① **Classical Computation**: manipulation of bits via logic gates (AND, OR, NOT, XOR)

For input $x \in \{0,1\}^n$, calculate $f(x)$ for some f.
(How to realize f with a circuit? How many gates/depth is necessary and/or sufficient?)

⊛ We rarely think about computation in these terms anymore!

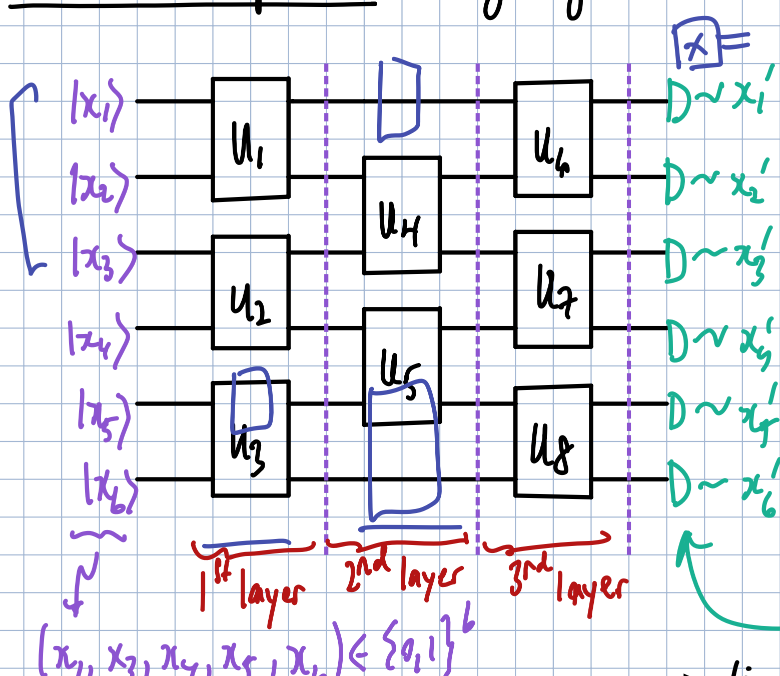
⊛ We program in some high-level language (C++, python, julia) and then the code gets compiled into the machine-level language.

⊛ Quantum computing is still thought about at the gate level.

cirq

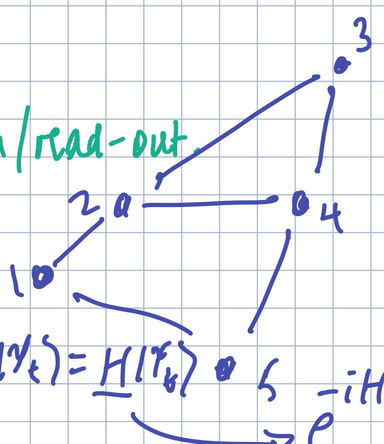
(Although "quantum programming languages" are being developed — qiskit is a simple example.)

② **Quantum Computation**: logic gates are unitaries! ($U: U^\dagger U = UU^\dagger = \mathbb{1}$).



⊛ To calculate the output state, multiply the matrices in the correct order.

$$U_8 U_7 U_6 U_5 U_4 U_3 U_2 U_1 |x\rangle$$



⊛ **Observe**: H Hermitian $\Rightarrow \left(e^{-iHt}\right)$ is unitary! $\left(e^{-iHt}\right)^\dagger = e^{iH^\dagger t} = e^{iHt}$

$H^\dagger = H$ $\qquad e^{-iHt} \cdot e^{iHt} = \mathbb{1}$

$i\hbar \frac{\partial}{\partial t}|\psi_t\rangle = H|\psi_t\rangle$ $\qquad e^{-iHt}$

Shor Grover

Tasks: • Classical problems (e.g., factoring, search)

→ How to (best) encode the classical input into the q. computer?

→ Is there an advantage/speedup over know classical algorithms?

• For $x \in \{0,1\}^n$, prepare $U|x\rangle$ for some unitary U.

→ How to realize U as a circuit? (optimally?).

We want a "universal gate set"

→ Special case: prepare $U|0\rangle^{\otimes n}$ (e.g., ground state of a Hamiltonian)

Recall: $H = \sum_{k=1}^{?} \lambda_k |\psi_k\rangle\langle\psi_k|$

"ground state" is an eigenstate corresponding to $\lambda_{min}(H)$ → the smallest eigenvalue of H.

→ Prepare samples from some probability distribution. (usually the probability distribution (or amplitudes) contains the answer to classical problems).

③ Quantum gates

⊛ Pauli gates:

$-\boxed{X}- \rightarrow X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad (X|0\rangle = |1\rangle,\ X|1\rangle = |0\rangle)$

$-\boxed{Y}- \rightarrow Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \quad (Y|0\rangle = i|1\rangle,\ Y|1\rangle = -i|0\rangle)$

$-\boxed{Z}- \rightarrow Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \quad \left(Z|0\rangle = |0\rangle,\ Z|1\rangle = -|1\rangle \right)$

⊛ **Hadamard gate**: $-\boxed{H}- \rightarrow H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$

$$H|0\rangle = |+\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right)$$
$$H|1\rangle = |-\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle - |1\rangle\right).$$

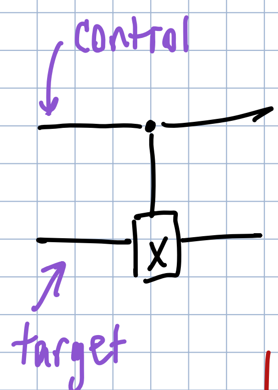
⊛ **Phase gate**: $-\boxed{S}- \rightarrow S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$

$\hookrightarrow e^{i\pi/2}$

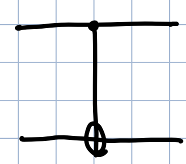
⊛ **T-gate**: $-\boxed{T}- \rightarrow T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}$

⊛ **CNOT / CX gate**:

"controlled-X"



or



$$\text{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

$$|0\rangle|0\rangle \longmapsto |0\rangle|0\rangle$$
$$|0\rangle|1\rangle \longmapsto |0\rangle|1\rangle$$
$$|1\rangle|0\rangle \longmapsto |1\rangle X|0\rangle = |1\rangle|1\rangle$$
$$|1\rangle|1\rangle \longmapsto |1\rangle X|1\rangle = |1\rangle|0\rangle$$

B/c of linearity, this determines the action on any state!

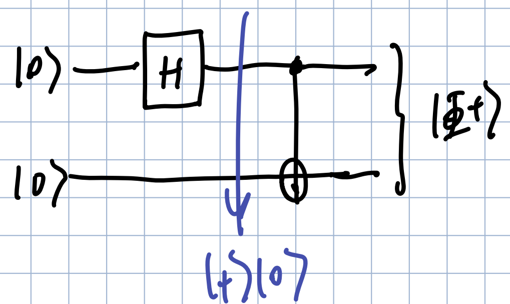
$$|\psi\rangle = a|0,0\rangle + b|0,1\rangle + c|1,0\rangle + d|1,1\rangle \mapsto a|0,0\rangle + b|0,1\rangle + c|1,1\rangle + d|1,0\rangle$$

CNOT is an <u>entangling gate</u>

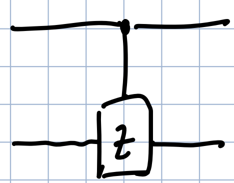
$$|+\rangle|0\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle+|1\rangle\right)\otimes|0\rangle = \frac{1}{\sqrt{2}}\left(|0,0\rangle+|1,0\rangle\right)$$

$$CNOT\,|+\rangle|0\rangle = \frac{1}{\sqrt{2}}\left(|0,0\rangle+|1,1\rangle\right) = |\Phi^+\rangle.$$

$$\frac{1}{\sqrt{2}}\left[\underbrace{CNOT\,|0,0\rangle}_{|0,0\rangle} + \underbrace{CNOT\,|1,0\rangle}_{|1,1\rangle}\right]$$



⊛ **Controlled-Z / CZ gate** :



$|0\rangle|0\rangle \mapsto |0\rangle|0\rangle$

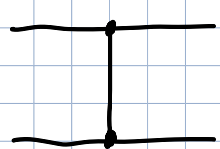
$|0\rangle|1\rangle \mapsto |0\rangle|1\rangle$

$|1\rangle|0\rangle \mapsto |1\rangle Z|0\rangle = |1\rangle|0\rangle$

$|1\rangle|1\rangle \mapsto |1\rangle Z|1\rangle = -|1\rangle|1\rangle$

$$CZ = \begin{pmatrix} 1 & & & 0 \\ & 1 & & \\ 0 & & 1 & \\ & & & -1 \end{pmatrix}$$

OR

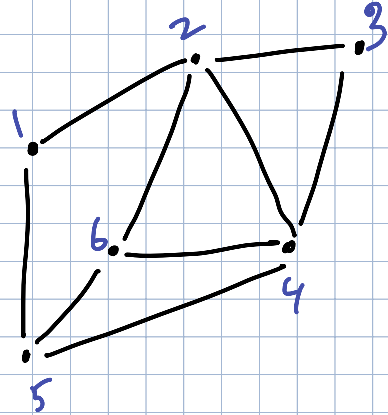


This is also an entangling gate!

Graph States $\quad G = (V, E)$ is a graph

edges.

vertices/nodes



Recipe:
1. Every node is a qubit
2. Prepare all qubits in $|+\rangle$
3. If there is an edge b/w

two nodes, apply CZ b/w the corresponding qubits.

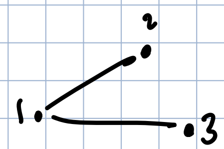
$$|+\rangle^{\otimes n} = \left(\frac{1}{\sqrt{2}}(|0\rangle+|1\rangle)\right)^{\otimes n} = \frac{1}{\sqrt{2^n}}\sum_{\vec{x}\in\{0,1\}^n}|\vec{x}\rangle$$

$$CZ|x_1,x_2\rangle = (-1)^{x_1x_2}|x_1,x_2\rangle$$

adjacency matrix of graph.

$$\Rightarrow |+\rangle^{\otimes n} \longmapsto \frac{1}{\sqrt{2^n}}\sum_{x\in\{0,1\}^n}(-1)^{\frac{1}{2}\vec{x}^T A\vec{x}}|x\rangle \qquad \to (-1)^{x_1x_2+x_1x_3}$$

Example:



$$A = \begin{array}{c c} & \begin{array}{ccc}1&2&3\end{array} \\ \begin{array}{c}1\\2\\3\end{array} & \begin{pmatrix}0&1&1\\1&0&0\\1&0&0\end{pmatrix}\end{array}$$

$$(x_1\ x_2\ x_3)\begin{pmatrix}0&1&1\\1&0&0\\1&0&0\end{pmatrix}\begin{pmatrix}x_1\\x_2\\x_3\end{pmatrix} = x_1x_2+x_1x_3$$

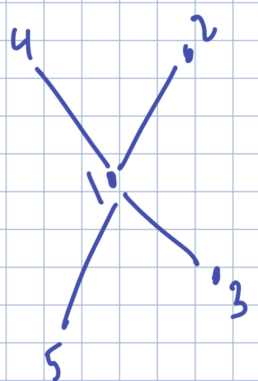
$$|+\rangle|+\rangle|+\rangle = \frac{1}{\sqrt{2^3}}\Big(|0\rangle|0\rangle|0\rangle+|0\rangle|0\rangle|1\rangle+|0\rangle|1\rangle|0\rangle+|0\rangle|1\rangle|1\rangle$$
$$|1\rangle|0\rangle|0\rangle+|1\rangle|0\rangle|1\rangle+|1\rangle|1\rangle|0\rangle+|1\rangle|1\rangle|1\rangle\Big)$$

$$\longmapsto \frac{1}{\sqrt{2^3}}\Big(|0\rangle|0\rangle|0\rangle+|0\rangle|0\rangle|1\rangle+|0\rangle|1\rangle|0\rangle+|0\rangle|1\rangle|1\rangle$$
$$+|1\rangle|0\rangle|0\rangle-|1\rangle|0\rangle|1\rangle-|1\rangle|1\rangle|0\rangle+(-1)^2|1\rangle|1\rangle|1\rangle\Big).$$

$(-1)^{0+1}$

$$=\frac{1}{\sqrt{2^3}}\Big(|0\rangle|0\rangle(|0\rangle+|1\rangle)+|0\rangle|1\rangle(|0\rangle+|1\rangle)$$
$$+|1\rangle|0\rangle(|0\rangle-|1\rangle)-|1\rangle|1\rangle(|0\rangle-|1\rangle)\Big)$$

$$=\frac{1}{\sqrt{2^2}}\Big(|0\rangle|0\rangle|+\rangle+|0\rangle|1\rangle|+\rangle$$
$$+|1\rangle|0\rangle|-\rangle-|1\rangle|1\rangle|-\rangle\Big)$$



$$=\frac{1}{\sqrt{2^2}}\Big(|0\rangle(|0\rangle+|1\rangle)|+\rangle+|1\rangle(|0\rangle-|1\rangle)|-\rangle\Big)$$

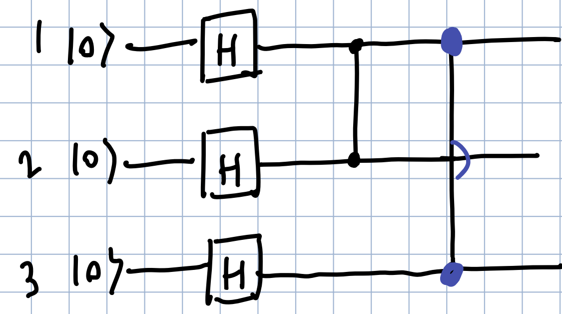
$$=\frac{1}{\sqrt{2}}\Big(|0\rangle|+\rangle|+\rangle+|1\rangle|-\rangle|-\rangle\Big)$$

$H|0\rangle=|+\rangle$

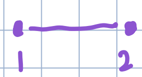
$H|1\rangle=|-\rangle$

$$=(\mathbb{1}\otimes H\otimes H)\underbrace{\frac{1}{\sqrt{2}}\Big(|0\rangle|0\rangle|0\rangle+|1\rangle|1\rangle|1\rangle\Big)}_{|GHZ_3\rangle}$$

As a circuit:



Star graphs $\longleftrightarrow$ GHZ states. $\frac{1}{\sqrt{2}}\left(|0\rangle^{\otimes n} + |1\rangle^{\otimes n}\right).$



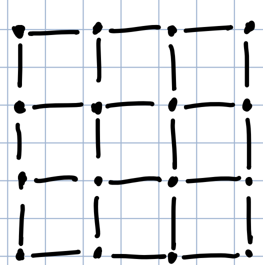
$\longleftrightarrow$ $|+\rangle|+\rangle = \frac{1}{\sqrt{2}^2}\left(|0\rangle|0\rangle + |0\rangle|1\rangle + |1\rangle|0\rangle + |1\rangle|1\rangle\right)$

$$\mapsto \frac{1}{\sqrt{2}^2}\left(|0\rangle|0\rangle + |0\rangle|1\rangle + |1\rangle|0\rangle - |1\rangle|1\rangle\right)$$

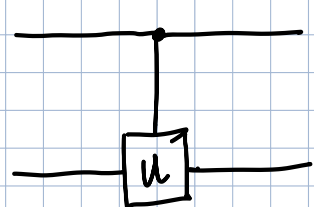
$$= \frac{1}{\sqrt{2}}\left(|0\rangle|+\rangle + |1\rangle|-\rangle\right)$$

$$= (\mathbb{1} \otimes H)\underbrace{\frac{1}{\sqrt{2}}\left(|0\rangle|0\rangle + |1\rangle|1\rangle\right)}_{|\Phi^+\rangle}$$

Square lattice $\longleftrightarrow$ Cluster states.



⊛ Controlled Unitary



$|0\rangle|0\rangle \longmapsto |0\rangle|0\rangle$

$|0\rangle|1\rangle \longmapsto |0\rangle|1\rangle$

$|1\rangle|0\rangle \longmapsto |1\rangle U|0\rangle$

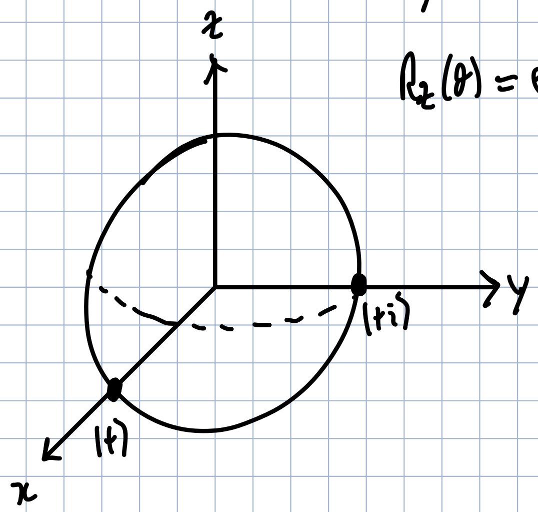
$|1\rangle|1\rangle \longmapsto |1\rangle U|1\rangle$

$$cU = \begin{pmatrix} \mathbb{1} & 0 \\ 0 & U \end{pmatrix} = |0\rangle\langle 0| \otimes \mathbb{1} + |1\rangle\langle 1| \otimes U.$$

(*) Rotation Gates: $R_X(\theta) = e^{-i\frac{\theta}{2}X}$ → rotation around X-axis by angle θ.

$R_Y(\theta) = e^{-i\frac{\theta}{2}Y}$ → rotation around Y-axis by angle θ.

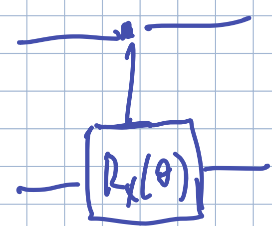
$R_Z(\theta) = e^{-i\frac{\theta}{2}Z}$ → rotation around Z-axis by angle θ.



$X^2 = \mathbb{1}$

$$e^M := \sum_{k=0}^{\infty} \frac{1}{k!} M^k$$

$$e^X = \sum_{k=0}^{\infty} \frac{1}{k!} X^k$$



$$\Rightarrow e^{-i\frac{\theta}{2}X} = \sum_{k=0}^{\infty} \frac{1}{k!} \left(-i\frac{\theta}{2}X\right)^k$$

$$\begin{pmatrix} X^2 = \mathbb{1} \\ \Rightarrow X^k = \mathbb{1},\ k \text{ even} \\ X^k = X,\ k \text{ odd} \end{pmatrix}$$

$$= \sum_{k=0}^{\infty} \frac{1}{\underbrace{(2k)!}_{0,2,4,6,\dots}} \left(-i\frac{\theta}{2}X\right)^{2k} + \sum_{k=0}^{\infty} \frac{1}{\underbrace{(2k+1)!}_{1,3,5,7,\dots}} \left(-i\frac{\theta}{2}X\right)^{2k+1}$$

$$= \sum_{k=0}^{\infty} \frac{1}{(2k)!} \left(\frac{\theta}{2}\right)^{2k} \underbrace{(-i)^{2k}} \mathbb{1} + \sum_{k=0}^{\infty} \frac{1}{(2k+1)!} \left(\frac{\theta}{2}\right)^{2k+1} (-i)^{2k+1} X$$

$(-i)^1 = -i \quad (k=0)$

$(-i)^3 = \underbrace{-i\cdot -i}_{-1}\cdot -i = i \quad (k=1)$

$(-i)^5 = -i \quad (k=2)$

$\vdots$

$(-i)^{2k+1} = -i(-1)^k$

$(-i)^2 = -i\cdot -i = -1 \quad (k=1)$

$(-i)^4 = ((-i)^2)^2 = 1 \quad (k=2)$

$(-i)^6 = -1 \quad (k=3)$

$\vdots$

$(-i)^{2k} = (-1)^k$

$$= \underbrace{\sum_{k=0}^{\infty} \frac{1}{(2k)!}\left(\frac{\theta}{2}\right)^{2k}(-1)^k}_{\cos\left(\frac{\theta}{2}\right)} \mathbb{1} - i \underbrace{\sum_{k=0}^{\infty} \frac{1}{(2k+1)!}\left(\frac{\theta}{2}\right)^{2k+1}(-1)^k}_{\sin\left(\frac{\theta}{2}\right)} X$$

$$= \cos\left(\frac{\theta}{2}\right)\mathbb{1} - i\sin\left(\frac{\theta}{2}\right) X$$

$e^{-i\frac{\theta}{2}}$

$\left(\cos\left(\frac{\theta}{2}\right) - i\sin\left(\frac{\theta}{2}\right)\right.$

$\cos\left(\frac{\theta}{2}\right) + i\sin\left(\frac{\theta}{2}\right) \rightarrow e^{i\frac{\theta}{2}}$

⊛ Similar argument to show $R_y(\theta) = e^{-i\frac{\theta}{2}Y} = \cos\left(\frac{\theta}{2}\right)\mathbb{1} - i\sin\left(\frac{\theta}{2}\right)Y$

$R_z(\theta) = e^{-i\frac{\theta}{2}Z} = \underline{\cos\left(\frac{\theta}{2}\right)\mathbb{1} - i\sin\left(\frac{\theta}{2}\right)Z}$

$\left(\cos\left(\frac{\theta}{2}\right) \pm i\sin\left(\frac{\theta}{2}\right) = e^{\pm i\frac{\theta}{2}}\right)$

⊛ Note: $R_z(\theta) = \begin{pmatrix} e^{-i\theta/2} & \\ & e^{i\theta/2} \end{pmatrix}$

$$R_z\left(\frac{\pi}{2}\right) = \begin{pmatrix} e^{-i\pi/4} & \\ & e^{i\pi/4} \end{pmatrix} = e^{-i\frac{\pi}{4}}\begin{pmatrix} 1 & 0 \\ 0 & \underbrace{e^{i\frac{\pi}{2}}}_{=i} \end{pmatrix} = e^{-i\frac{\pi}{4}}\begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix} \sim \underline{S} \text{ gate!}$$

(up to global phase).

④ Clifford gates → Important class of gates!

IEEE TRANSACTIONS ON INFORMATION THEORY, VOL. 44, NO. 4, JULY 1998 arXiv:quant-ph/9608006 1369

Quantum Error Correction Via Codes Over GF (4)

A. Robert Calderbank, *Fellow, IEEE*, Eric M. Rains, P. W. Shor, and Neil J. A. Sloane, *Fellow, IEEE*

Surviving as a Quantum Computer in a Classical World

Daniel Gottesman (Chapter 6).

May 7, 2024

http://www.cs.umd.edu/class/spring2024/cmsc858G/QECCbook-2024-ch1-15.pdf

Clifford group

Maris Ozols

July 31, 2008

http://home.lu.lv/~sd20008/papers/essays/Clifford%20group%20%5Bpaper%5D.pdf

- Recall Pauli operators on n qubits: $\mathcal{P}_n = \{I, X, Y, Z\}^{\otimes n} \rightarrow 4^n$

  $\rightarrow$ Let $\mathcal{P}_n^* = \mathcal{P}_n \setminus \{I^{\otimes n}\}$

- Define the **n-qubit Clifford group** as

mod out global phase

$$\mathcal{C}_n := \{ U : P \in \pm \mathcal{P}_n^* \Rightarrow UPU^\dagger \in \pm \mathcal{P}_n^* \} / U(1)$$

$\rightarrow$ Transforms Paulis to Paulis! (up to sign)

⊛ There are other ways to define it! (See Gottesman book).

- Examples: Consider the Hadamard gate $H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$

$$HXH = Z, \quad HYH = -Y, \quad HZH = X$$

$\Rightarrow$ $H \in \mathcal{C}_1$!

$S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix} \rightarrow \quad SXS^\dagger = Y, \quad SYS^\dagger = -X, \quad SZS^\dagger = Z$

$\Rightarrow$ $S \in \mathcal{C}_1$!

$$CNOT = \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & 0 & 1 \\ & & 1 & 0 \end{pmatrix} = |0\rangle\langle 0| \otimes \mathbb{1} + |1\rangle\langle 1| \otimes X$$

$X \otimes I \mapsto X \otimes X$

$Z \otimes I \mapsto Z \otimes I$

$I \otimes X \mapsto I \otimes X$

$I \otimes Z \mapsto Z \otimes Z$

$$Z \otimes X = (Z \otimes I)(I \otimes X) \mapsto CNOT(Z \otimes I)(I \otimes X)CNOT$$

$$= \underbrace{CNOT(Z \otimes I)CNOT}_{Z \otimes I}\underbrace{CNOT(I \otimes X)CNOT}_{I \otimes X} \quad (CNOT\,CNOT = \mathbb{1}.)$$

$$= Z \otimes X$$

$\Rightarrow$ <u>CNOT $\in \mathcal{C}_2$</u> !

- <u>Size of $\mathcal{C}_n$</u>:

n	1	2	3	4	5
$\lvert\mathcal{C}_n\rvert$	24	11520	92897280	12128668876800	25410822678459187200

$$|\mathcal{C}_n| = \prod_{j=1}^{n} 2(4^j - 1)4^j = 2^{n^2+2n} \prod_{j=1}^{n} (4^j - 1).$$

- <u>Theorem</u>: $\mathcal{C}_n = \langle H_i, S_i, CNOT_{i,j} : i,j \in \{1,2,\dots,n\} \rangle / U(1)$

⊛ H, S, CNOT are <u>generators</u> of the Clifford group!

⊛ (Take all possible products of H, S, CNOT, and ignore global phase.)

<u>Proof</u>: See Sec. 6.3 of Gottesman book (requires some additional math).

- <u>Theorem (Gottesman-Knill)</u>: Circuits consisting of only Clifford gates can be simulated classically in poly time.

  ⊛ In particular, Clifford gates are not universal!

PHYSICAL REVIEW A **70**, 052328 (2004)
arXiv:quant-ph/0406196

Improved simulation of stabilizer circuits

Scott Aaronson*
Computer Science Department, University of California, Berkeley, California 94720, USA

Daniel Gottesman†
Perimeter Institute, Waterloo, Canada N2L 2Y5
(Received 25 June 2004; published 30 November 2004)

- <u>Theorem</u>: Clifford gates + one single-qubit non-Clifford gate is universal!