The phase-state duality in reversible circuit design

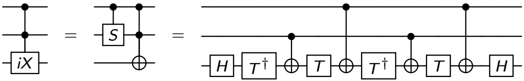
Matthew Amy & Neil J. Ross

Dalhousie University, Halifax, Canada

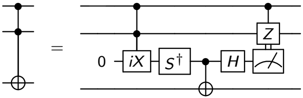
International Workshop on Quantum Compilation,
Cambridge Quantum Computing

The reversible circuit construction zoo

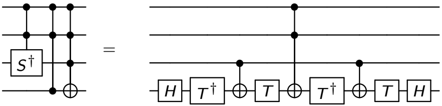
- ancilla-free multiply-controlled *iX* gates [Sel13, GS13]



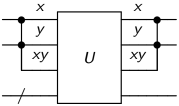
- *T*-count 4 measurement-assisted Toffoli [Jones13]



- ancilla-free, *T*-count 8 relative-phase Toffoli-4 [Mas16]



- *T*-count 4 temporary logical-AND [Gid18]



There's got to be a better way!

Goal:

Generalize these constructions and unify them within a framework of **reusable**, **automatable** *design techniques.*

It's a process. It's a process. It's a process.

Not all the way there yet...

It's a process. It's a process. It's a process.

Not all the way there yet...

...but we have applications!

- one dirty ancilla, T-count $8(k-1)-4$ relative phase Toffoli
- ancilla-free, T-count $4(k-1)$ temporary degree k functions
- ancilla-free, T-count $8(k-2)$ temporary logical-k-AND
- measurement-assisted uncomputation for 3-, 4-, and 5-ANDs

Generally useful for **low-space** implementations

Case study I: dirty ancillas

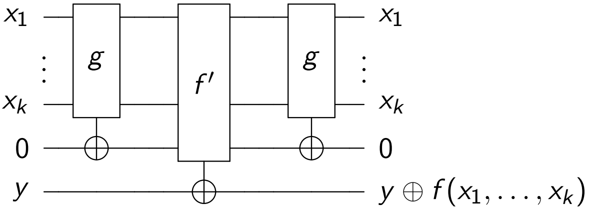
or, Building a Better Barenco

Implementing a classical oracle

Given a Boolean function $f : \mathbb{Z}_2^k \to \mathbb{Z}$, we want to implement the reversible map

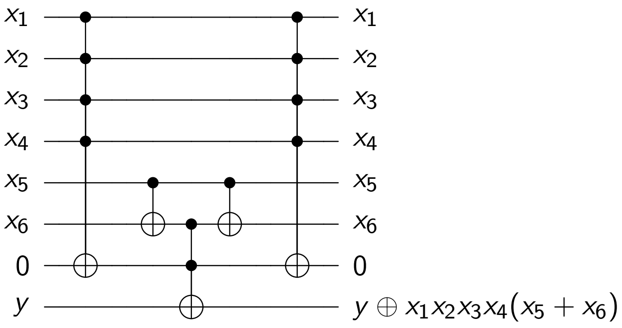
$$|x_1 \cdots x_k\rangle\,|y\rangle \mapsto |x_1 \cdots x_k\rangle\,|y \oplus f(x_1, \ldots, x_k)\rangle$$

Typical solution is to compute **temporary** values into ancillas, then later **uncompute**



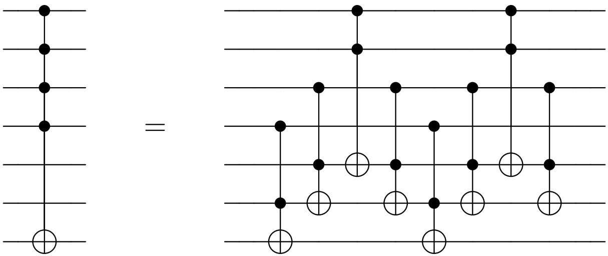
A concrete example

Suppose $f(x_1, \ldots, x_6) = x_1x_2x_3x_4x_5 + x_1x_2x_3x_4x_6$. With one clean ancilla we can factor as $x_1x_2x_3x_4(x_5 + x_6)$ and implement as below:



A canonical implementation

First attempt [BBC+95]:

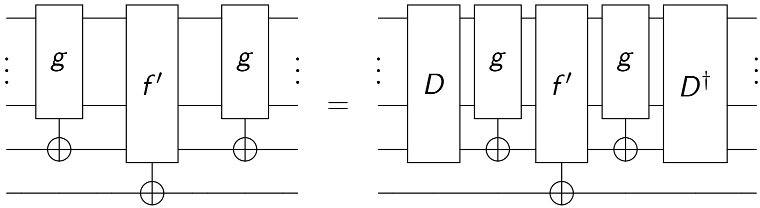


- T-count 24 ($8(k-1)$ for k controls) [Mas16]
- ancilla count $\lceil \frac{k-2}{2} \rceil$ for k controls [Mas16]

Can we do better?

Relative phase

For any diagonal gate D, we have the following equivalence

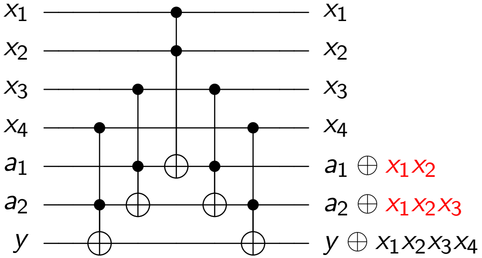


Designing up to phase

Recall that conjugation by H gates swaps state and phase

$$(-1)^{yg(x_1,\ldots,x_n)} |y\rangle\langle y| \overset{H(\cdot)}{\longleftrightarrow} |y \oplus g(x_1,\ldots,x_n)\rangle\langle y|$$

We can use this to turn **state garbage** which would otherwise require uncomputation into irrelevant **phase garbage**

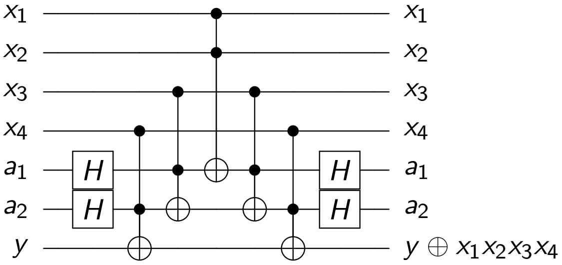


Designing up to phase

Recall that conjugation by H gates swaps state and phase

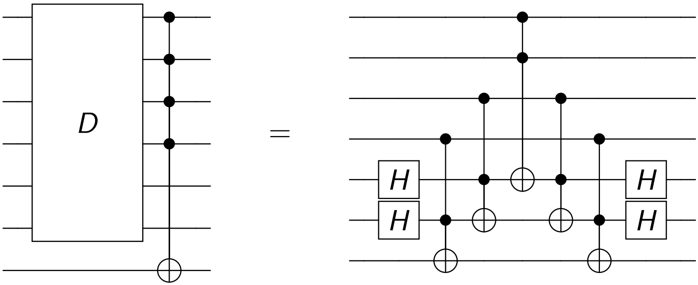
$$(-1)^{yg(x_1,\ldots,x_n)} |y\rangle \langle y| \xleftrightarrow{H(\cdot)} |y \oplus g(x_1,\ldots,x_n)\rangle \langle y|$$

We can use this to turn **state garbage** which would otherwise require uncomputation into irrelevant **phase garbage**



A slight improvement

Second attempt:



- T-count 20 ($8(k-1)-4$ for k controls)
- **matches the usual clean ancilla construction**

Can we do even better?

Complete re-synthesis

To implement the Toffoli-5 directly **up to phase**, we only need to compute some phase

$$(-1)^{yx_1x_2x_3x_4}e^{i\theta(x_1,\ldots,x_4)}$$

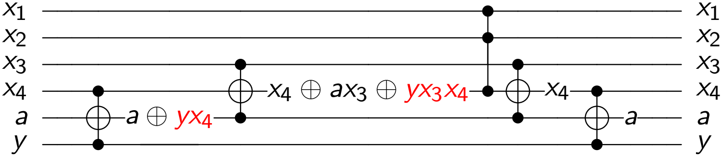
and then swap the phase and state space

Complete re-synthesis

To implement the Toffoli-5 directly **up to phase**, we only need to compute some phase

$$(-1)^{yx_1x_2x_3x_4}e^{i\theta(x_1,\ldots,x_4)}$$

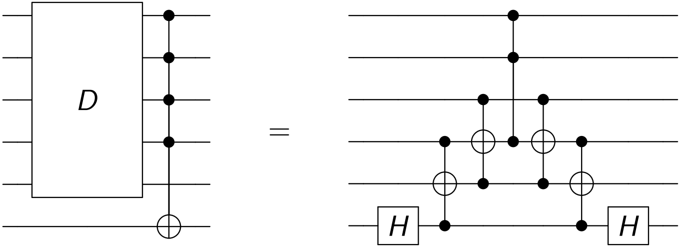
and then swap the phase and state space



Phase: $(-1)^{x_1x_2x_4+ax_1x_2x_3+yx_1x_2x_3x_4}$

Construction I

Single dirty ancilla Toffoli up to phase



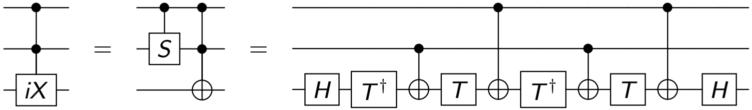
- T-count 20 ($8(k-1)-4$ for k controls)
- only uses **1 ancilla**

Case study II: ancilla-free constructions

or, The Church of the 4 Phases

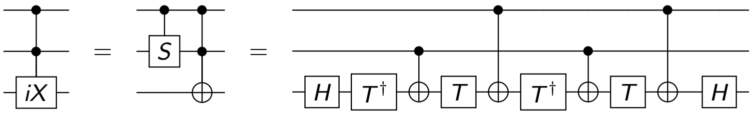
The *cciX* gate

Selinger's *cciX* gate [Sel13]:

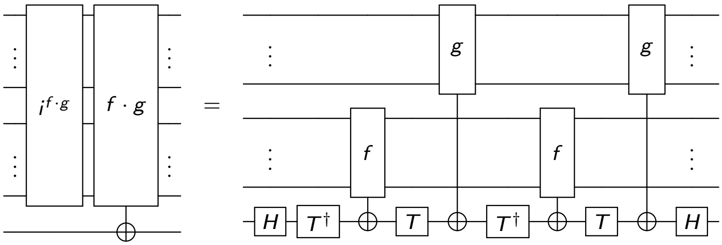


The *cciX* gate

Selinger's *cciX* gate [Sel13]:

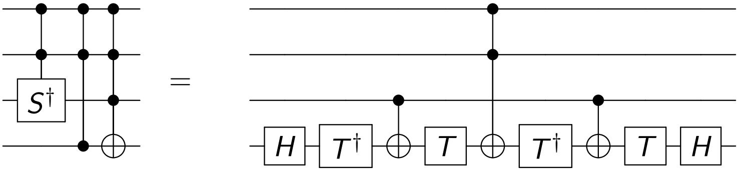


General form (**balanced** multiplication):



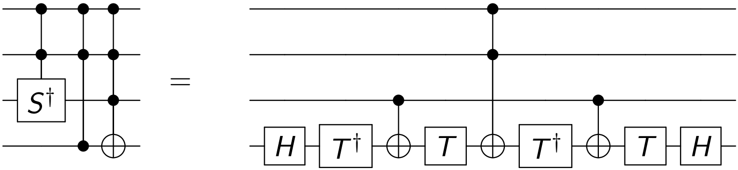
The relative phase Toffoli-4

Maslov's relative phase Toffoli-4 [Mas16]:

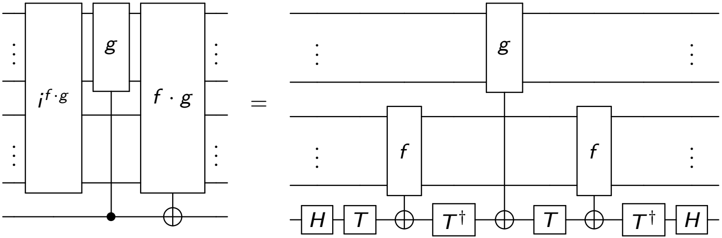


The relative phase Toffoli-4

Maslov's relative phase Toffoli-4 [Mas16]:



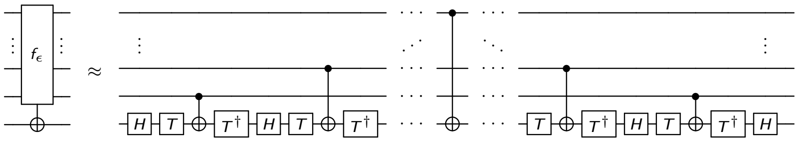
General form (**unbalanced** multiplication):



Construction II

Ancilla-free high-degree functions

Iterating **unbalanced** multiplication with $f(x) = x$ generates **high degree oracles** up to phase

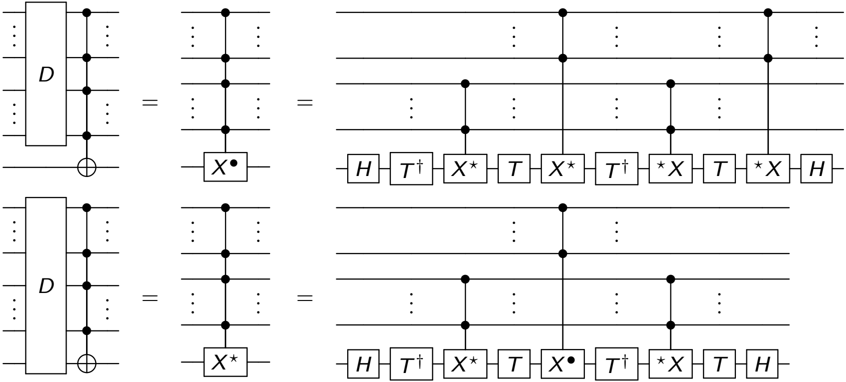


E.g., $f_\epsilon(x_1, x_2, x_3, x_4) = x_1x_2x_3x_4 + x_1x_4 + x_3x_4$

- T-count $4(k-1)$ vs $16(k-1) + O(1)$ [GS13+Mas16]
- can get distinct **Clifford-equivalence classes** of functions by inserting Clifford gates intermittently
- applications to LUT-based synthesis [SRW+18]

Ancilla-free multiply-controlled Toffolis

To compute a simple k-AND, need to eliminate extra terms by interleaving **balanced** multiplication

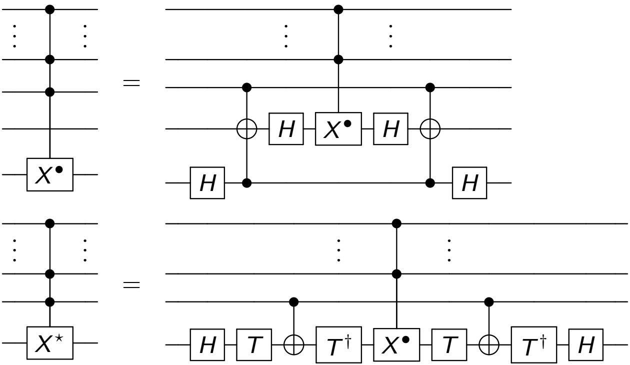


Problem: scales **non-linearly**

Construction III:

Ancilla-free relative phase Toffoli$-k$

Solution: bootstrap with a dirty-ancilla $X^{\bullet}$



- T-count: $8(k-2)$ vs $16(k-2)+4$ [GS13+Mas16]
- further improves T-count for $< \lceil \frac{k-2}{2} \rceil$ clean ancillas

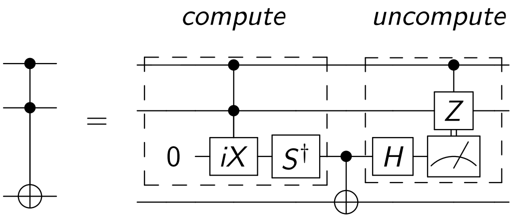
Case study III: measurement-assisted uncomputation

or, Trading Measurement for T

The measurement-assisted Toffoli

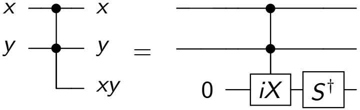
Recall [Jon13]:

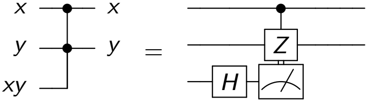
> the Toffoli gate can be implemented with 4 T gates, one measurement and a classically controlled Clifford



Computing & uncomputing a two-bit product

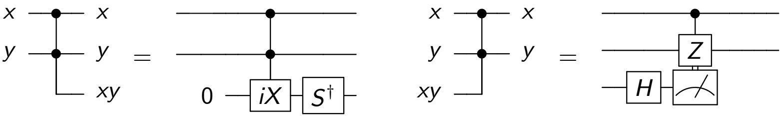
Rather than copy out the result to implement a full Toffoli, use these subcircuits to compute and uncompute products [Gid18]



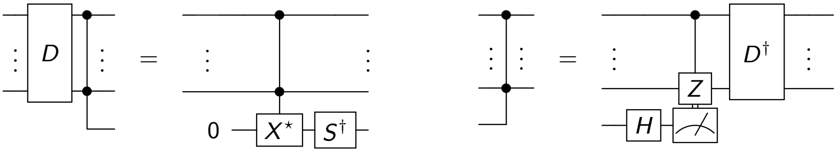


Computing & uncomputing a two-bit product

Rather than copy out the result to implement a full Toffoli, use these subcircuits to compute and uncompute products [Gid18]



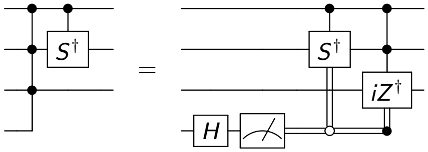
More generally,



Construction IV

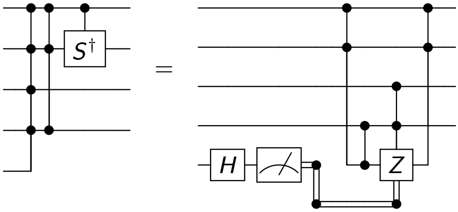
un-logical-AND

un-3-AND:



T-count: 3.5 on average (vs. 8)

un-4-AND:



T-count 7.5 on average (vs. 16)

Conclusion

or, The End

Conclusions & Future Work

In this talk...

- classes of degree k functions with T-count $4(k-1)$
- temporary logical-k-ANDs with T-count down to $8(k-2)$
- some measurement-assisted uncomputation circuits

Main takeaway: improvements can be made by designing with both phase- and state-space in mind

Future work

- measurement-assisted un-k-AND for any k
- automated synthesis
- (user-friendly) computer-aided design tools

References

[BBC+95] A. Barenco, C. H. Bennett, R. Cleve, D. P. DiVincenzo, N. Margolus, P. Shor, T. Sleator, J. A. Smolin, and H. Weinfurter. Elementary gates for quantum computation. Physical Review A, 52:3457–3467, 1995.

[Sel13] P. Selinger. Quantum circuits of T-depth one. Physical Review A, 87:042302, 2013

[GS13] B. Giles and P. Selinger. Exact synthesis of multiqubit Clifford+T circuits. Physical Review A, 87:032332, 2013.

[Jon13] C. Jones. Low-overhead constructions for the fault-tolerant toffoli gate. Physical Review A, 87:022328, 2013.

[Mas16] D. Maslov. Advantages of using Relative-Phase Toffoli Gates with an Application to Multiple Control Toffoli Optimization. Physical Review A, 93:022311, 2016.

[Gid18] C. Gidney. Halving the cost of quantum addition. Quantum, 2:74, 2018.

[SRW+19] M. Soeken, M. Roetteler, N. Wiebe, and G. D. Micheli. LUT-Based Hierarchical Reversible Logic Synthesis. IEEE Trans- actions on Computer-Aided Design of Integrated Circuits and Systems, 38(9):1675–1688, 2019.

Thank you!