Quantum Circuit Optimisation with the ZX-calculus

Ross Duncan	Cambridge Quantum Computing
Simon Perdrix	Université de Lorraine
Aleks Kissinger	Oxford University
John van de Wetering	Radboud University Nijmegen

April 8, 2020

Quantum circuit optimisation

- We want to use quantum resources as efficiently as possible.

Quantum circuit optimisation

- We want to use quantum resources as efficiently as possible.
- So quantum circuits should contain as few gates as possible.

Quantum circuit optimisation

- We want to use quantum resources as efficiently as possible.
- So quantum circuits should contain as few gates as possible.
- Several important metrics:

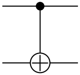
Quantum circuit optimisation

- We want to use quantum resources as efficiently as possible.
- So quantum circuits should contain as few gates as possible.
- Several important metrics:
  - Gate-depth
  - 2-qubit gate count

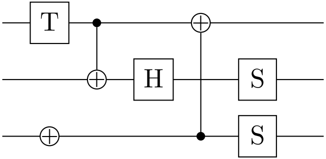
Quantum circuit optimisation

- We want to use quantum resources as efficiently as possible.
- So quantum circuits should contain as few gates as possible.
- Several important metrics:
  - Gate-depth
  - 2-qubit gate count
  - Number of T gates: *T-count* $[\ \mathrm{T} = R_Z(\frac{\pi}{4})\ ]$

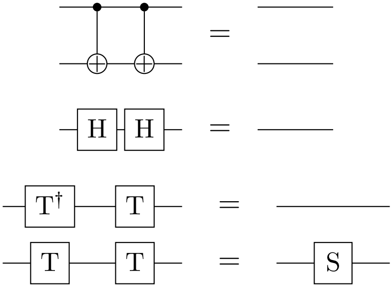
Circuit diagrams

NOT $=$  CNOT $=$ 

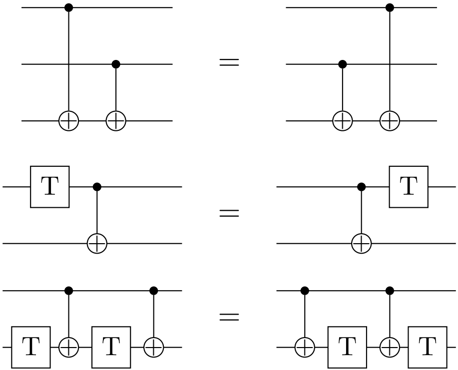
An example quantum circuit:



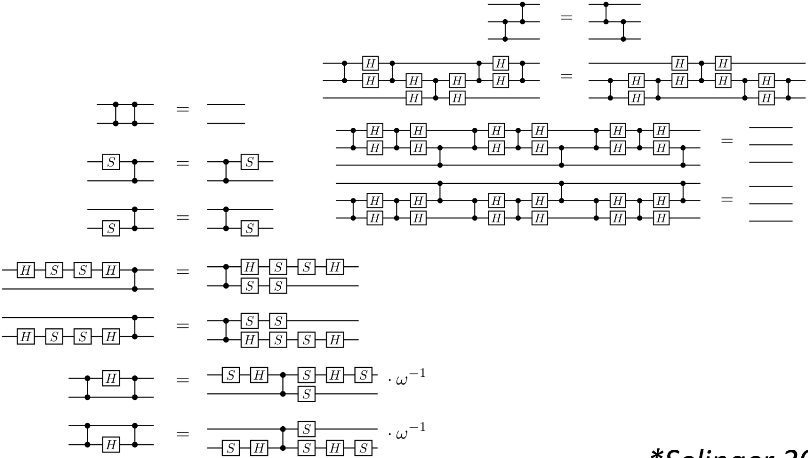
Circuit identities



Gate commutation

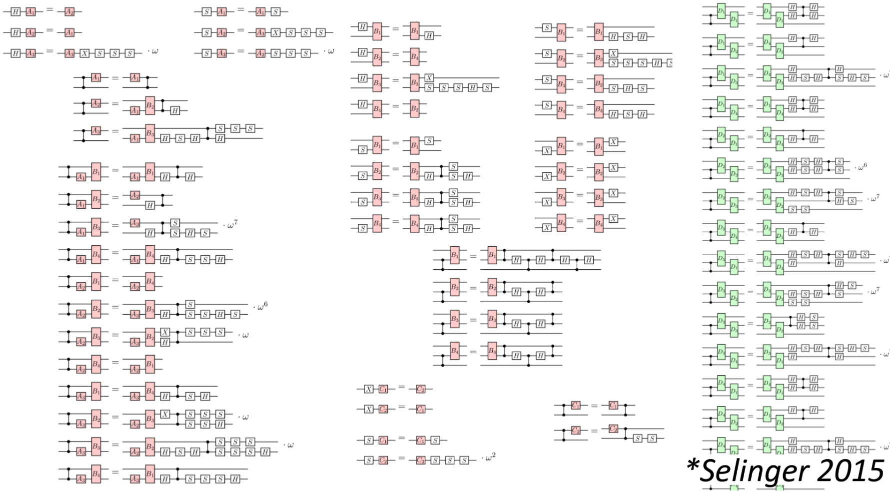


More circuit equalities



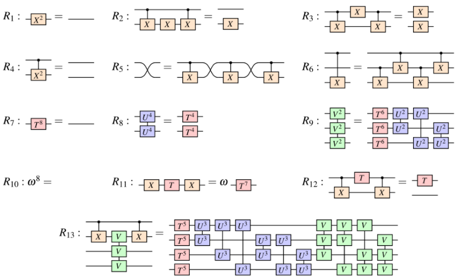
*Selinger 2015

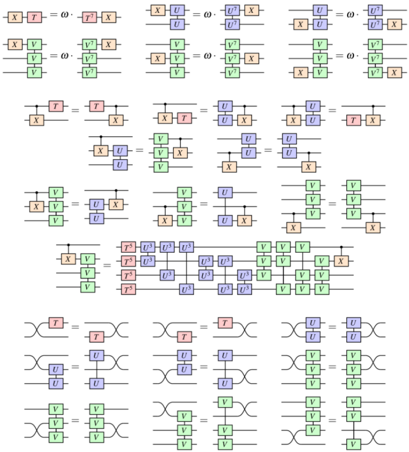
And more circuit equalities



*Selinger 2015

And even more circuit equalities





*Amy, Chen, & Ross 2018

Things get messy
because circuits are very rigid

Things get messy
because circuits are very rigid

Enter ZX-diagrams

ZX-diagrams

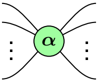
What gates are to circuits, *spiders* are to ZX-diagrams.

ZX-diagrams

What gates are to circuits, *spiders* are to ZX-diagrams.

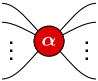
Z-spider

$$|0\cdots0\rangle\langle0\cdots0| + e^{i\alpha}|1\cdots1\rangle\langle1\cdots1|$$



X-spider

$$|+\cdots+\rangle\langle+\cdots+| + e^{i\alpha}|-\cdots-\rangle\langle-\cdots-|$$

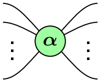


ZX-diagrams

What gates are to circuits, *spiders* are to ZX-diagrams.

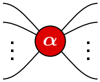
Z-spider

$$|0\cdots0\rangle\langle0\cdots0| + e^{i\alpha}\,|1\cdots1\rangle\langle1\cdots1|$$

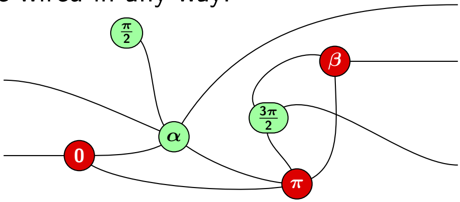


X-spider

$$|+\cdots+\rangle\langle+\cdots+| + e^{i\alpha}\,|-\cdots-\rangle\langle-\cdots-|$$

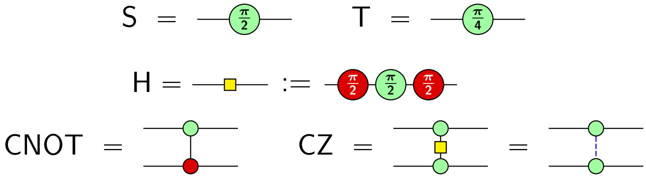


Spiders can be wired in any way:



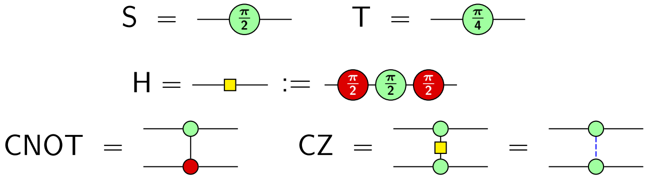
Quantum gates as ZX-diagrams

Every quantum gate can be written as a ZX-diagram:



Quantum gates as ZX-diagrams

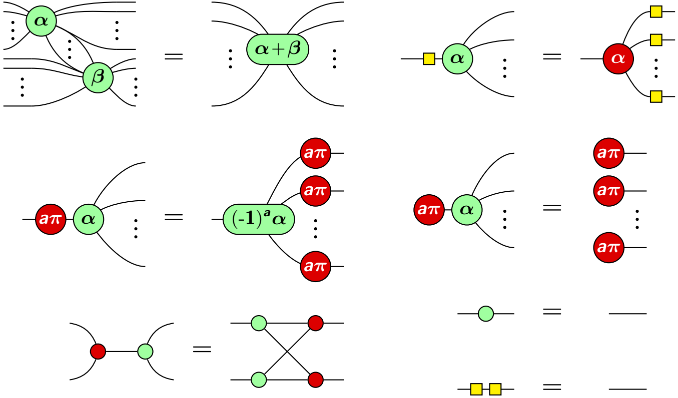
Every quantum gate can be written as a ZX-diagram:



Universality

Any linear map between qubits can be represented as a ZX-diagram.

Rules for ZX-diagrams: The ZX-calculus



$\alpha, \beta \in [0, 2\pi]$, $a \in \{0, 1\}$

Completeness of the ZX-calculus

Theorem (Vilmart 2018)

If two ZX-diagrams represent the same linear map, then they can be transformed into one another using the previous rules (and one additional one).

Completeness of the ZX-calculus

Theorem (Vilmart 2018)

If two ZX-diagrams represent the same linear map,
then they can be transformed into one another
using the previous rules (and one additional one).

So instead of dozens of circuit equalities,
we just need a few simple rules.

Optimisation using ZX-diagrams

- Write circuit as ZX-diagram.

Optimisation using ZX-diagrams

- Write circuit as ZX-diagram.
- Turn it into graph-like ZX-diagram.

Optimisation using ZX-diagrams

- Write circuit as ZX-diagram.
- Turn it into graph-like ZX-diagram.
- Simplify the diagram.

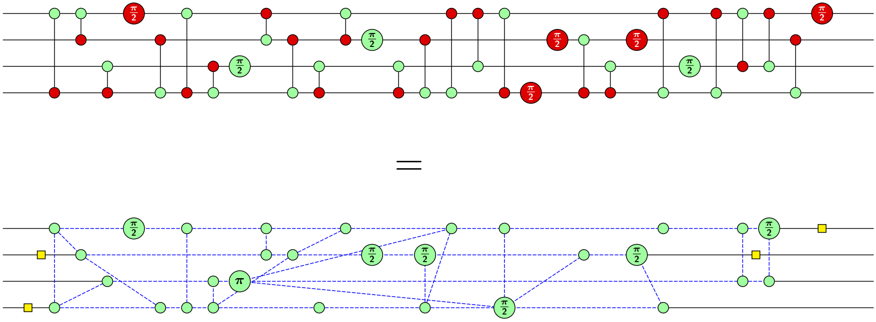
Optimisation using ZX-diagrams

- Write circuit as ZX-diagram.
- Turn it into graph-like ZX-diagram.
- Simplify the diagram.
- *Extract* a circuit from the diagram.

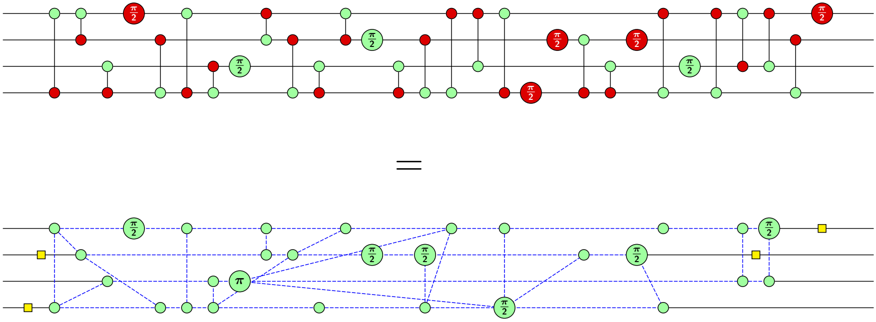
PyZX

- PyZX is an open-source Python library.
- `https://github.com/Quantomatic/pyzx`
- It allows easy manipulation of large ZX-diagrams.

Graph-like diagrams

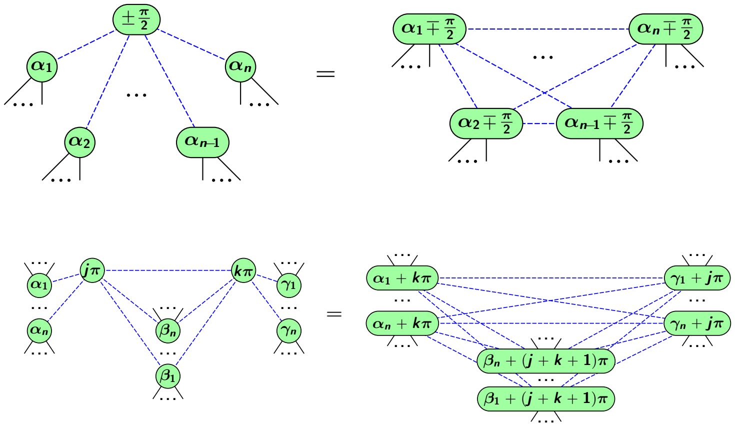


Graph-like diagrams



Now we are ready for simplification.
The game: Remove as many interior vertices as possible.

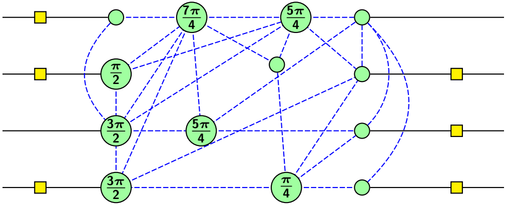
The tools: Local complementation and pivoting



Duncan, Kissinger, Perdrix, vdW (2019)

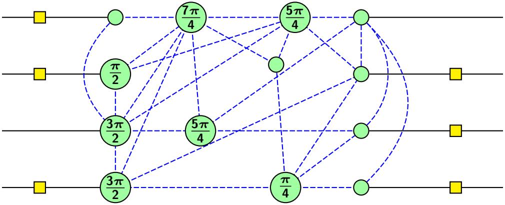
Example

Example result after simplification:



Example

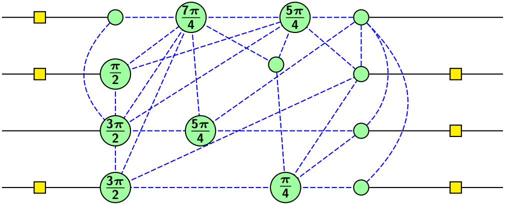
Example result after simplification:



Problem: does not look a circuit.

Example

Example result after simplification:

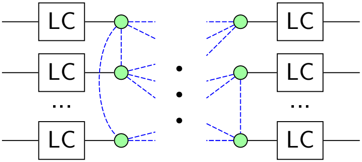


Problem: does not look a circuit.
Solution: all rewrites preserve *gflow*.

- Duncan, Perdrix, Kissinger, vdW (2019). *Graph-theoretic Simplification of Quantum Circuits with the ZX-calculus*.
- Backens, Miller-Bakewell, de Felice, Lobski, vdW (2020). *There and back again: A circuit extraction tale*.

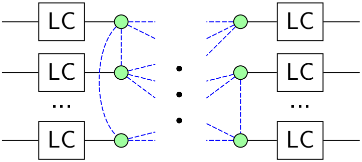
Clifford simplification

Clifford circuits are reduced to a pseudo-normal form:

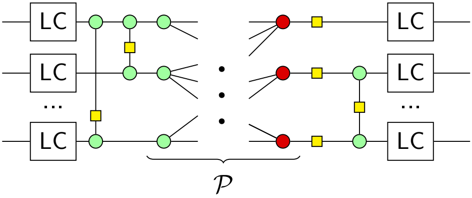


Clifford simplification

Clifford circuits are reduced to a pseudo-normal form:

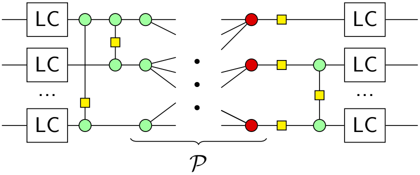


This is equal to:



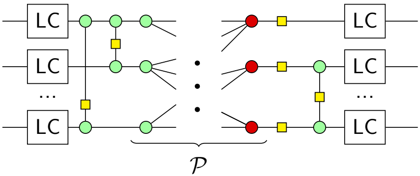
e.g. $\mathcal{P}\,|x_1, x_2, x_3, x_4\rangle \mapsto |x_1 \oplus x_2, x_1 \oplus x_3, x_4, x_3\rangle\,.$

Clifford normal form



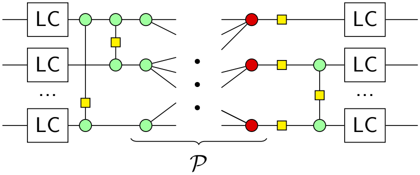
- Extracts to circuit with 8 layers:
- H - S - CZ - CNOT - H - CZ - S - H

Clifford normal form



- Extracts to circuit with 8 layers:
- H - S - CZ - CNOT - H - CZ - S - H
- Asymptotically optimal number of free parameters, like normal form of [Maslov & Roetteler 2019].

Clifford normal form

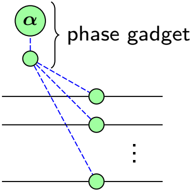


- Extracts to circuit with 8 layers:
- H - S - CZ - CNOT - H - CZ - S - H
- Asymptotically optimal number of free parameters, like normal form of [Maslov & Roetteler 2019].
- But additionally, linear nearest neighbour depth of $9n - 2$, a new record (Recently matched by [Bravyi & Maslov 2020]).

Non-Clifford optimisation

Non-Clifford optimisation

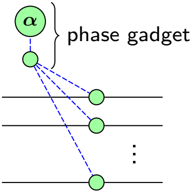
Additional rules for *phase gadgets*:



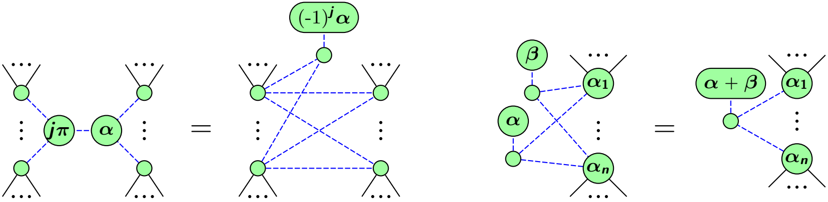
$$:: \quad |x_1, \ldots, x_n\rangle \mapsto e^{i\alpha(x_1 \oplus \ldots \oplus x_n)} |x_1, \ldots, x_n\rangle$$

Non-Clifford optimisation

Additional rules for *phase gadgets*:



$$:: \quad |x_1, \ldots, x_n\rangle \mapsto e^{i\alpha(x_1 \oplus \ldots \oplus x_n)} |x_1, \ldots, x_n\rangle$$



Kissinger, vdW 2019: *Reducing T-count with the ZX-calculus*

T-count optimisation

- At time of publishing, our method improved upon previous best T-counts for 6/36 benchmark circuits — in one case by 50%.

T-count optimisation

- At time of publishing, our method improved upon previous best T-counts for 6/36 benchmark circuits — in one case by 50%.
- Combining with TODD [Heyfron & Campbell 2018] we improved T-counts for 20/36 circuits.

T-count optimisation

- At time of publishing, our method improved upon previous best T-counts for 6/36 benchmark circuits — in one case by 50%.
- Combining with TODD [Heyfron & Campbell 2018] we improved T-counts for 20/36 circuits.
- Note: [Zhang & Chen 2019] use a different method that achieves nearly identical T-counts.

CNOT optimisation

- Circuit extraction resynthesises two-qubit gates.
- Sometimes this is beneficial, but sometimes it is not.

CNOT optimisation

- Circuit extraction resynthesises two-qubit gates.
- Sometimes this is beneficial, but sometimes it is not.
- Can be circumvented using *phase teleportation* [Kissinger & vdW 2019].

CNOT optimisation

- Circuit extraction resynthesises two-qubit gates.
- Sometimes this is beneficial, but sometimes it is not.
- Can be circumvented using *phase teleportation* [Kissinger & vdW 2019].
- Improves on previous best for quantum chemistry circuits of [Cowtan et al. 2019].

Conclusion

Using the ZX-calculus we found new techniques to improve depth, two-qubit gate count and T-count of realistic benchmark circuits.

Conclusion

Using the ZX-calculus we found new techniques to improve depth, two-qubit gate count and T-count of realistic benchmark circuits.

Future work:

- Allow routing for restricted architectures.

Conclusion

Using the ZX-calculus we found new techniques to improve depth, two-qubit gate count and T-count of realistic benchmark circuits.

Future work:

- Allow routing for restricted architectures.
- Improve extraction to reduce CNOT count.

Conclusion

Using the ZX-calculus we found new techniques to improve depth, two-qubit gate count and T-count of realistic benchmark circuits.

Future work:

- Allow routing for restricted architectures.
- Improve extraction to reduce CNOT count.
- Find ways to incorporate ancillae.

Thank you for your attention

References

Duncan, Kissinger, Perdrix & vdW 2019, arXiv:1902.03178
Graph-theoretic Simplification of Quantum Circuits with the ZX-calculus

Kissinger & vdW 2019, arXiv:1903.10477
Reducing T-count with the ZX-calculus

Backens, Miller-Bakewell, de Felice, Lobski & vdW 2020, arXiv:2003.01664
There and back again: A circuit extraction tale

Heyfron & Campbell 2018, arxiv:1712.01557
An Efficient Quantum Compiler that reduces T count

Maslov & Roetteler 2018, arXiv:1705.09176
Shorter stabilizer circuits via Bruhat decomposition and quantum circuit transformations

Zhang & Chen 2019, arXiv:1903.12456
Optimizing T gates in Clifford+T circuit as $\pi/4$ rotations around Paulis

Bravyi & Maslov 2020, arXiv:2003.09412
Hadamard-free circuits expose the structure of the Clifford group