilation problems:

- Implementing the QFT
- Implementing the oracle

Search-based algorithms

Incl. grover, optimization, amplitude amplification (used in linear systems), etc.

1. While i < k
   a. Apply oracle O
   b. Apply diffusion operator



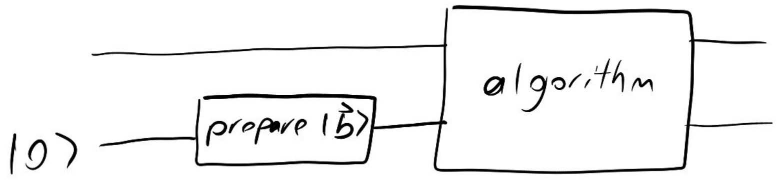
Compilation problems:

- Implementing the oracle

QRAM-based algorithms

Really, specific to linear systems

1. Given classical data vector b,
   prepare quantum state (qram) |b>
2. Apply algorithm with initial state |b>



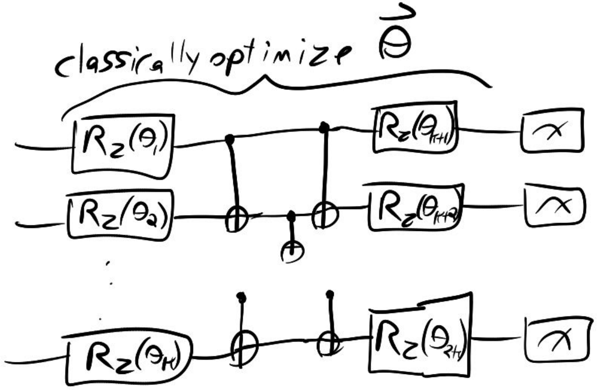
Compilation problems:

- Implementing the state preparation circuit
- Whatever the algorithm is (often Fourier-based)

Hybrid algorithms

VQE, QAOA

1. Select initial parameter vector theta
2. Compute expectation value of some observable on U(theta)
3. While expectation value is not minimized
   a. Modify parameters theta
   b. Go to step 2



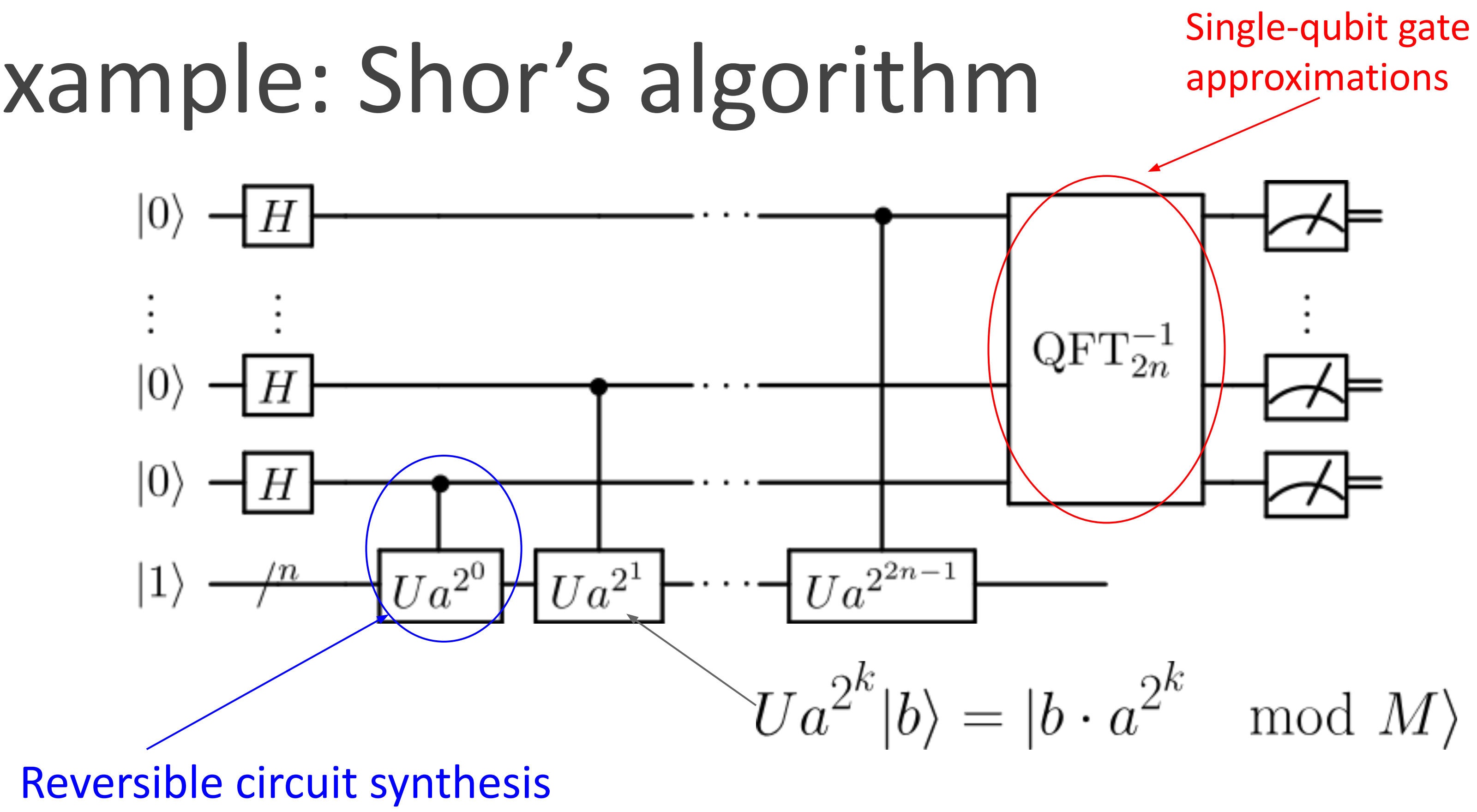
Compilation problems:

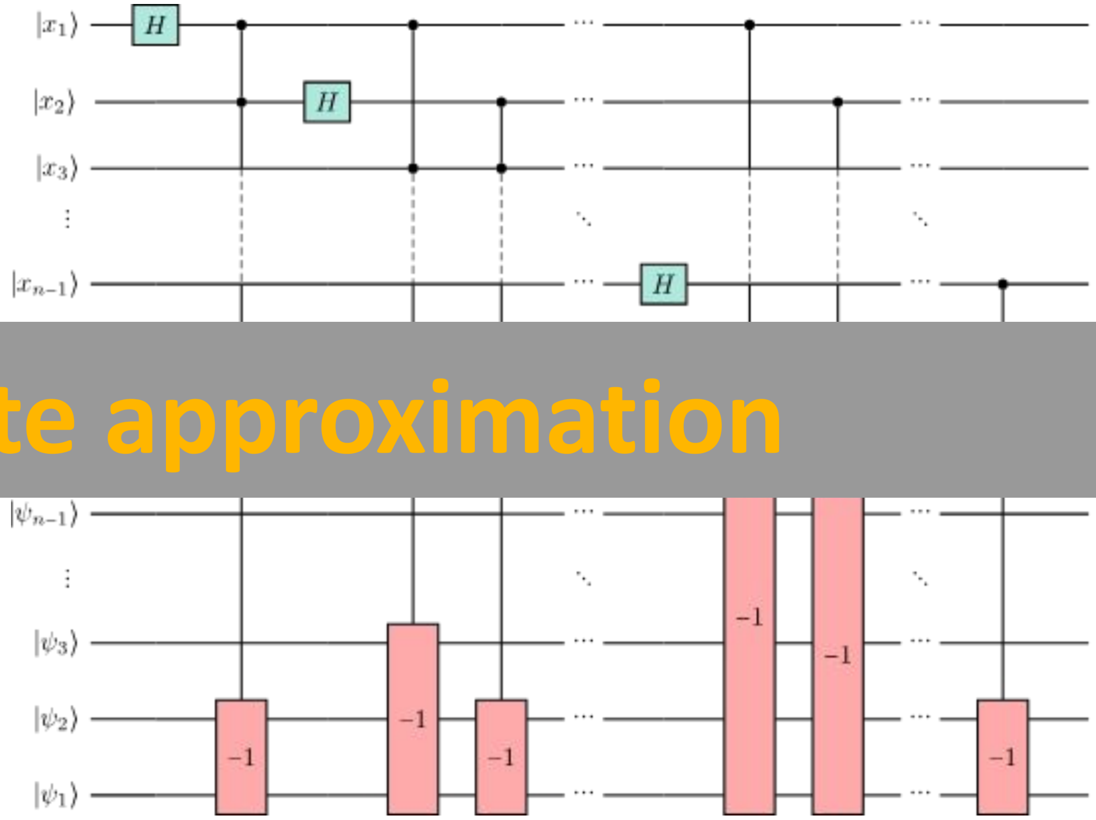
- Compiling a template (ansatz) circuit U for a given problem
- Route to hardware topology
- Minimizing runs to compute the expectation

What’s in the box (oracle O)?

- **Classical functions** (f: $\{0,1\}^n \rightarrow \{0,1\}^m$)
  - Arithmetic (Shor, DLog, HSP)
  - Cryptographic functions (Hash inversion)
  - Graph algorithms (quantum walks)
  - Search problems (e.g. SAT)
  - Optimization problems (NP-complete optimization problems)
- **Time-evolution operators** (e^{iHt} for Hermitian matrix H)
  - Fermionic hamiltonians (Quantum chemistry)
  - k-local hamiltonians (After Jordan-Wigner, physics problems)
  - General linear systems (for HHL)
  - Ansatz for variational (NISQ) algorithms

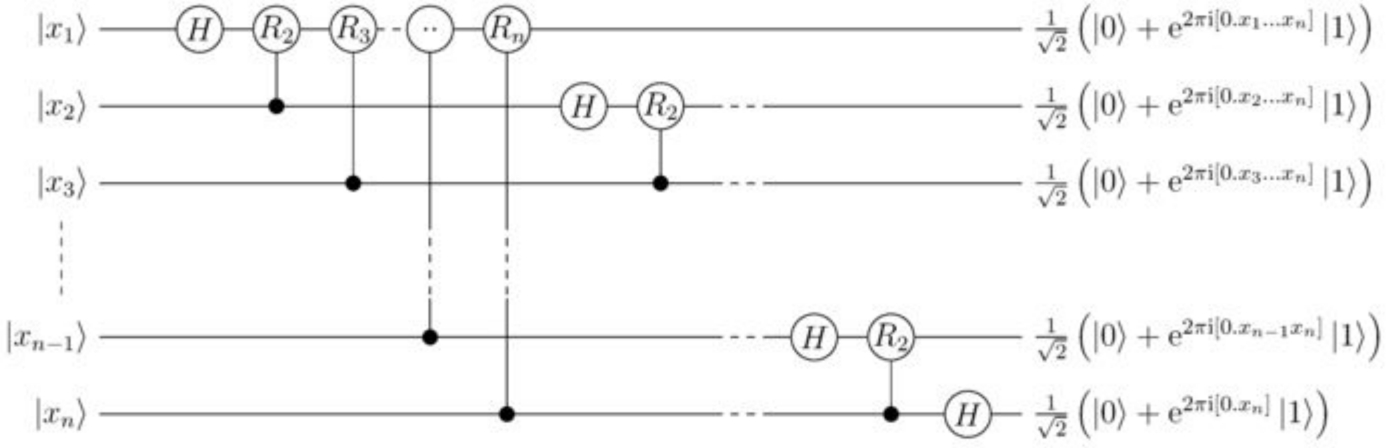
Example: Shor's algorithm





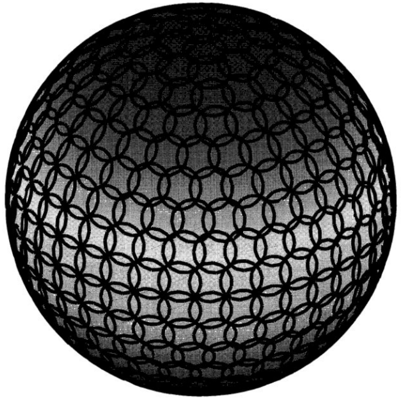
Gate approximation

The Quantum Fourier Transform



- $O(n^2)$ controlled-phase rotations
  - ⇒ each implemented using 3 single-qubit phase rotations
- k >= 3 ⇒ needs approximation (over Clifford+T)!

Single-qubit approximation



- Historically, based on Solovay-Kitaev algorithm
  - $O(\log^c(1/e))$ where $c \approx 4$
  - Idea is that approximating **group commutators** $UVU^\dagger V^\dagger$ centered on a point offers additional error suppression
  - Information theoretic lower bound was $O(\log(1/e))$ so people wondered…

- Solved in 2012 via the number-theoretic method, combining 2 parts
  - An optimal algorithm for synthesizing U(2, Z[1/sqrt{2}, i]) over Clifford+T
  - An algorithm for **rounding off** U(2) in U(2, Z[1/sqrt{2}, i]) with asymptotically optimal cost
  - Overall gate count is $3\log(1/e) + O(\log\log(1/e))$ **for Z-axis rotations**

Algebraic number rings

Recall: a ring R=(S,+,•) is a set S equipped with binary operators +, • such that

- (S,+) is a group (every element has an additive inverse)
- (S,•) is a monoid (multiplication is associative with an identity
- • distributes over +

A ring extension R[a] is (roughly, if a is algebraic) “R-valued polynomials in a”

E.g., $R_0 + r_1 a + r_2 a^2 + r_3 a^3 + \ldots r_k a^k$

D[ω]

Ring of dyadic fractions:

$$D = \{a/2^b \mid a, b \text{ are integers}\}$$

D[ω] = Z[1/sqrt{2}, i] is obtained by adjoining an 8th root of unity to D:

$$D[\omega] = \{a + b\omega + c\omega^2 + d\omega^3 \mid a,b,c,d \text{ dyadic fractions}\}$$

The least denominator exponent (**lde**) of r in D[ω] is the smallest b such that

$$r*\text{sqrt}\{2\}^b = a + b\omega + c\omega^2 + d\omega^3 \mid a,b,c,d \text{ are integers}$$

LDE-based Exact synthesis

- (Kliuchnikov, Maslov, Mosca, 2013) U(2,D[ω]) = <H, T>
- (Giles & Selinger, 2013) U(n,D[ω]) = <H, CNOT, T>
- Proof in either case is by giving an **exact synthesis algorithm**

LDE-based exact synthesis:

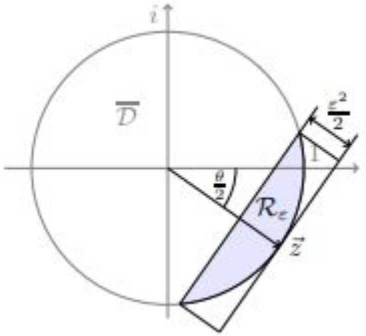
1. Given an n by n unitary $U = [u_1 \ u_2 \ \ldots \ u_n]$
2. For i from 1 to n
   a. While $lde(u_i) > 0$
      i. Pick two rows of u_i with maximal LDE
      ii. Apply HT^k on those rows to reduce their LDE

Important: T-optimal for 1 qubit

Example

Ring round-off

- A few different methods…
- Ross & Selinger's 2016 grid-synth algorithm gets (almost) optimal T-counts
- Rough sketch
  - Enumerate points within a region of the unit circle in order of increasing LDE
  - Given such a point u1, find a point u2 that gives a unit vector $[u1\ u2]^T$
    - Requires solving a diophantine equation…
    - …But in practice get second from optimal efficiently



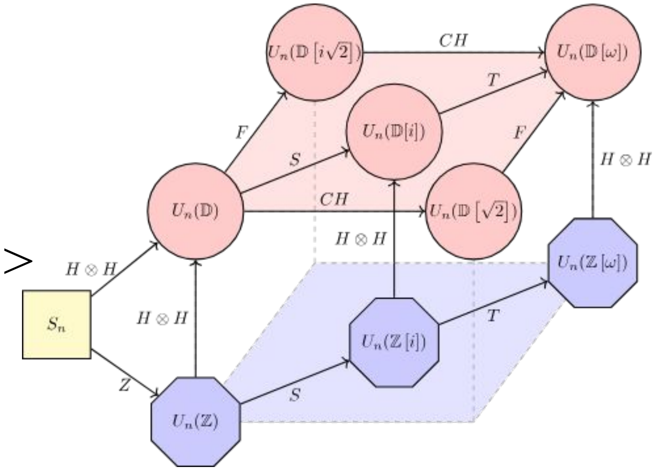
Open questions for gate approximation

- Optimal approximation of **non-Z-axis** rotations
- Optimal approximation & synthesis over <H, diag(1, exp(i pi / 2^k))>
- Trade-offs with probabilistic techniques
  - Repeat-until-success circuits known which approximate with expected T-count just log(1/e)
- Trade-offs with (cascading) gate teleportation
  - If can get 50% shorter sequences with sqrt{T}, is it worth the cost of cascaded teleportation?
- Gate sets which fill up the Bloch-sphere most efficiently
  - Called “golden gates”
  - Number-theorists have at least partially solved this
- **For which gate sets do there exist similar characterizations?**

Number-theoretic characterizations

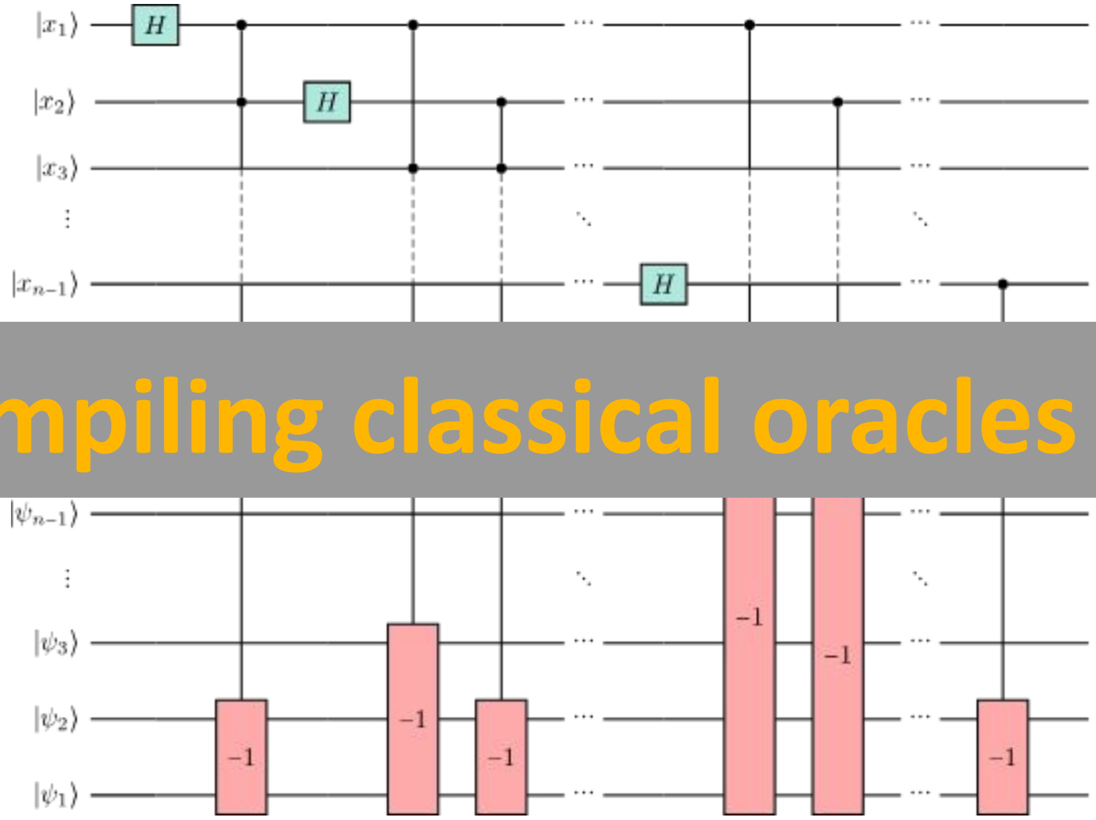
A number of other such exact characterizations exist:

- U(n,D) = < Toffoli, H⊗H >
- U(n, D[sqrt{2}]) = < Toffoli, H, CH >
- U(n, D[i]) = < Toffoli, ωH, S >
- U(n, D[sqrt{-2}] = < Toffoli, ∝sqrt{H} >
- U(n, D[ω]) = < CNOT, H, T >
- U(n, D[exp(2pi/2^k)]) = < CNOT, H, R_k >

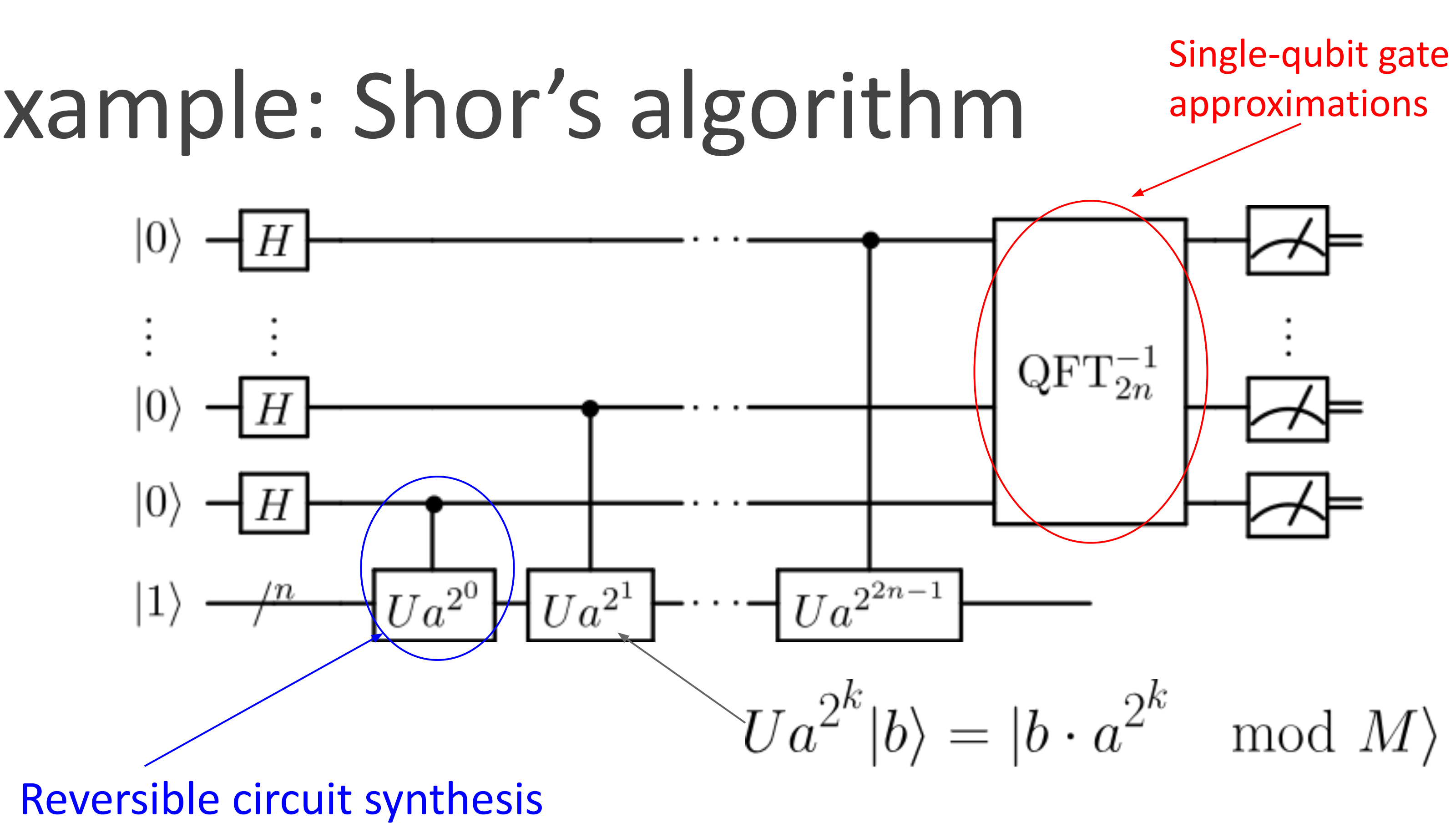


Philosophical implications: **domains for quantum computing**

Compiling classical oracles



Example: Shor's algorithm



Classical logic synthesis

Problem:

Given a classical function/code f: $\{0,1\}^n \rightarrow \{0,1\}^m$ implement the oracle

U_f: |x>|0> → |x>|f(x)> (out of place)

U_f: |x> → |f(x)> (in place)

Compilation flow:

1. Start from an irreversible, bit-wise algorithm (e.g. binary addition)
2. Make reversible by adding temporary values & uncomputations
3. Expand to Clifford+T (or other gate set)

Example: binary addition

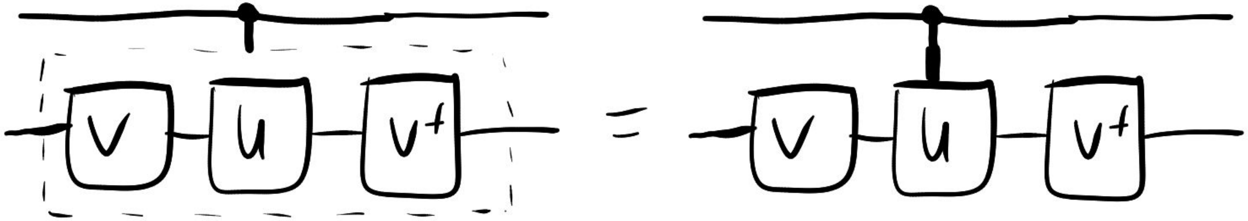
The dark art of quantum circuit synthesis

- Getting efficient (**time & space**) circuits in the end is about
  - knowing context
    - Is it being controlled?
    - Is it in a larger computation which can re-use resources?
  - plus a big ole' bag of tricks, like
    - Palindromes
    - Pebble games
    - Dirty/borrowed ancillas
    - Phase polynomials
    - Relative phases
    - Measurement assisted uncomputation

Example: controlling a sub-circuit

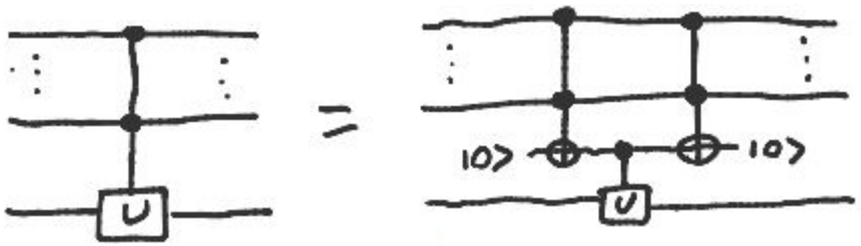
- Easy!

- Palindromes ($V^\dagger UV$) only require a single control:



Multiply-controlled gates

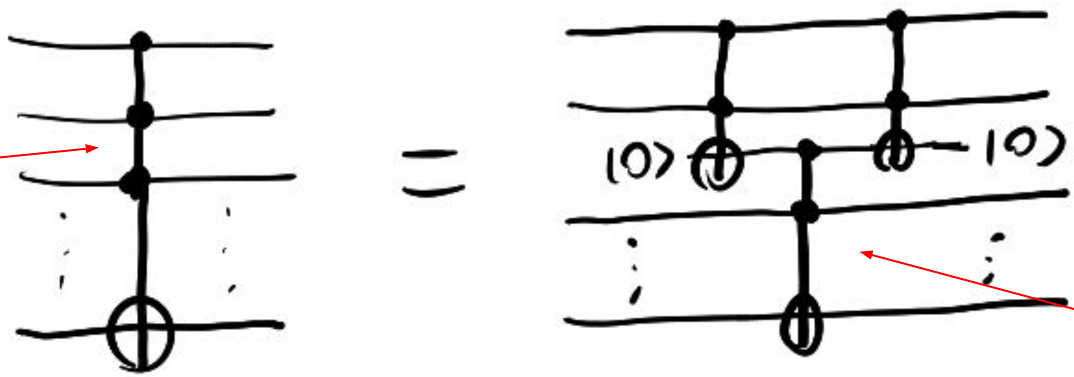
- Problem : expanding out controls may result in multiply-controlled gates
- Can use multiply-controlled Toffoli gates to reduce down to a single control:



- Compute-control-uncompute pattern is highly optimizable at a quantum level

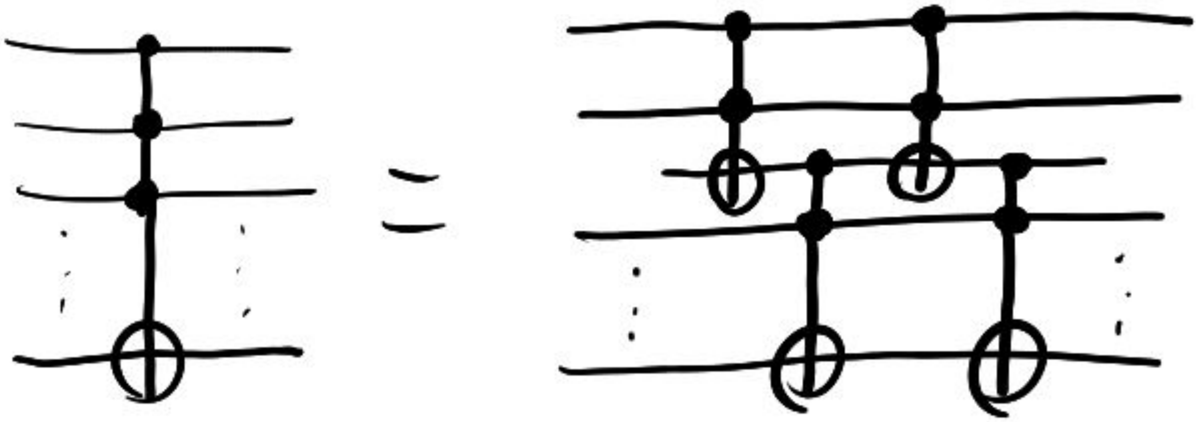
Multiply-controlled Toffoli gates

- Bread-and-butter of reversible computation (and compilation)
- Implement k-ary Boolean products
- Much work has gone into optimizing these gates (+ proving lower bounds) using 2-control Toffoli gates



Dirty/borrowed ancillas

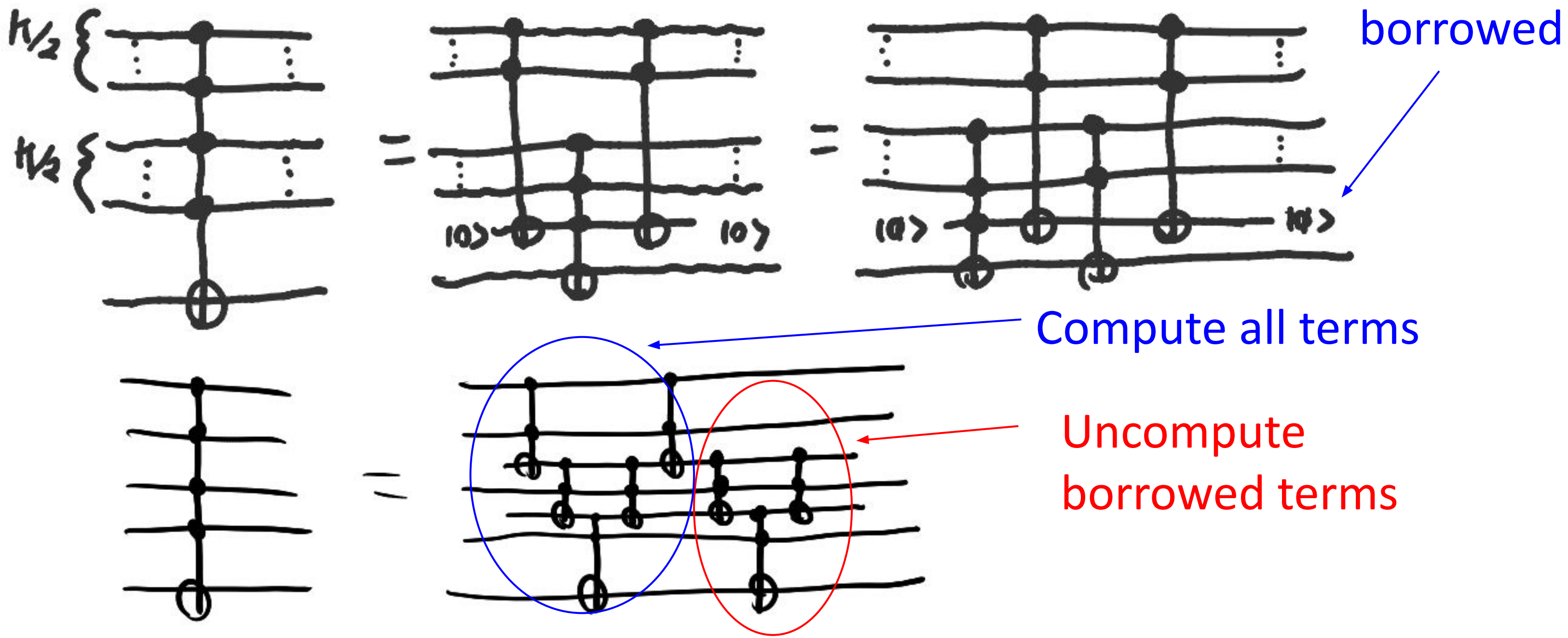
- Previous construction used a linear number of ancillas
- Can get it down to 1 by temporarily borrowing other active qubits



Problem: exponential gate count!

Linear complexity MCT with 1 ancilla

- Solution is to split controls in half & use k/2 (dirty) ancillas



How many ancillas do we actually need?

Theorem:

Any reversible function on n >= 4 bits requires at most one ancilla to implement over {X, CNOT, Toffoli}, and no ancillas if it has determinant 1

Proof idea:

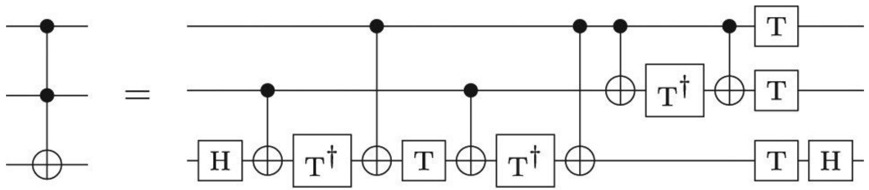
- Reversible n-bit functions are permutations on {0,...,2^n-1}
- A permutation is even iff det(P) = 1, and -1 otherwise
- {X, CNOT, Toffoli} all have determinant 1 on n >= 4 bits
- An odd permutation on n bits can be embedded as an even permutation on n+1 bits
- Even permutations can be implemented without ancillas (Shende, Prasad, Markov, Hayes 2003)

Interaction with fault tolerance

- Recall: it's more efficient in practice to use Clifford+T
- Typical compilation goes

  Classical function → Reversible embedding → Toffoli gates → Clifford+T

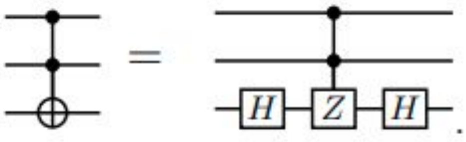
  using 7 T gates per Toffoli:



Let's dig into this because it will tell us a lot about reversible computations in Clifford+T

The Toffoli gate

- Toffoli gate is equivalent to a doubly-controlled Z up to Cliffords:



- Doubly-controlled Z implements |x,y,z> → $(-1)^{xyz}$|x,y,z>
- Clifford+T implementation arises through the Fourier expansion of xyz

$$
\begin{aligned}
2xy &= x + y - (x \oplus y) \\
4xyz &= 2x(y + z - (y \oplus z)) \\
&= x + y + z - (x \oplus y) - (x \oplus z) - (y \oplus z) + (x \oplus y \oplus z)
\end{aligned}
$$

The CCZ gate

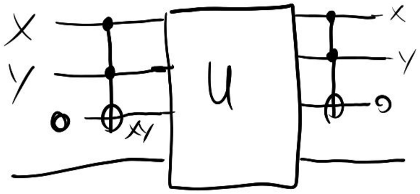
"Phase polynomial"

Goal is to implement

$$CCZ : |x, y, z\rangle \mapsto \omega^{x+y+z-(x\oplus y)-(x\oplus z)-(y\oplus z)+(x\oplus y\oplus z)}|x, y, z\rangle$$

Relative phase

- Common pattern is to compute & uncompute a binary product

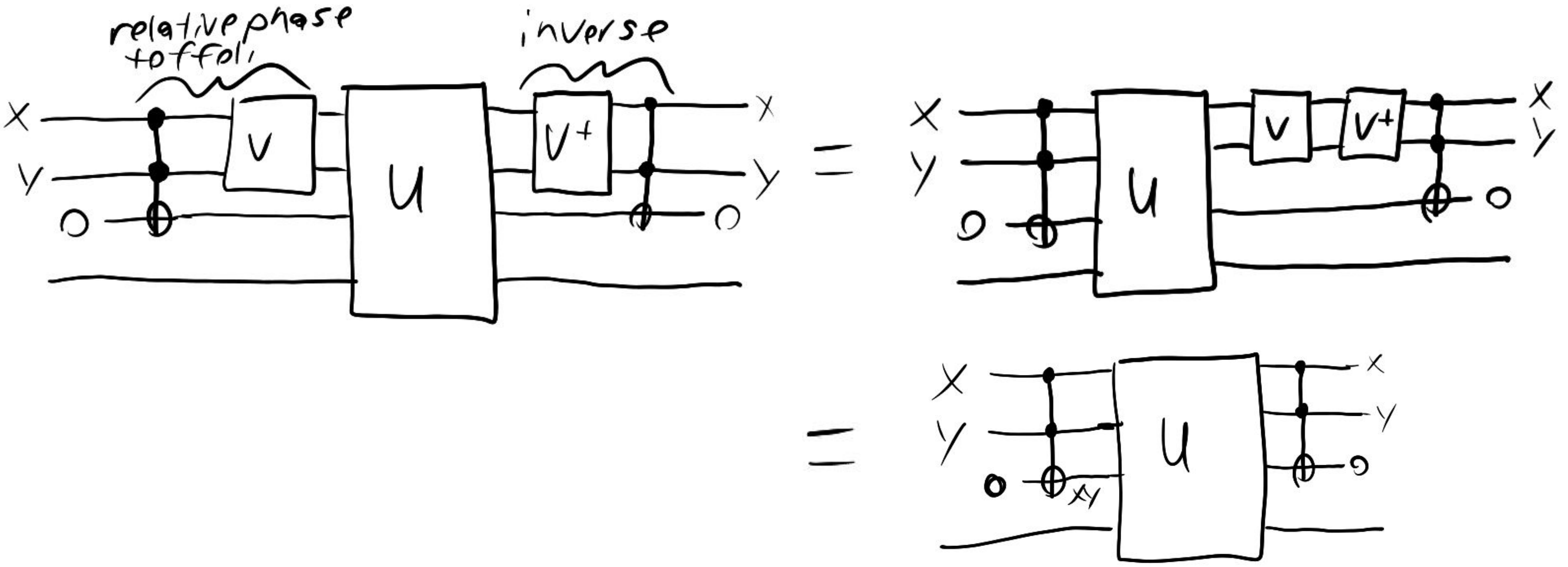


- 3 terms of the CCZ gate phase polynomial only involve controls, so they can be factored out & cancelled with the uncomputation

$$CCZ : |x, y, z\rangle \mapsto \omega^{x+y+z-(x\oplus y)-(x\oplus z)-(y\oplus z)+(x\oplus y\oplus z)}|x, y, z\rangle$$

Selinger, quantum circuits of T-depth 1, 2012

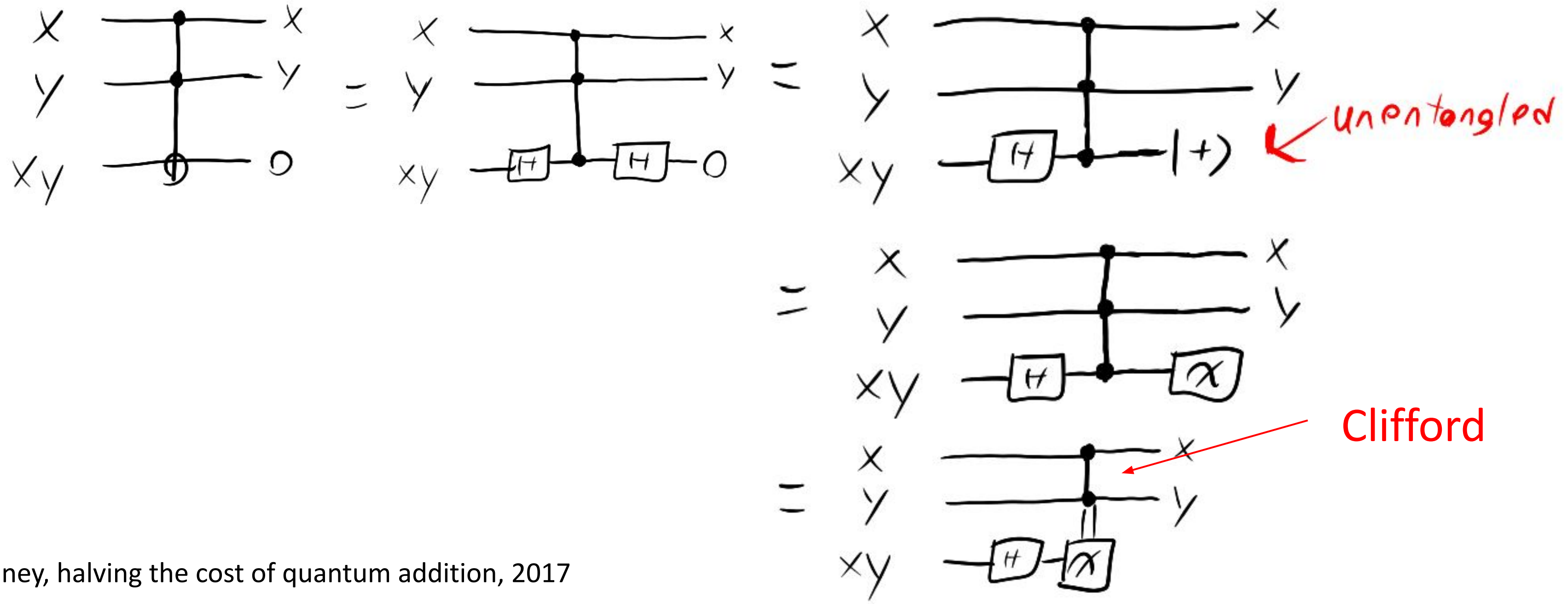
Relative phase



Selinger, quantum circuits of T-depth 1, 2012

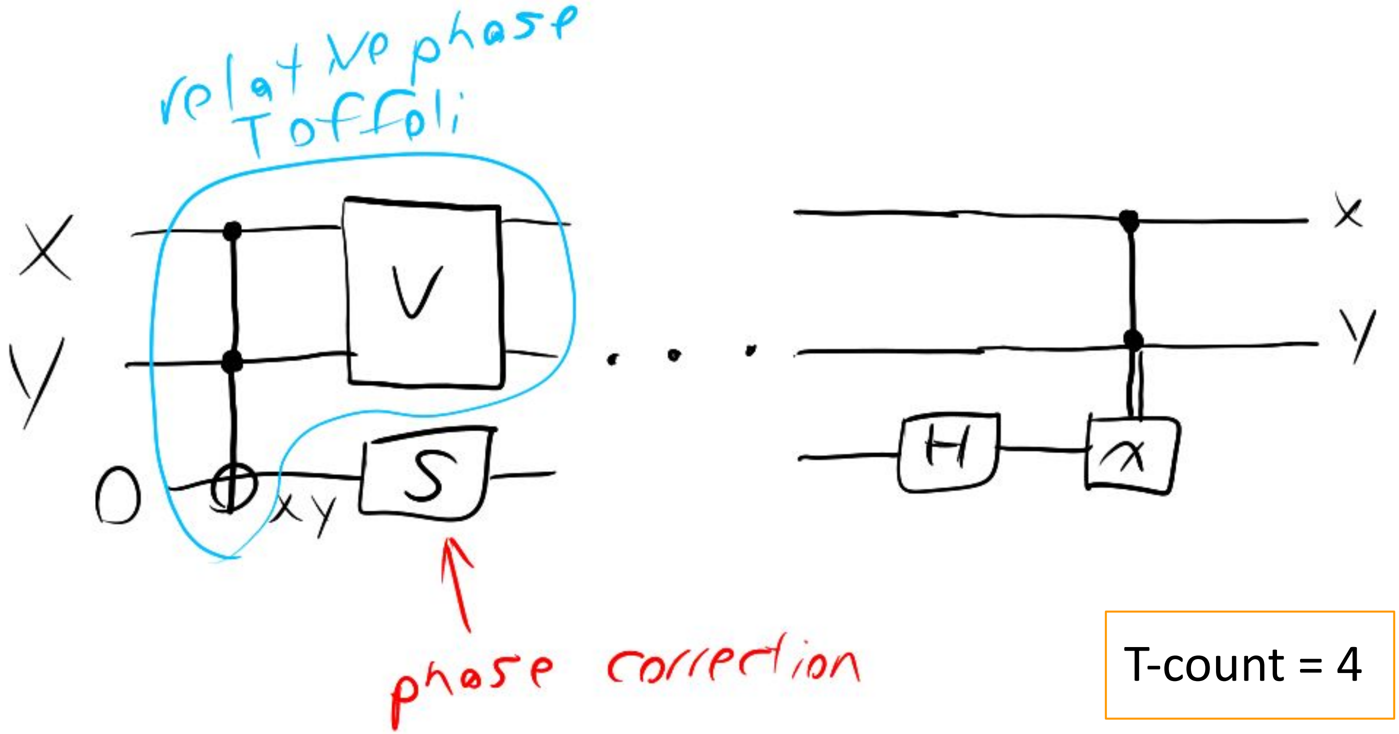
Measurement-assisted uncomputation

- Can do even better on the right-hand side if we can discard the final state



Gidney, halving the cost of quantum addition, 2017

Efficient temporary AND



T-count = 4

Upshot

- In the limit of many ancillas, reversible computation takes
  - 2(k-2)+1 Toffolis per k-control Toffoli
  - ~4 T gates per Toffoli
- Still need reasonable decompositions into few Toffolis/MCT
  - E.g. $O(n)$ algorithm vs $O(n^2)$ algorithm at the Toffoli level
- Open questions:
  - How to get the space usage down?
  - How to generalize relative phase synthesis?
  - How to generalize measurement-based techniques?
  - Tight lower bounds for the T-count of reversible computations?
    - Best known lower bounds give k T gates for a k-1 control Toffoli

Beverland et al., Lower bounds on the non-Clifford resources for quantum computations, 2019

Readings for next week

- Posted to the website
  - Patel, Markov, Hayes, *Efficient Synthesis of Linear Reversible Circuits*. arXiv:quant-ph/0302002
  - Meuli, Soeken, Roetteler, Bjorner, de Micheli, *Reversible Pebbling Game for Quantum Memory Management*. arXiv:1904.02121
  - Amy, Ross, *The phase/state duality in reversible circuit design*. arXiv:2105.13410
  - Khattar, Gidney, *Rise of conditionally clean ancillae for optimizing quantum circuits*. arXiv:2407.17966
- Send me a short (paragraph or two) summary of **ONE (1)** paper of your choice before next class
- Be prepared to give a quick (up to 5 minutes) summary of any of the readings. I’ll ask for a volunteer to summarize and kick off the discussion for each paper