Constructing quantum circuits with global gates

John van de Wetering

john@vdwetering.name

Radboud University Nijmegen
Oxford University

July 17, 2021

Announcement: Quantum Pubquiz

- What: A quantum Pubquiz!
- When: Thursday 20:30CEST
- Where: Gathertown pub

No need to register, just show up :)

The problem

- Most quantum circuit synthesis algorithms use 2-qubit gates (i.e. CNOTs)

The problem

- Most quantum circuit synthesis algorithms use 2-qubit gates (i.e. CNOTs)
- This is fine for many hardware architectures...
  ...but not all.

The problem

- Most quantum circuit synthesis algorithms use 2-qubit gates (i.e. CNOTs)
- This is fine for many hardware architectures...
  ...but not all.
- Ion trap quantum computers use *Mølmer-Sørensen* interaction that targets *many* qubits.

The problem

- Most quantum circuit synthesis algorithms use 2-qubit gates (i.e. CNOTs)
- This is fine for many hardware architectures...
  ...but not all.
- Ion trap quantum computers use *Mølmer-Sørensen* interaction that targets *many* qubits.
- Can we use this to our advantage somehow?

The problem

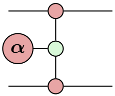
- Most quantum circuit synthesis algorithms use 2-qubit gates (i.e. CNOTs)
- This is fine for many hardware architectures...
  ...but not all.
- Ion trap quantum computers use *Mølmer-Sørensen* interaction that targets *many* qubits.
- Can we use this to our advantage somehow?
- As it turns out: yes!

Mølmer-Sørensen interaction

- For Pauli string $\vec{P}$ write $\vec{P}(\alpha) := \exp(-i\frac{\alpha}{2}\vec{P})$.

Mølmer-Sørensen interaction

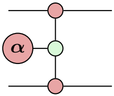
- For Pauli string $\vec{P}$ write $\vec{P}(\alpha) := \exp(-i\frac{\alpha}{2}\vec{P})$.
- 'Local' MS gate acting on qubits i and j implements $XX_{ij}(\alpha)$:

in ZX-calculus: $XX_{ij}(\alpha) \quad =$ 

Mølmer-Sørensen interaction

- For Pauli string $\vec{P}$ write $\vec{P}(\alpha) := \exp(-i\frac{\alpha}{2}\vec{P})$.
- 'Local' MS gate acting on qubits i and j implements $XX_{ij}(\alpha)$:

in ZX-calculus: $XX_{ij}(\alpha) \quad =$



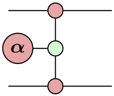
- *Global* MS gate (GMS) acting on set of qubits S is

$$\mathrm{GMS}_S(\alpha) = \prod_{i<j\in S} XX_{ij}(\alpha).$$

Mølmer-Sørensen interaction

- For Pauli string $\vec{P}$ write $\vec{P}(\alpha) := \exp(-i\frac{\alpha}{2}\vec{P})$.
- 'Local' MS gate acting on qubits i and j implements $XX_{ij}(\alpha)$:

in ZX-calculus: $XX_{ij}(\alpha) \quad =$



- *Global* MS gate (GMS) acting on set of qubits S is

$$\mathrm{GMS}_S(\alpha) = \prod_{i<j\in S} XX_{ij}(\alpha).$$

- If S can be arbitrary, we say the interaction is *targeted*
- If S is necessarily *all* the qubits, it is *untargeted*.

Some observations

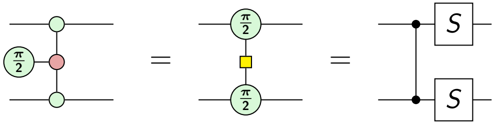
- Observation 1: Conjugating GMS gate by Hadamards gives diagonal gate $\prod_{i<j\in S} ZZ_{ij}(\alpha)$. This is just a phase polynomial.

Some observations

- Observation 1: Conjugating GMS gate by Hadamards gives diagonal gate $\prod_{i<j\in S} ZZ_{ij}(\alpha)$. This is just a phase polynomial.
- Observation 2: For $\alpha = \frac{\pi}{2}$ the GMS gate is Clifford.

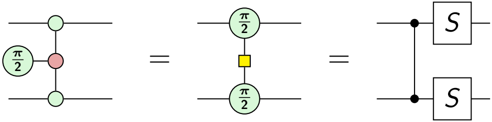
Some observations

- Observation 1: Conjugating GMS gate by Hadamards gives diagonal gate $\prod_{i<j\in S} ZZ_{ij}(\alpha)$. This is just a phase polynomial.
- Observation 2: For $\alpha = \frac{\pi}{2}$ the GMS gate is Clifford.
- Observation 3: For $\alpha = \frac{\pi}{2}$ the GMS gate is within local Cliffords of applying CZ gates everywhere:



Some observations

- Observation 1: Conjugating GMS gate by Hadamards gives diagonal gate $\prod_{i<j\in S} ZZ_{ij}(\alpha)$. This is just a phase polynomial.
- Observation 2: For $\alpha = \frac{\pi}{2}$ the GMS gate is Clifford.
- Observation 3: For $\alpha = \frac{\pi}{2}$ the GMS gate is within local Cliffords of applying CZ gates everywhere:



- Observation 4: A *global* CZ gate (GCZ) can be implemented using a single GMS gate + local Cliffords.

Compiling for targeted GMS gates

Input: circuit of Clifford gates and non-Clifford Z-phase gates.
Output: circuit of single-qubit gates and targeted GMS gates.

Compiling for targeted GMS gates

Input: circuit of Clifford gates and non-Clifford Z-phase gates.
Output: circuit of single-qubit gates and targeted GMS gates.

Algorithm:

1. Take first occurrence of non-Clifford gate $Z(\alpha)$.
   Push it all the way to the beginning to get $\vec{P}(\alpha)$.

$$Z_i(\alpha)C = C\exp(-i\frac{\alpha}{2}C^\dagger Z_i C) = C\exp(-i\frac{\alpha}{2}\vec{P}).$$

Compiling for targeted GMS gates

Input: circuit of Clifford gates and non-Clifford Z-phase gates.
Output: circuit of single-qubit gates and targeted GMS gates.

Algorithm:

1. Take first occurrence of non-Clifford gate $Z(\alpha)$.
   Push it all the way to the beginning to get $\vec{P}(\alpha)$.

$$Z_i(\alpha)C = C\exp(-i\frac{\alpha}{2}C^\dagger Z_i C) = C\exp(-i\frac{\alpha}{2}\vec{P}).$$

2. Conjugate $\vec{P}(\alpha)$ by Cliffords to reduce to phase gadget $\vec{P}'(\alpha)$ where each $P'_i = I$ or $P'_i = Z$.

Compiling for targeted GMS gates

Input: circuit of Clifford gates and non-Clifford Z-phase gates.
Output: circuit of single-qubit gates and targeted GMS gates.

Algorithm:

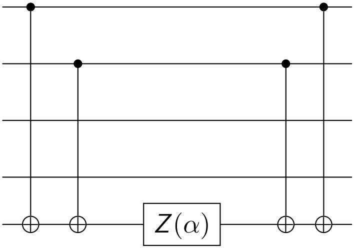
1. Take first occurrence of non-Clifford gate $Z(\alpha)$.
   Push it all the way to the beginning to get $\vec{P}(\alpha)$.

$$Z_i(\alpha)C = C\exp(-i\frac{\alpha}{2}C^\dagger Z_i C) = C\exp(-i\frac{\alpha}{2}\vec{P}).$$

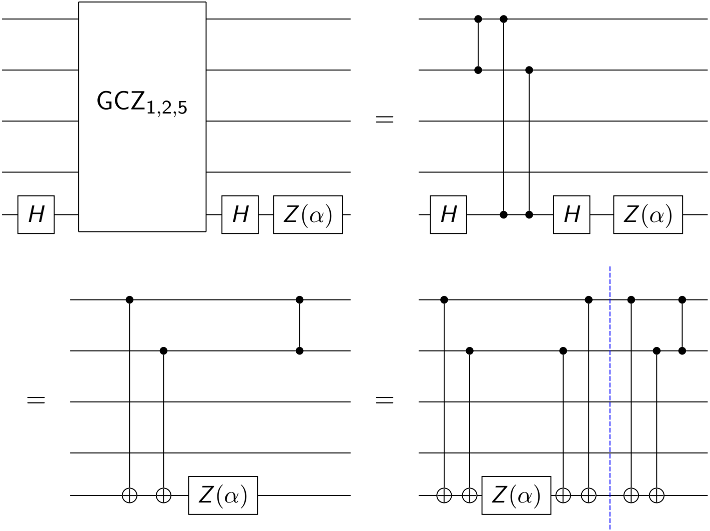
2. Conjugate $\vec{P}(\alpha)$ by Cliffords to reduce to phase gadget $\vec{P}'(\alpha)$ where each $P'_i = I$ or $P'_i = Z$.
3. Implement $\vec{P}'(\alpha)$ using GCZ gate up to 'Clifford garbage'...

Phase gadget CNOT ladder

$$ZZIIZ(\alpha) \quad =$$



Phase gadget using GCZ gate



Compiling for targeted GMS gates contd.

1. Take first occurrence of non-Clifford gate $Z(\alpha)$.
   Push it all the way to the left to get $\vec{P}(\alpha)$.
2. Conjugate it by Cliffords to reduce to phase polynomial $\vec{P}'(\alpha)$ where each $P_i' = I$ or $P_i' = Z$.
3. Implement $\vec{P}'(\alpha)$ using GCZ gate up to 'Clifford garbage'.

Compiling for targeted GMS gates contd.

1. Take first occurrence of non-Clifford gate $Z(\alpha)$.
   Push it all the way to the left to get $\vec{P}(\alpha)$.
2. Conjugate it by Cliffords to reduce to phase polynomial $\vec{P}'(\alpha)$ where each $P'_i = I$ or $P'_i = Z$.
3. Implement $\vec{P}'(\alpha)$ using GCZ gate up to 'Clifford garbage'.
4. Push next non-Clifford gate to frontier, repeat previous steps.
   If all non-Clifford gates processed go to next step.

Compiling for targeted GMS gates contd.

1. Take first occurrence of non-Clifford gate $Z(\alpha)$.
   Push it all the way to the left to get $\vec{P}(\alpha)$.
2. Conjugate it by Cliffords to reduce to phase polynomial $\vec{P'}(\alpha)$ where each $P'_i = I$ or $P'_i = Z$.
3. Implement $\vec{P'}(\alpha)$ using GCZ gate up to 'Clifford garbage'.
4. Push next non-Clifford gate to frontier, repeat previous steps.
   If all non-Clifford gates processed go to next step.
5. Synthesise remaining Clifford circuit.

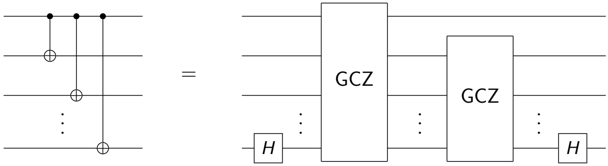
Clifford circuits with targeted GMS gates

Clifford circuits with targeted GMS gates

- Reduce the Clifford circuit to a normal form, i.e. Gottesman-Aaronson: C-P-C-P-H-P-C-P-C.

Clifford circuits with targeted GMS gates

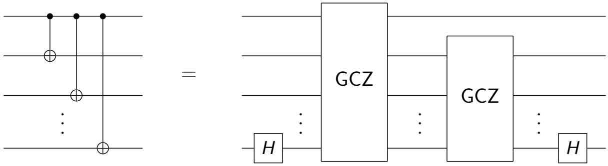
- Reduce the Clifford circuit to a normal form, i.e. Gottesman-Aaronson: C-P-C-P-H-P-C-P-C.
- Each CNOT 'fan-out' gate requires 2 GMS gates.



Hence, upper-triangular Boolean matrix requires $2n - 3$ GMS gates

Clifford circuits with targeted GMS gates

- Reduce the Clifford circuit to a normal form, i.e. Gottesman-Aaronson: C-P-C-P-H-P-C-P-C.
- Each CNOT 'fan-out' gate requires 2 GMS gates.

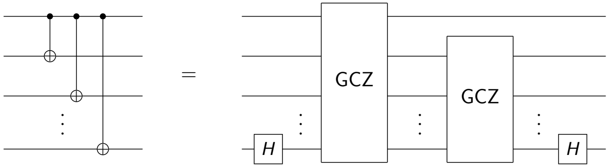


Hence, upper-triangular Boolean matrix requires $2n - 3$ GMS gates .

- The above normal form then requires $12n - 18$ GMS gates.

Clifford circuits with targeted GMS gates

- Reduce the Clifford circuit to a normal form, i.e. Gottesman-Aaronson: C-P-C-P-H-P-C-P-C.
- Each CNOT 'fan-out' gate requires 2 GMS gates.



Hence, upper-triangular Boolean matrix requires $2n - 3$ GMS gates .

- The above normal form then requires $12n - 18$ GMS gates.
- However: using 'GSLC' normal form H-S-CZ-CNOT-H-CZ-S-H we get $6n - 8$.

Compiling for targeted GMS gates

Theorem

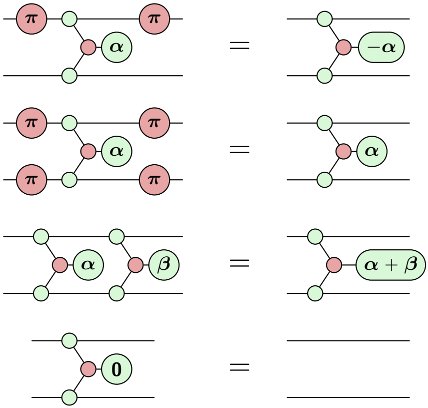
An n-qubit circuit consisting of

- Clifford gates
- and N non-Clifford Z-phase gates

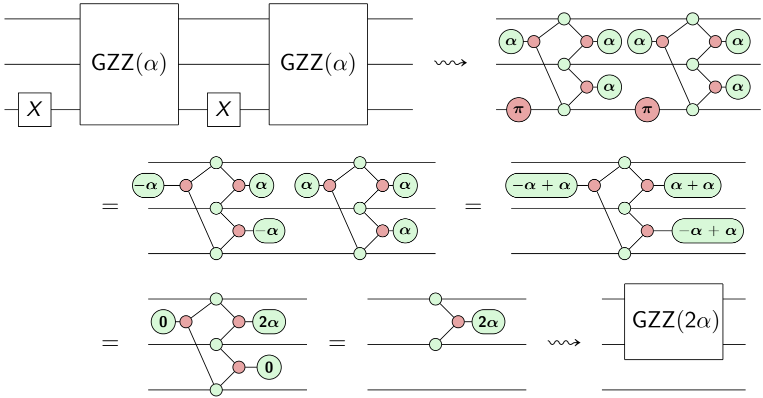
can be implemented using single qubit unitaries and at most $N + 6n - 8$ GMS gates.

Now to use untargeted GMS gates instead

Phase gadget identities

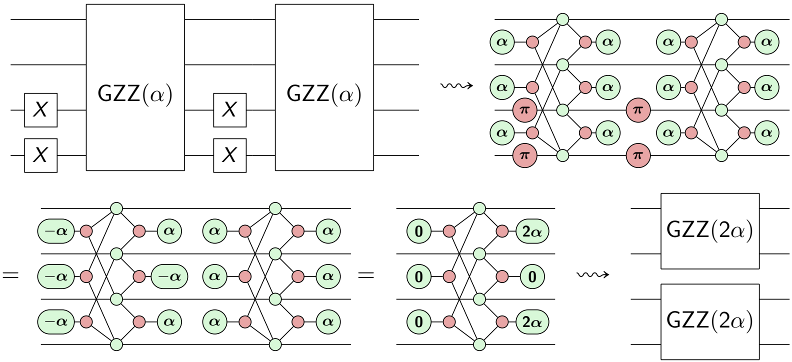


From global to local

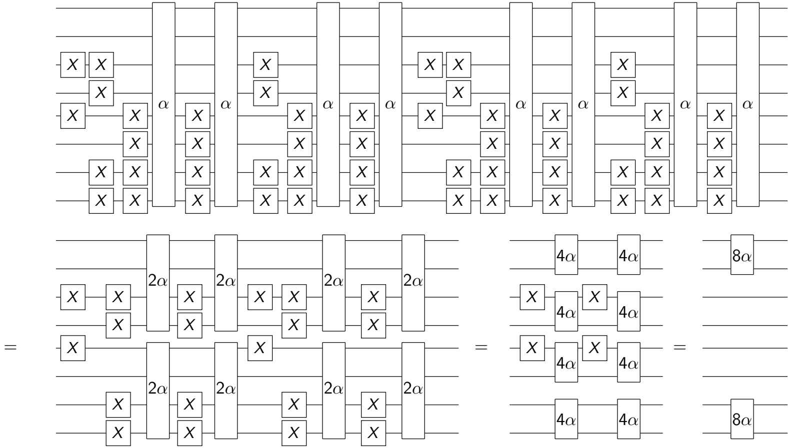


Identity from (Maslov & Nam, 2018)
To go from global on n qubits, to 'local' on k qubits requires 2^{n-k} GMS gates.

From global to more local



From 8-global to 2-local



Using the other method for the same result would require 128 GMS gates.

Cliffords using untargeted GMS gates

Proposition

An n-qubit CNOT circuit of depth d can be synthesised with local Cliffords and $< dn$ untargeted n-qubit $\mathrm{GZZ}(\frac{\pi}{2n})$ gates.

Cliffords using untargeted GMS gates

Proposition

An n-qubit CNOT circuit of depth d can be synthesised with local Cliffords and $< dn$ untargeted n-qubit $\mathrm{GZZ}(\frac{\pi}{2n})$ gates.

Asymptotically optimal depth of ancilla-free CNOT circuit on n qubits is $O(n/\log(n))$, hence:

Theorem

An n-qubit Clifford circuit can be synthesised using local Clifford gates and $O(n^2/\log(n))$ untargeted n-qubit $\mathrm{GZZ}(\frac{\pi}{2n})$ gates.

Cliffords using untargeted GMS gates

Proposition

An n-qubit CNOT circuit of depth d can be synthesised with local Cliffords and $< dn$ untargeted n-qubit $\mathrm{GZZ}(\frac{\pi}{2n})$ gates.

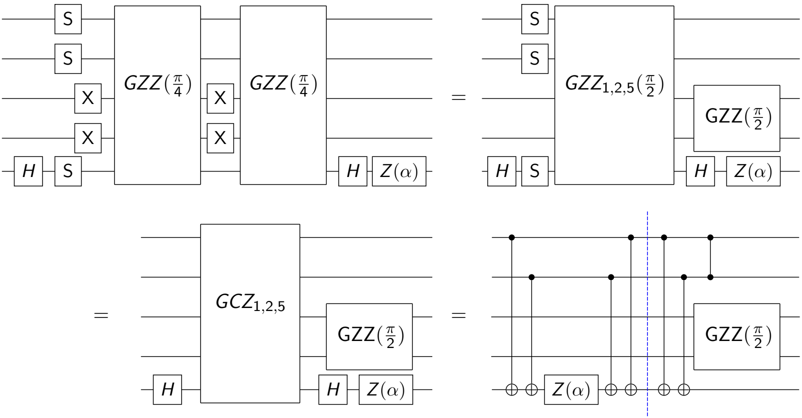
Asymptotically optimal depth of ancilla-free CNOT circuit on n qubits is $O(n/\log(n))$, hence:

Theorem

An n-qubit Clifford circuit can be synthesised using local Clifford gates and $O(n^2/\log(n))$ untargeted n-qubit $\mathrm{GZZ}(\frac{\pi}{2n})$ gates.

Note: this matches $O(n^2/\log(n))$ CNOT count for implementing an n-qubit Clifford.

Phase gadgets using untargeted GMS gates



Circuits to untargeted GMS gates

Theorem

An n-qubit circuit of

- Clifford gates
- and N non-Clifford Z-phase gates

can be synthesised using single qubit unitaries and $2N + O(n^2/\log(n))$ untargeted GMS gates.

Conclusion

Results for n-qubit circuit with N non-Clifford phases.

Conclusion

Results for n-qubit circuit with N non-Clifford phases.

Targeted GMS gates:

- Clifford: $6n - 8$
- Arbitrary: $N + 6n - 8$

Conclusion

Results for n-qubit circuit with N non-Clifford phases.

Targeted GMS gates:

- Clifford: $6n - 8$
- Arbitrary: $N + 6n - 8$

Untargeted GMS gates:

- Clifford: $O(n^2/\log(n))$
- Arbitrary: $2N + O(n^2/\log(n))$

Conclusion

Results for n-qubit circuit with N non-Clifford phases.

Targeted GMS gates:

- Clifford: $6n - 8$
- Arbitrary: $N + 6n - 8$

Untargeted GMS gates:

- Clifford: $O(n^2/\log(n))$
- Arbitrary: $2N + O(n^2/\log(n))$

Future questions:

- Are these bounds tight?
- Is there a benefit to allow α to vary?

Thank you for your attention

vdW 2020, arXiv:2012.09061
Constructing quantum circuits with global gates

Maslov & Nam 2017, arXiv:1707.06356
Use of global interactions in efficient quantum circuit constructions

vdW 2020, arXiv:2012.13966
ZX-calculus for the working quantum computer scientist

Announcement: Quantum Pubquiz

- What: A quantum Pubquiz!
- When: Thursday 20:30CEST
- Where: Gathertown pub

No need to register, just show up :)