Optimal Hadamard gate reduction in Clifford+R_Z circuits

Simon Martiel, Simon Perdrix, Vivien Vandaele and Christophe Vuillot

Défi EQIP

Motivations

- T gates are expensive to implement in most quantum codes.

Motivations

- T gates are expensive to implement in most quantum codes.
- Reducing the number of Hadamard gates can lead to a reduction in the number of T gates.

Motivations

- T gates are expensive to implement in most quantum codes.
- Reducing the number of Hadamard gates can lead to a reduction in the number of T gates.
- Every Clifford+T circuit can be optimized so that:

$$\tau \leqslant (n+1)(n+h)$$

where τ is the number of T gates in the circuit, h is the number of internal Hadamard gates in the circuit and n is the number of qubits.

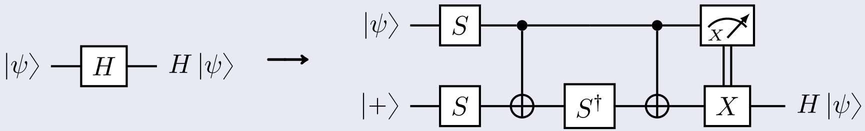
Internal Hadamard gate

A Hadamard gate is said to be internal if and only if there is at least one T gate that precedes it and one T gate that succeeds it.

Motivations

- Some T-count optimizers are gadgetizing internal Hadamard gates in order to further reduce the number of T gates.

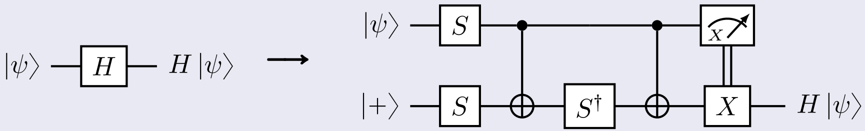
Hadamard gate gadgetization



Motivations

- Some T-count optimizers are gadgetizing internal Hadamard gates in order to further reduce the number of T gates.

Hadamard gate gadgetization



- This procedure requires one ancilla qubit per internal Hadamard gate, which motivates the minimization of internal Hadamard gates.

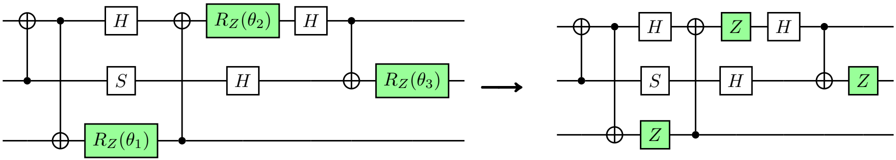
Clifford+R_Z circuits

Pauli rotation

$$R_P(\theta) = \cos(\theta/2)I - i\sin(\theta/2)P$$

where P is a Pauli operator and $\theta \in \mathbb{R}$.

For example the T gate is a $\pi/4$ Pauli Z rotation: $T = R_Z(\pi/4)$.

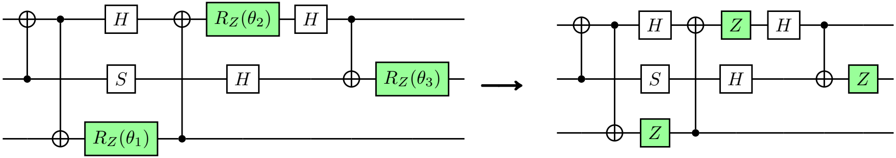


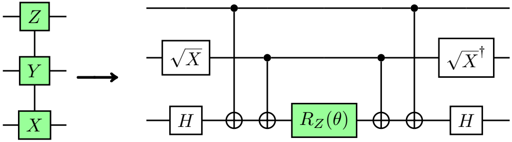
Pauli rotation

$$R_P(\theta) = \cos(\theta/2)I - i\sin(\theta/2)P$$

where P is a Pauli operator and $\theta \in \mathbb{R}$.

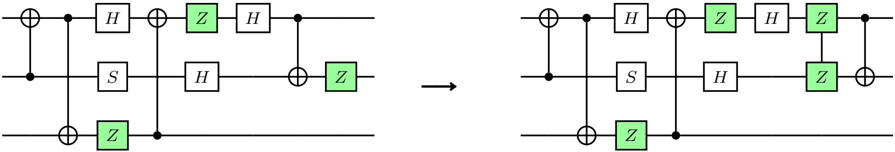
For example the T gate is a $\pi/4$ Pauli Z rotation: $T = R_Z(\pi/4)$.



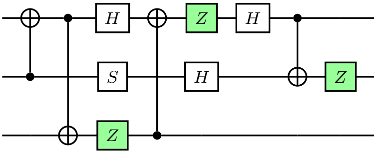


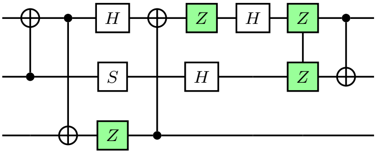
Clifford+R_Z circuits

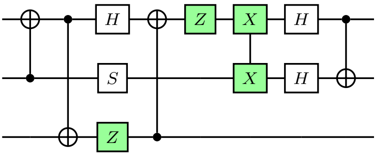
- Every Clifford+R_Z circuit can be characterized by a sequence of Pauli rotations followed by a final Clifford operator.



- Every Clifford+R_Z circuit can be characterized by a sequence of Pauli rotations followed by a final Clifford operator.

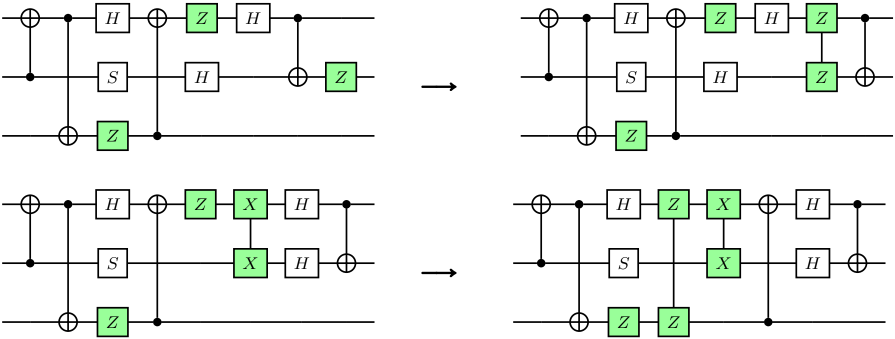






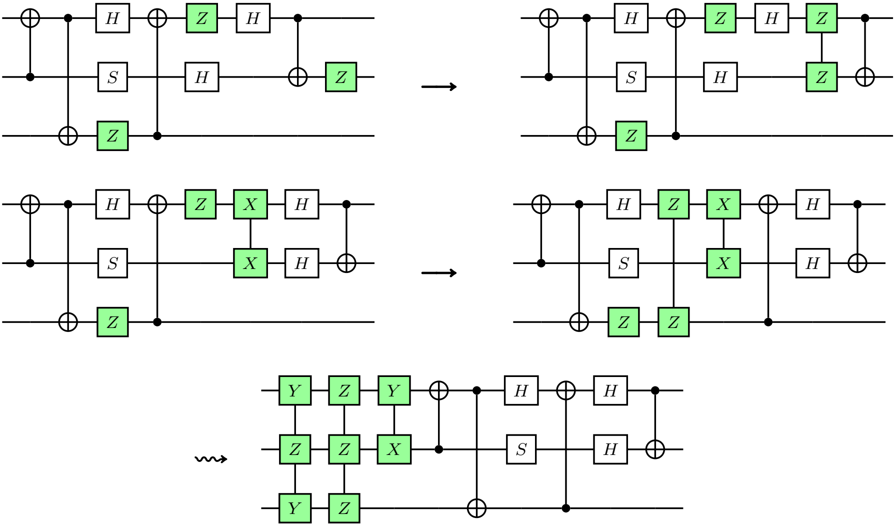
Clifford+R_Z circuits

- Every Clifford+R_Z circuit can be characterized by a sequence of Pauli rotations followed by a final Clifford operator.

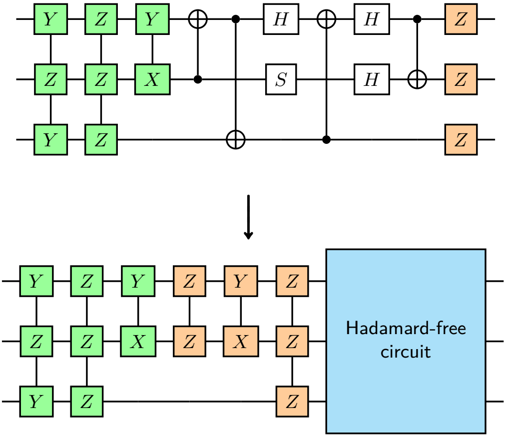


Clifford+R_Z circuits

- Every Clifford+R_Z circuit can be characterized by a sequence of Pauli rotations followed by a final Clifford operator.

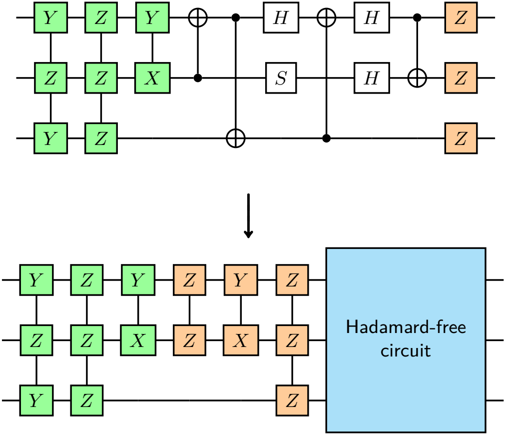


- A Clifford operator can be characterized by n stabilizers up to a $\{X, \mathrm{CNOT}, S\}$ circuit.



Clifford+R_Z circuits

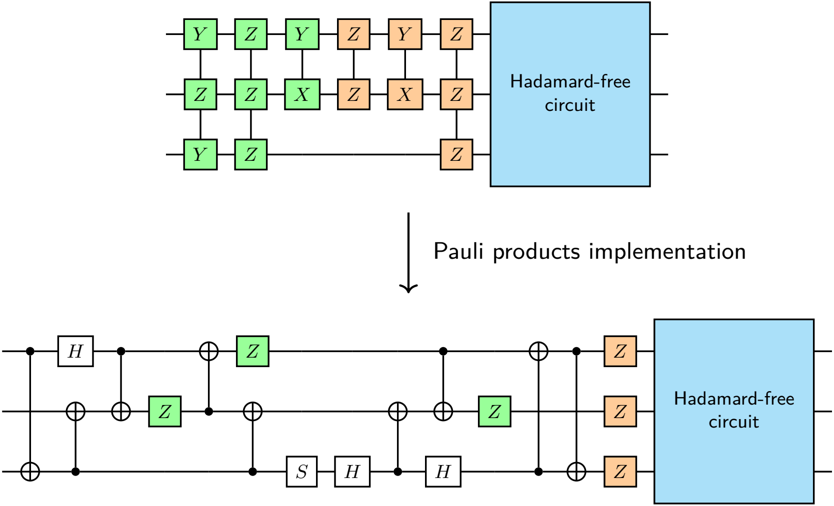
- A Clifford operator can be characterized by n stabilizers up to a $\{X, \mathrm{CNOT}, S\}$ circuit.



- We want to implement this sequence of Pauli products with a minimal number of Hadamard gates.

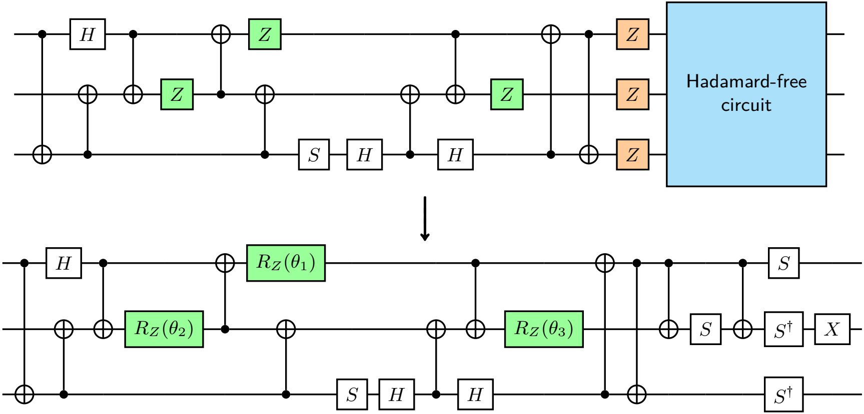
Clifford+R_Z circuits

- Implementing a sequence of Pauli products is done by inserting Clifford gates so that each Pauli products is composed of exactly one Z element.



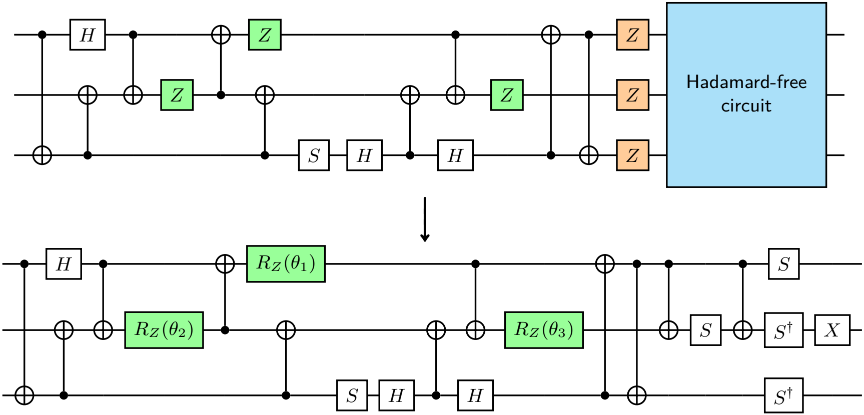
Clifford+R_Z circuits

- From this we can easily finish the synthesis of the circuit by inserting R_Z gates and the final Hadamard-free circuit.

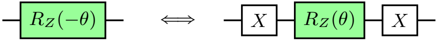


Clifford+R_Z circuits

- From this we can easily finish the synthesis of the circuit by inserting R_Z gates and the final Hadamard-free circuit.



- The sign can be switched using X gates:



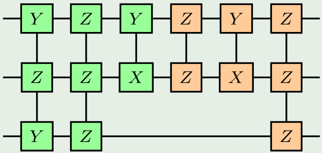
Pauli products encoding

The Pauli matrices can be encoded using 2 bits:

$$\begin{aligned} I &= (0,0) \qquad & Z &= (1,0) \\ Y &= (1,1) \qquad & X &= (0,1) \end{aligned}$$

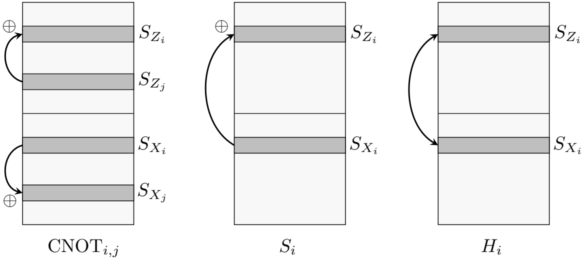
A sequence of m Pauli products can be encoded in a block matrix of size $2n \times m$: $S = \begin{bmatrix} S_Z \\ S_X \end{bmatrix}$.

Example

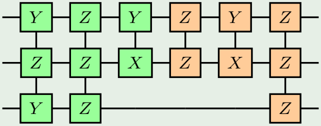


$$\longrightarrow \quad S = \begin{bmatrix} S_Z \\ S_X \end{bmatrix} = \left(\begin{array}{cccccc} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 \\ \hline 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{array}\right)$$

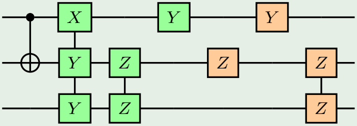
Pauli products encoding



Example



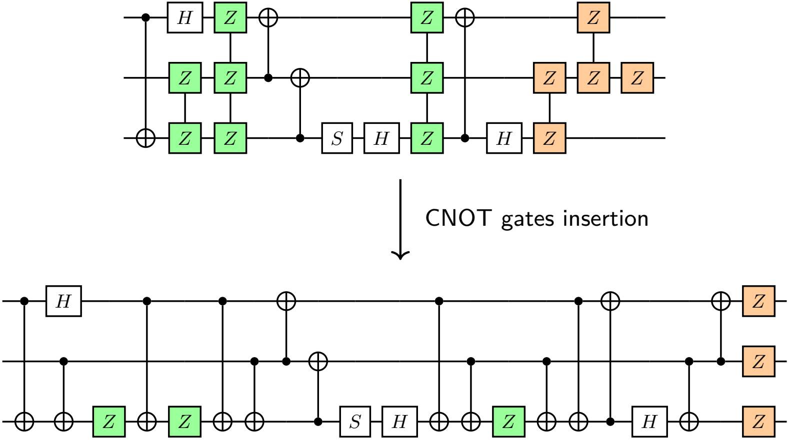
$\xrightarrow{\mathrm{CNOT}_{0,1}}$



$$S = \begin{bmatrix} S_Z \\ S_X \end{bmatrix} = \left(\begin{array}{cccccc} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 \\ \hline 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{array}\right)$$

$$S = \begin{bmatrix} S_Z \\ S_X \end{bmatrix} = \left(\begin{array}{cccccc} 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 \\ \hline 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{array}\right)$$

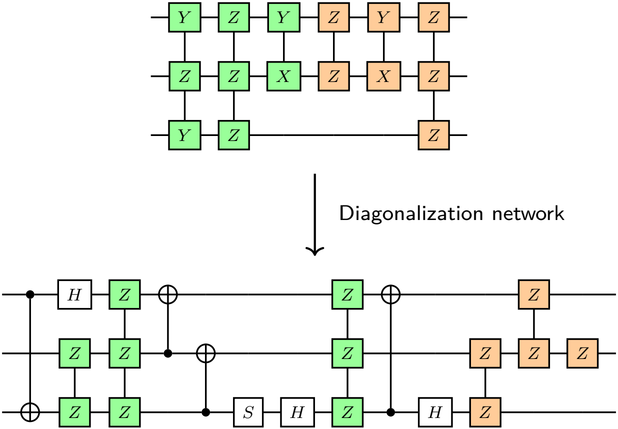
- A Pauli product is diagonal if its components are all Z or I matrices.
- If all Pauli products are diagonal, then their implementation can be completed by inserting only CNOT gates.



Diagonalization of Pauli products

Diagonalization network

A diagonalization network is a circuit in which all Pauli products are diagonal.



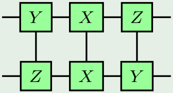
- Objective: construct a diagonalization network with a minimal number of Hadamard gates.

Simultaneous diagonalization

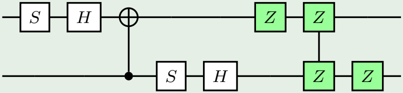
Simultaneous diagonalization problem

Find a Clifford circuit C, containing a minimal number of H gates, such that all Pauli products are diagonalized by C.

Example



$\longrightarrow$



$$S = \left(\begin{array}{ccc} 1 & 0 & 1 \\ 1 & 0 & 1 \\ \hline 1 & 1 & 0 \\ 0 & 1 & 1 \end{array}\right) \qquad S = \left(\begin{array}{ccc} 1 & 1 & 0 \\ 0 & 1 & 1 \\ \hline 0 & 0 & 0 \\ 0 & 0 & 0 \end{array}\right)$$

Simultaneous diagonalization problem

Find a Clifford circuit C, containing a minimal number of H gates, such that all Pauli products are diagonalized by C.

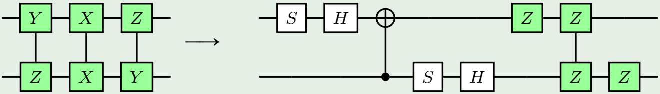
Proposition 1

At least $\mathrm{rank}(S_X)$ *Hadamard gates are required to simultaneously diagonalize the Pauli products encoded in* S.

Proof:

1. The only gate that can lower (by at most 1) the rank of S_X is the Hadamard gate.
2. If all Pauli products of S are diagonalized then $\mathrm{rank}(S_X) = 0$.

Example



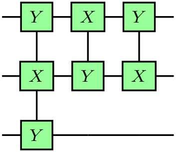
$$S = \left(\begin{array}{ccc} 1 & 0 & 1 \\ 1 & 0 & 1 \\ \hline 1 & 1 & 0 \\ 0 & 1 & 1 \end{array}\right) \qquad S = \left(\begin{array}{ccc} 1 & 1 & 0 \\ 0 & 1 & 1 \\ \hline 0 & 0 & 0 \\ 0 & 0 & 0 \end{array}\right)$$

Diagonalization network synthesis algorithm

Algorithm 1

Let S be a matrix encoded a sequence of m Pauli products.
For i going from 1 to m:

- If the ith Pauli product of S is not diagonal, then diagonalize it using one Hadamard gate.



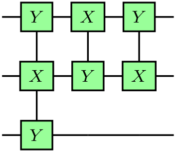
$$S = \left(\begin{array}{ccc} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ \hline 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 0 \end{array}\right)$$

Diagonalization network synthesis algorithm

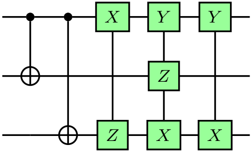
Algorithm 1

Let S be a matrix encoded a sequence of m Pauli products.
For i going from 1 to m:

- If the ith Pauli product of S is not diagonal, then diagonalize it using one Hadamard gate.



$$\xrightarrow[\text{CNOT}_{0,2}]{\text{CNOT}_{0,1}}$$



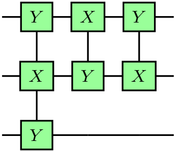
$$S = \left(\begin{array}{ccc} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ \hline 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 0 \end{array}\right) \qquad \left(\begin{array}{ccc} 0 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ \hline 1 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 1 & 1 \end{array}\right)$$

Diagonalization network synthesis algorithm

Algorithm 1

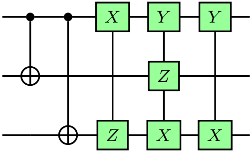
Let S be a matrix encoded a sequence of m Pauli products.
For i going from 1 to m:

- If the ith Pauli product of S is not diagonal, then diagonalize it using one Hadamard gate.



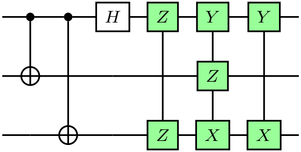
$$S = \left(\begin{array}{ccc} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ \hline 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 0 \end{array}\right)$$

$$\xrightarrow[\text{CNOT}_{0,2}]{\text{CNOT}_{0,1}}$$



$$\left(\begin{array}{ccc} 0 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ \hline 1 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 1 & 1 \end{array}\right)$$

$$\xrightarrow{H_0}$$



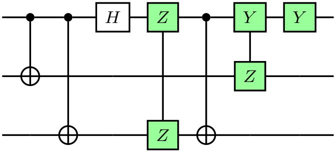
$$\left(\begin{array}{ccc} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ \hline 0 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 1 & 1 \end{array}\right)$$

Diagonalization network synthesis algorithm

Algorithm 1

Let S be a matrix encoded a sequence of m Pauli products.
For i going from 1 to m:

- If the ith Pauli product of S is not diagonal, then diagonalize it using one Hadamard gate.



$\xrightarrow{\text{CNOT}_{0,2}}$

$$\left(\begin{array}{cc} 1 & 1 \\ 1 & 0 \\ 0 & 0 \\ \hline 1 & 1 \\ 0 & 0 \\ 0 & 0 \end{array}\right)$$

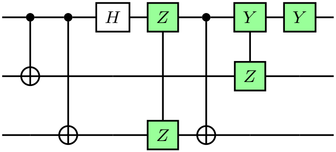
Diagonalization network synthesis algorithm

Algorithm 1

Let S be a matrix encoded a sequence of m Pauli products.
For i going from 1 to m:

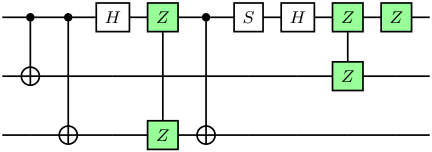
- If the ith Pauli product of S is not diagonal, then diagonalize it using one Hadamard gate.

$\xrightarrow{\text{CNOT}_{0,2}}$



$$\left(\begin{array}{cc} 1 & 1 \\ 1 & 0 \\ 0 & 0 \\ \hline 1 & 1 \\ 0 & 0 \\ 0 & 0 \end{array}\right)$$

$\xrightarrow[H_0]{S_0}$



$$\left(\begin{array}{cc} 1 & 1 \\ 1 & 0 \\ 0 & 0 \\ \hline 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{array}\right)$$

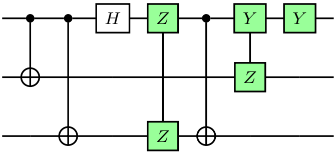
Diagonalization network synthesis algorithm

Algorithm 1

Let S be a matrix encoded a sequence of m Pauli products.
For i going from 1 to m:

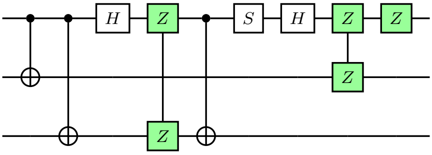
- If the ith Pauli product of S is not diagonal, then diagonalize it using one Hadamard gate.

$\xrightarrow{\text{CNOT}_{0,2}}$



$$\left(\begin{array}{cc} 1 & 1 \\ 1 & 0 \\ 0 & 0 \\ \hline 1 & 1 \\ 0 & 0 \\ 0 & 0 \end{array}\right)$$

$\xrightarrow[H_0]{S_0}$



$$\left(\begin{array}{cc} 1 & 1 \\ 1 & 0 \\ 0 & 0 \\ \hline 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{array}\right)$$

Proposition 2

The circuit constructed by Algorithm 1 contains $\mathrm{rank}(S_X)$ *Hadamard gates and is an optimal solution to the simultaneous diagonalization network problem.*

Diagonalization network synthesis

Diagonalization network synthesis problem

Let S be a matrix encoding a sequence of m Pauli products.
Find a diagonalization network for S containing a minimal number of Hadamard gates.

Diagonalization network synthesis

Diagonalization network synthesis problem

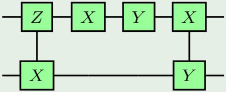
Let S be a matrix encoding a sequence of m Pauli products.
Find a diagonalization network for S containing a minimal number of Hadamard gates.

Commutativity matrix

The commutativity matrix A associated with S is a strictly upper triangular Boolean matrix of size $m \times m$ such that for all $i < j$:

$$\begin{aligned} A_{i,j} &= 0 \quad \text{if } S_{:,i} \text{ commutes with } S_{:,j}, \\ A_{i,j} &= 1 \quad \text{if } S_{:,i} \text{ anticommutes with } S_{:,j}. \end{aligned}$$

Example



$$\longrightarrow \quad A = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

$$S = \left(\begin{array}{cccc} 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ \hline 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \end{array}\right)$$

Diagonalization network synthesis

Theorem 1

Algorithm 1 solves the diagonalization network synthesis problem optimally using $\mathrm{rank}(M)$ *Hadamard gates, where* $M = \begin{bmatrix} S_X \\ A \end{bmatrix}$.

Diagonalization network synthesis

Theorem 1

Algorithm 1 solves the diagonalization network synthesis problem optimally using $\mathrm{rank}(M)$ *Hadamard gates, where* $M = \begin{bmatrix} S_X \\ A \end{bmatrix}$.

Internal Hadamard gates minimization problem

Let S be a matrix encoding a sequence of Pauli products.
Find a diagonalization network for S containing a minimal number of **internal** Hadamard gates.

Diagonalization network synthesis

Theorem 1

Algorithm 1 solves the diagonalization network synthesis problem optimally using $\mathrm{rank}(M)$ *Hadamard gates, where* $M = \begin{bmatrix} S_X \\ A \end{bmatrix}$.

Internal Hadamard gates minimization problem

Let S be a matrix encoding a sequence of Pauli products.
Find a diagonalization network for S containing a minimal number of **internal** Hadamard gates.

Theorem 2

There exists an algorithm solving the internal Hadamard gates minimization problem optimally using $\mathrm{rank}(A)$ *internal Hadamard gates.*

Summary

Value to optimize	H-count	Complexity
H gates	$\text{rank}(M)$	$\mathcal{O}(n^2m)$
Internal H gates (approximation)	$\leqslant n + \text{rank}(A)$	$\mathcal{O}(n^2m)$
Internal H gates	$\text{rank}(A)$	$\mathcal{O}(m^3)$

For a sequence S of Pauli products where

- n is the number of qubits, m is the number of Pauli products,
- A is the commutativity matrix associated with S,
- $M = \begin{bmatrix} S_X \\ A \end{bmatrix}$.