The story so far

We have classical computers

binary in $\rightarrow$ $\boxed{\text{Boolean Logic}}$ $\rightarrow$ binary out

This can reproduce any mathematical function, and so solve any mathematical problem

But for some problems, like quantum simulation, they are really slow

Simulating n particles takes $O(e^{O(n)})$ gates (and time)

So we created the quantum simulator

$$|\psi\rangle \rightarrow \boxed{U|\psi\rangle} \rightarrow \text{measurement results}$$

This can simulate realistic Hamiltonians evolutions efficiently

poly(n) gates (and time)

This is a big breakthrough!

The quantum simulator is a computing device that can solve a mathematical problem more efficiently than a classical computer

Can it solve problems that are not related to simulation?

If so, can it do it faster than a classical computer?

These are the kind of questions we will now consider.

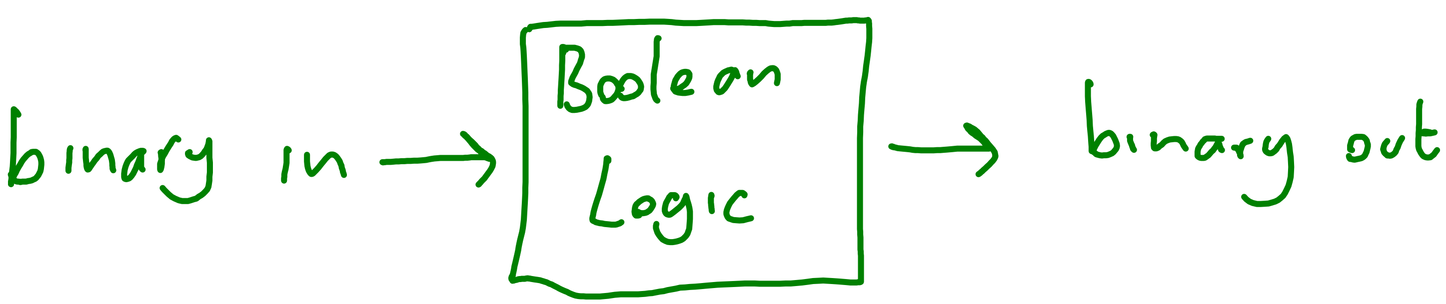
We are going to stop thinking of our device as a quantum simulator, and start thinking of it as a quantum computer.

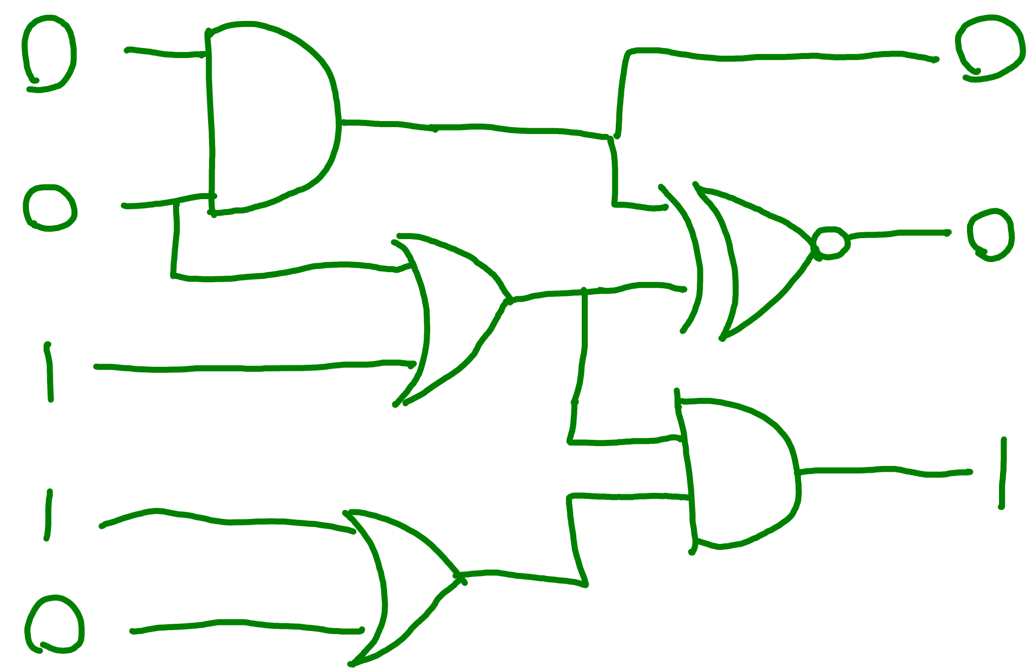
We will look at the 'models of quantum computation', which move us away from the concrete example of simulation, and towards a more abstract idea

The circuit model

The simplest and most widely used model of QC is the circuit model

It is based on the circuit model of classical computation





We have, basically, been using the circuit model so far:

applied to the case of quantum simulation

without the nice notation

So it needs little introduction

Let's return to our first simple model of quantum simulation

Numbers in (to specify $|\psi(0)\rangle$) $\rightarrow$ $\boxed{\text{maths } (H, t)}$ $\rightarrow$ numbers out (measurement results)

We gave the simulator numbers (to specify the initial state), it does some quantum stuff, and then gives us numbers (measurement results)

The numbers in can be considered to be binary

00110101101

These correspond directly to a multi qubit state

$|00110101101\rangle$

We assume that n qubit states can be prepared in O(n) time

We can consider this to be the input to our quantum circuit

Even if the actual computation does not act on states with this encoding, a rotation to the right encoding can be made. So no loss of generality

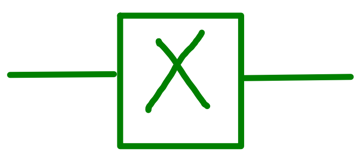
Once we have the input state, we act on it with quantum gates

Using only single qubit and controlled gates, we can do any unitary, so lets consider only these for now

Common unitaries have their own special notation

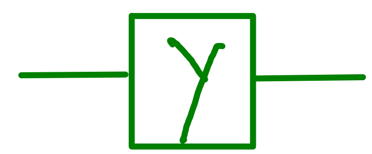
Pauli σ_x

$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$



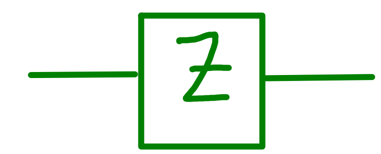
Pauli σ_y

$\begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$



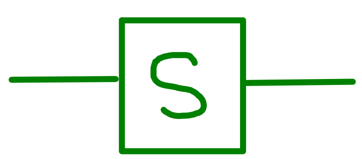
Pauli σ_z

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



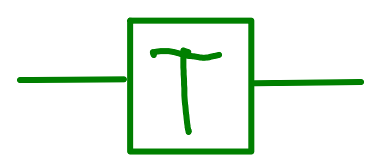
Phase

$\begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix} = (\sigma_z)^{1/2}$



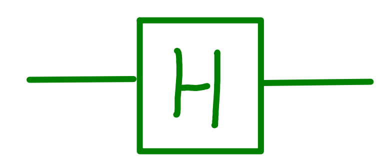
$\pi/8$

$\begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix} = (\sigma_z)^{1/4}$



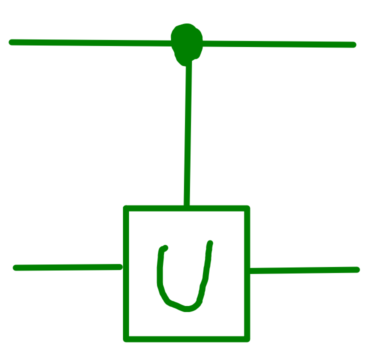
Hadamard

$\frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$



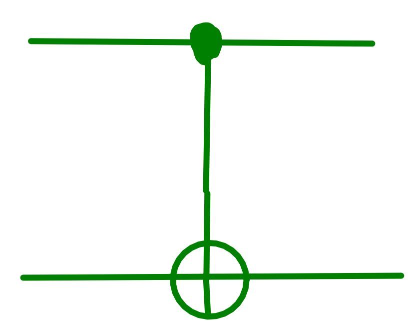
Controlled U

$|0\rangle\langle 0| \otimes \mathbb{1} + |1\rangle\langle 1| \otimes U$



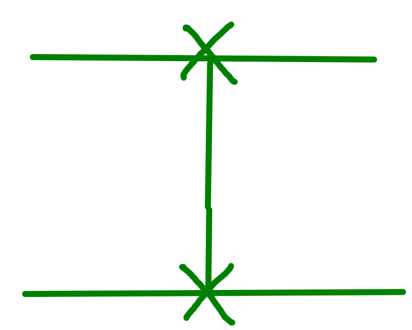
Controlled-NOT

$|0\rangle\langle 0| \otimes \mathbb{1} + |1\rangle\langle 1| \otimes \sigma_x$

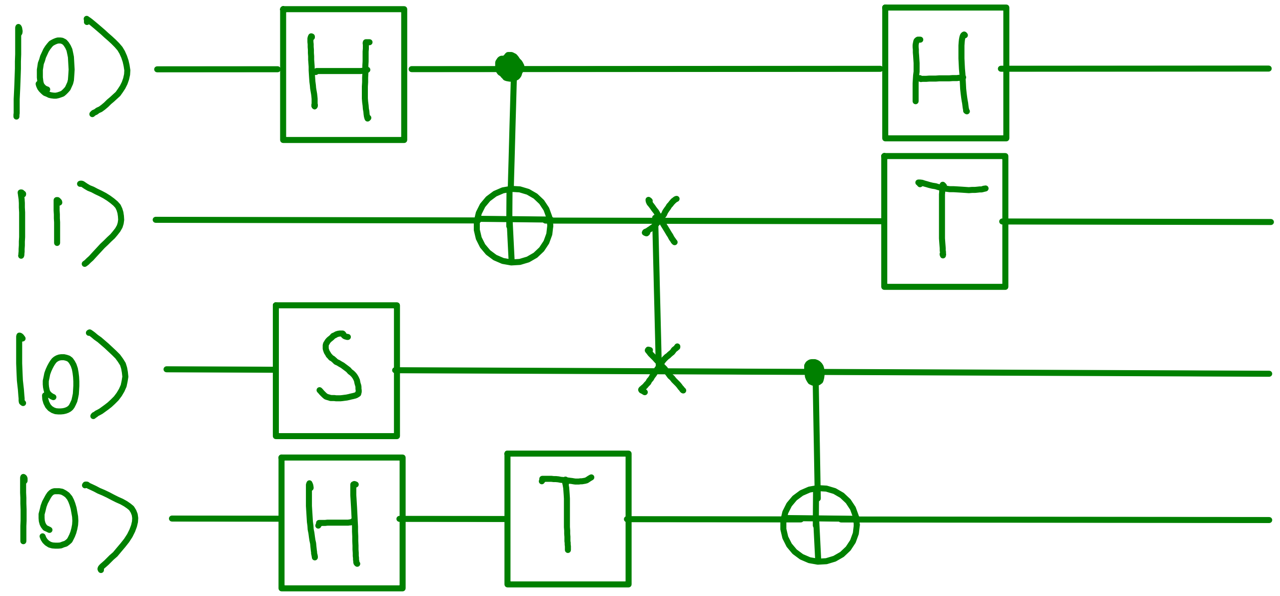


Swap

$\sum_{ij} |ij\rangle\langle ji|$

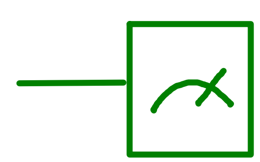


Any evolution on any initial state can then be represented by circuits of the form



The output is made using measurements. Without loss of generality, we can consider these to in the Z basis only

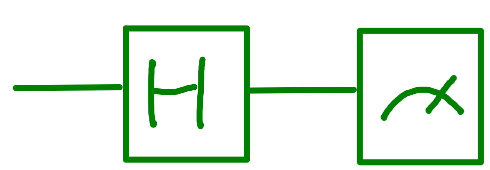
σ_z Measurement
$|0\rangle\langle 0|$, $|1\rangle\langle 1|$



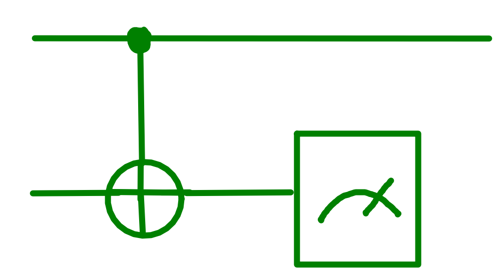
bit

Any other measurements (even multi qubit) can be made by rotating first

σ_x Measurement
$|+\rangle\langle +|$, $|-\rangle\langle -|$

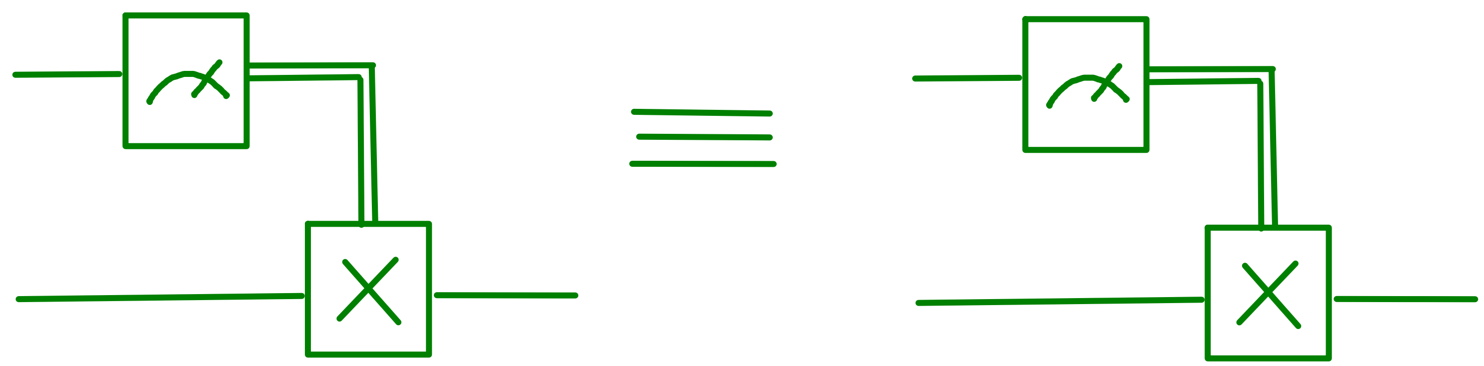


$\sigma_z \otimes \sigma_z$ measurement
$|00\rangle\langle 00| + |11\rangle\langle 11|$, $|01\rangle\langle 01| + |10\rangle\langle 10|$



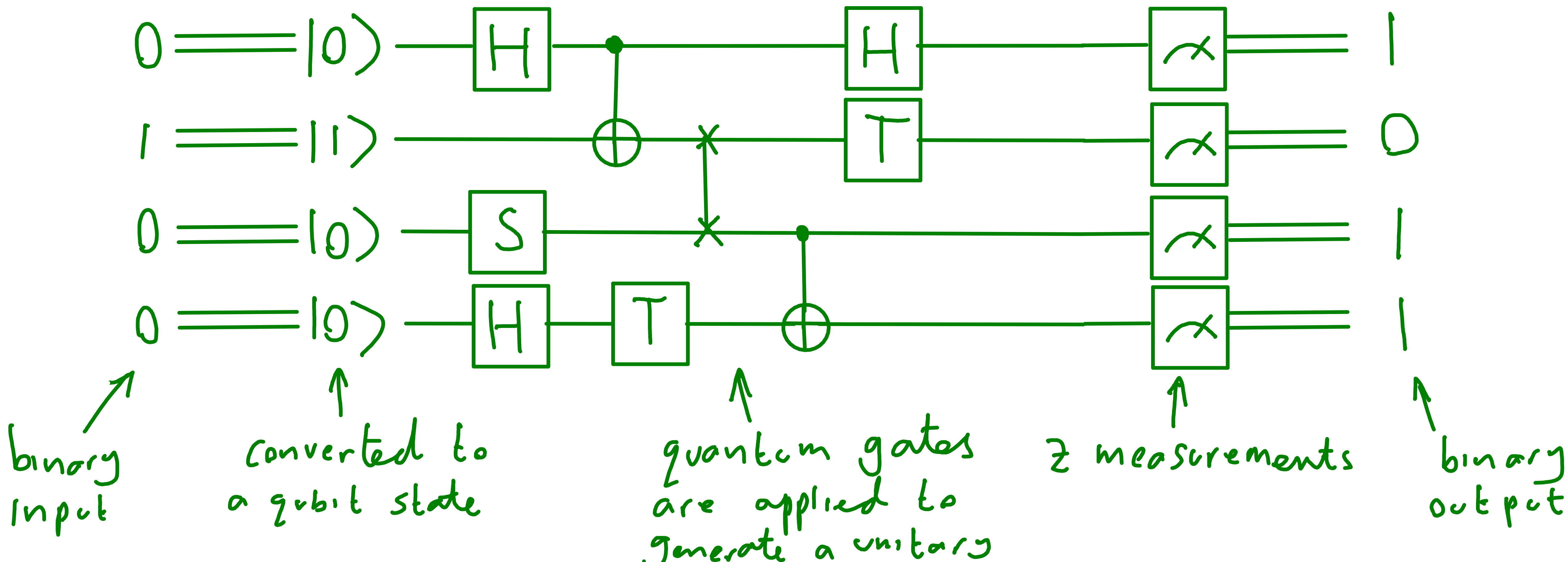
All measurement can be deferred to the end

Any operations that depend on measurement results can be performed using controlled operations



This is not always practical, but it makes our abstract model simpler

Any quantum computation is then of the form



This is a quantum computer, according to the circuit model

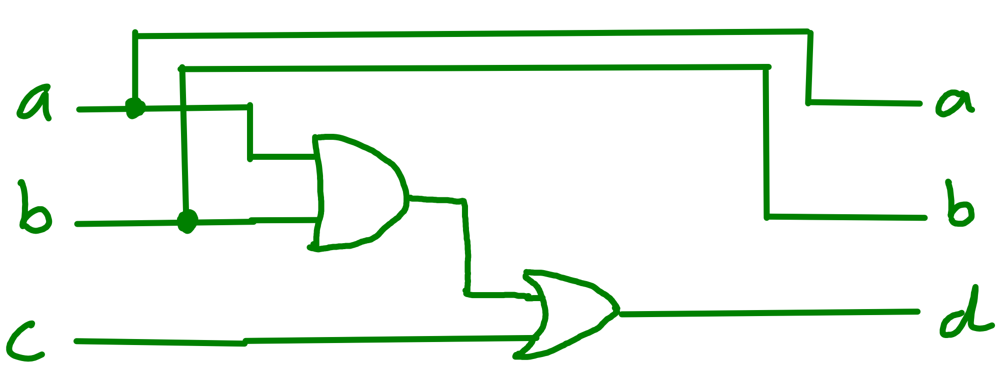
No reference to simulation. It just takes a number, makes it into a quantum state, applies gates, measures and outputs the results

What can such a device be used for?

We know it can be used as a quantum simulator

We can show that it can also reproduce any classical circuit with n Boolean gates using O(n) quantum gates
(but not vice-versa)

The reversible (N)AND

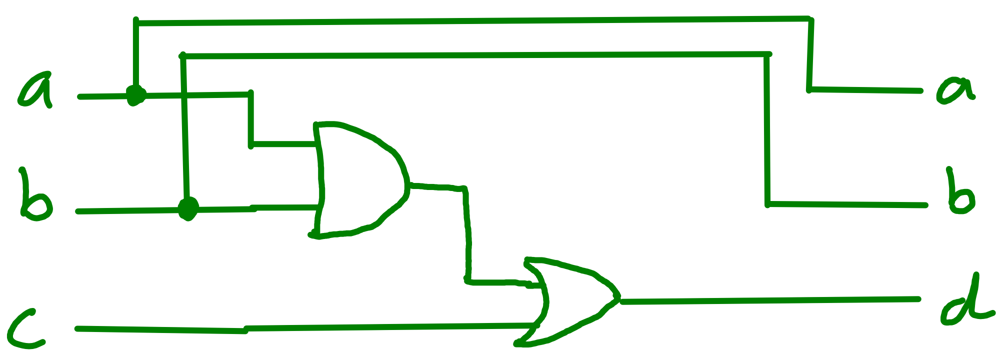


Performs AND or NAND depending on value of c. Initial a and b are also output

a	b	c	d	
0	0	0	0	AND
0	1	0	0	
1	0	0	0	
1	1	0	1	
0	0	1	1	NAND
0	1	1	1	
1	0	1	1	
1	1	1	0	

a and b are explicitly given in output. c can be determined by comparing d with a AND b.
∴ reversible

Since the NAND is universal for classical computation, so is the reversible (N)AND



$d = (a \text{ AND } b) \text{ XOR } c$

a	b	c	d	
0	0	0	0	AND
0	1	0	0	
1	0	0	0	
1	1	0	1	
0	0	1	1	NAND
0	1	1	1	
1	0	1	1	
1	1	1	0	

If we can produce a quantum circuit to simulate this gate, quantum circuits are also universal for classical computation

We can, since any reversible truth table can be easily turned into a unitary

$$U = \sum_{a,b,c} |a, b, (a \text{ AND } b) \text{ XOR } c\rangle\langle a, b, c|$$

And any unitary can be reproduced using one qubit rotations and controlled-NOTs

So quantum computers can be used for Mariokart!

The circuit model gives us a general framework in which to determine

What a quantum computer is

What it can do (and how it can do it)

How efficiently it can do it

In summary, it assumes:

1) Qubits are available
2) Computational basis states of n qubits can be prepared in O(n) time
3) Any of a certain set of unitary operations (gates) can be applied to any qubit as and when desired
4) Measurement can be made in the computational basis on any qubits as and when desired

These are then used to take a classical input using (2), process it with (3) and produce a classical output with (4)

Note that, for the circuit model, the program run by the quantum computer is equivalent to the unitary performed

In order for a quantum computer to be universal (to be able to run any possible program) it must therefore be able to generate any unitary

The set of gates that we are allowed to use is called universal if it can generate (or approximate to any degree) any unitary

We know, from our studies of simulation, that

arbitrary single qubit rotations + CNOT = universal

But, more simply, it's also true that

H + T + CNOT = universal

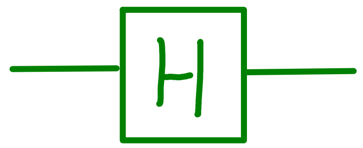
To prove this, it is sufficient to show that

$$H + T = \text{universal for single qubit}$$

Let's remind ourselves what these gates are

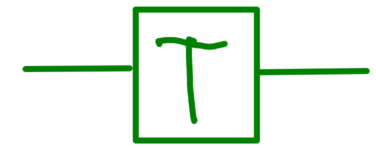
Hadamard

$$\frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$



$\pi/8$

$$\begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix} = (\sigma_z)^{1/4}$$



You've met the Hadamard before

$$U_{z\to x} = |+\rangle\langle 0| + |-\rangle\langle 1| = \frac{1}{\sqrt{2}}\left(|0\rangle\langle 0| + |1\rangle\langle 0|\right) + \frac{1}{\sqrt{2}}\left(|0\rangle\langle 1| - |1\rangle\langle 1|\right) = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} = H$$

$$\left(\text{note: } H = H^\dagger \ \therefore\ H = U_{z\to x} = U_{x\to z}\right)$$

Consider the similar transformation

$$U_{x\to y} = |\circlearrowright\rangle\langle +| + |\circlearrowleft\rangle\langle -| = \frac{1}{2}\left(|0\rangle\langle 0| + i|1\rangle\langle 0| + |0\rangle\langle 1| + i|1\rangle\langle 1|\right)$$

$$+ \frac{1}{2}\left(|0\rangle\langle 0| - i|1\rangle\langle 0| - |0\rangle\langle 1| + i|1\rangle\langle 1|\right) = \begin{pmatrix} 1 & \\ & i \end{pmatrix} = S = T^2$$

And note $U_{z\to y} = U_{x\to y} U_{z\to x}$, $U_{x\to y} = U_{y\to x}^\dagger$, etc

So H and T can map between all Pauli bases

Also, T gives us Pauli Z's

$$T^4 = \begin{pmatrix} 1 & \\ & e^{i\pi/4} \end{pmatrix}^4 = \begin{pmatrix} 1 & \\ & e^{i\pi} \end{pmatrix} = \begin{pmatrix} 1 & \\ & -1 \end{pmatrix} = \sigma_z$$

$$\therefore \quad -\boxed{T}-\boxed{T}-\boxed{T}-\boxed{T}- \;=\; -\boxed{Z}-$$

Conjugation gives Pauli X and Pauli Y

$$H \sigma_z H = \sigma_x \qquad T^2 H \sigma_z H T^2 = \sigma_y$$

Already we can see that H and T can do a lot. But we've not got arbitrary rotations quite yet

To proceed, note that

irrelevant global phase

$$\sigma_z = \begin{pmatrix} 1 & \\ & -1 \end{pmatrix} \therefore -i(\pi/8)\sigma_z = \begin{pmatrix} -i\pi/8 & 0 \\ 0 & i\pi/8 \end{pmatrix} \therefore e^{-i(\pi/8)\sigma_z} = \begin{pmatrix} e^{-i\pi/8} & 0 \\ 0 & e^{i\pi/8} \end{pmatrix} = e^{-i\pi/8} \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix} = T$$

$$\therefore \quad T = e^{-i(\pi/8)\sigma_z}, \qquad H T H = e^{-i(\pi/8)\sigma_x}$$

So we can rotate around the Z and X axis by pi/8

What if we do both?

$$T\,H\,T\,H = e^{-i\frac{\pi}{8}\sigma_z} e^{-i\frac{\pi}{8}\sigma_x} = \left(\mathbb{1}\cos\frac{\pi}{8} - i\sigma_z \sin\frac{\pi}{8}\right)\left(\mathbb{1}\cos\frac{\pi}{8} - i\sigma_x \sin\frac{\pi}{8}\right)$$

$$= \mathbb{1}\cos^2\frac{\pi}{8} - i(\sigma_x+\sigma_z)\sin\frac{\pi}{8}\cos\frac{\pi}{8} - \sigma_z\sigma_x \sin^2\frac{\pi}{8}$$

$$= \mathbb{1}\cos^2\frac{\pi}{8} - i\left[(\sigma_x+\sigma_z)\cos\frac{\pi}{8} + \sigma_y \sin\frac{\pi}{8}\right]\sin\frac{\pi}{8}$$

Any single qubit rotation is a rotation by some angle about some axis, and so may be expressed

$$e^{-i\vec{u}\cdot\vec{\sigma}\frac{\theta}{2}} = \mathbb{1}\cos\frac{\theta}{2} - i\,\vec{u}\cdot\vec{\sigma}\sin\frac{\theta}{2}$$

$\vec{u}$ is the unit vector for the axis

$$\vec{\sigma} = \begin{pmatrix}\sigma_x \\ \sigma_y \\ \sigma_z\end{pmatrix}$$

By inspection, for the rotation THTH

$$\cos\frac{\theta}{2} = \cos^2\frac{\pi}{8} \qquad \vec{u} = \frac{1}{\sqrt{1+\cos^2\frac{\pi}{8}}}\begin{pmatrix}\cos\pi/8 \\ \sin\pi/8 \\ \cos\pi/8\end{pmatrix}$$

It can be shown that θ is an irrational multiple of 2π

Repeated rotations by an irrational angle can be used to approximate any angle to any degree of accuracy

We want to rotate around $\vec{u}$ by φ with an accuracy $\delta = \frac{2\pi}{N}$

(So for an angle between $\varphi - \delta$ and $\varphi + \delta$)

define $\theta_j = \theta_j \bmod 2\pi$ for $j \in \{1, \dots, M\}$

Since θ is irrational, $\theta_i \neq \theta_j$ for $i \neq j$

Split the interval $[0, 2\pi)$ into N intervals $[0, \frac{2\pi}{N})$, $[\frac{2\pi}{N}, 2\frac{2\pi}{N})$, ... $[N\text{-}1\frac{2\pi}{N}, 2\pi)$

If $M > N$, at least one interval must contain more than one θ_k

So there exist i and j such that $\underbrace{0 < |\theta_j - \theta_i|}_{\text{irrational } \theta} \overbrace{< \frac{2\pi}{N}}^{\text{Pigeonhole principle}} = \delta$

Note $\theta_j - \theta_i = \theta_{j-i}$ $\therefore$ $\theta_k < \delta$ where $k = j - i$, $j > i$

The sequence θ_{lk} then fills the interval $[0, 2\pi)$ such that

$\theta_{lk} - \theta_{(l-1)k} < \delta$

There $\therefore$ exists a value of l, and hence an $n = lk$, such that
$|\theta_n - \varphi| < \delta$

So $(THTH)^n$ approximates rotation around $\vec{u}$ by φ to any δ

This is just for rotation around a single axis. What about arbitrary rotations?

Since $H\sigma_x H = \sigma_z$, $H\sigma_z H = \sigma_x$, $H\sigma_y H = -\sigma_y$

then $H : \vec{u} = \frac{1}{\sqrt{1+\cos^2\frac{\pi}{8}}}\begin{pmatrix}\cos\pi/8 \\ \sin\pi/8 \\ \cos\pi/8\end{pmatrix} \rightarrow \vec{v} = \frac{1}{\sqrt{1+\cos^2\frac{\pi}{8}}}\begin{pmatrix}\cos\pi/8 \\ -\sin\pi/8 \\ \cos\pi/8\end{pmatrix}$

so $H(THTH)^n H$ gives arbitrary rotations around an axis that is not parallel to $\vec{u}$

Any single qubit unitary can then be expressed (exercises)

$U = R_{\vec{u}}(\alpha)\, R_{\vec{v}}(\beta)\, R_{\vec{u}}(\delta)$, $R_{\vec{u}}(\alpha) = e^{-i\vec{u}\cdot\vec{\sigma}\alpha/2}$ etc

Since we can expect the angles to be distributed uniformly, we can expect $n = O(1/\delta)$

So the method is efficient

But this trick isn't the only way to rotate, for example we would use

$$T^4 = Z$$

Rather than

$$(THTH)^{n_\alpha} H (THTH)^{n_\beta} H (THTH)^{n_\gamma} \qquad n_\alpha, n_\beta, n_\gamma = O(1/\delta)$$

Using such tricks, the Solvay Kitaev theorem shows that the number of H's and T's used to approximate any single qubit unitary is

$$O(\log^c(1/\delta)), \quad c \approx 2$$

Which is much faster!