Quantum Computing Crash Course + VQE

Pranav Gokhale

ATPESC 2018

#1: States as Vectors

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \qquad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

#1: States as Vectors

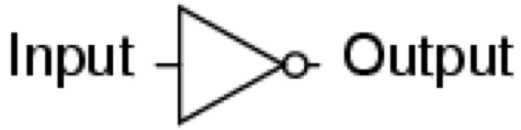
$$\begin{pmatrix} |00\rangle \\ |01\rangle \\ |10\rangle \\ |11\rangle \end{pmatrix}$$

#1: States as Vectors

$$|00\rangle = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \qquad |01\rangle = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} \qquad |10\rangle = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} \qquad |11\rangle = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

#2: Gates as Matrices

NOT gate truth table



Input	Output
0	1
1	0

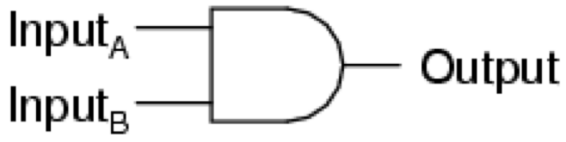
$$\text{NOT}\begin{pmatrix}1\\0\end{pmatrix} = \begin{pmatrix}0\\1\end{pmatrix}$$

$$\text{NOT}\begin{pmatrix}0\\1\end{pmatrix} = \begin{pmatrix}1\\0\end{pmatrix}$$

$$\text{NOT} = \begin{pmatrix}0 & 1\\1 & 0\end{pmatrix}$$

#2: Gates as Matrices

2-input AND gate



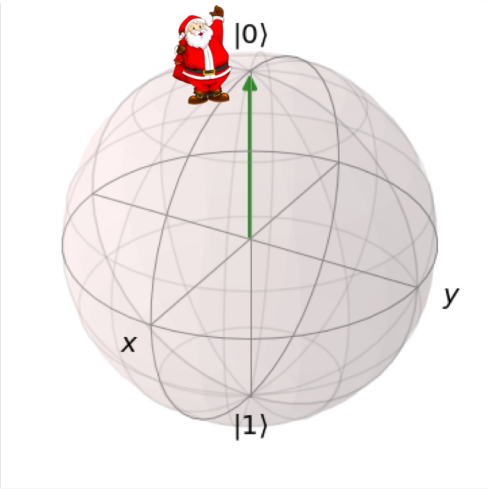
A	B	Output
0	0	0
0	1	0
1	0	0
1	1	1

$$\text{AND}\begin{pmatrix}1\\0\\0\\0\end{pmatrix}=\begin{pmatrix}1\\0\end{pmatrix} \qquad \text{AND}\begin{pmatrix}0\\0\\1\\0\end{pmatrix}=\begin{pmatrix}1\\0\end{pmatrix}$$

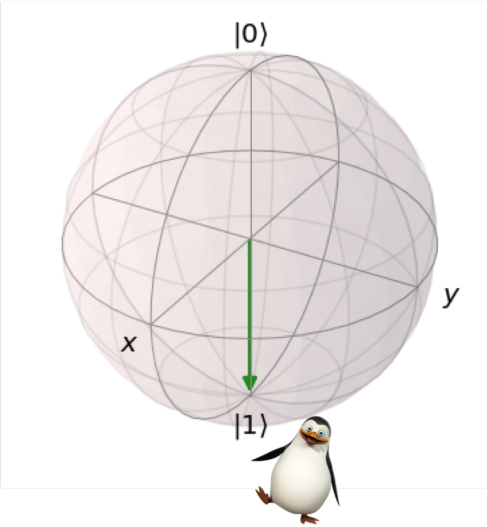
$$\text{AND}\begin{pmatrix}0\\1\\0\\0\end{pmatrix}=\begin{pmatrix}1\\0\end{pmatrix} \qquad \text{AND}\begin{pmatrix}0\\0\\0\\1\end{pmatrix}=\begin{pmatrix}0\\1\end{pmatrix}$$

$$\rightarrow \quad \text{AND}=\begin{pmatrix}1&1&1&0\\0&0&0&1\end{pmatrix}$$

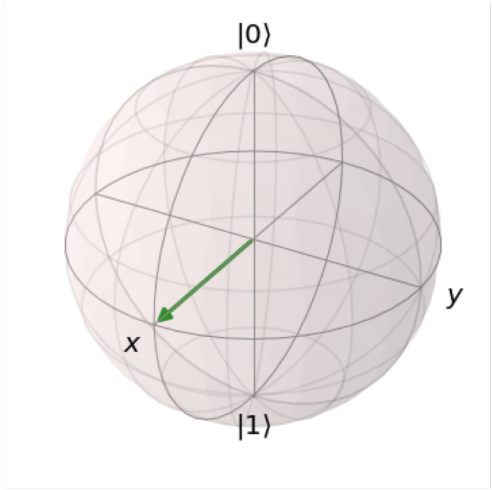
#3: Quantum States



$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$



$$|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$



$$\frac{|0\rangle + |1\rangle}{\sqrt{2}} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

50% probability of measuring 0
50% probability of measuring 1

#4: Superposition

$$(|0\rangle + |1\rangle)(|0\rangle + |1\rangle) = |00\rangle + |01\rangle + |10\rangle + |11\rangle = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

$$(|0\rangle + |1\rangle)(|0\rangle + |1\rangle)(|0\rangle + |1\rangle) = \begin{matrix} |000\rangle + |001\rangle + |010\rangle + |011\rangle + \\ |100\rangle + |101\rangle + |110\rangle + |111\rangle \end{matrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

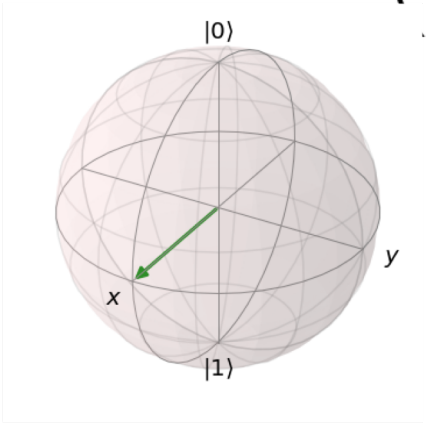
(Normalization factors omitted)

#5: Quantum Gates (Single Qubit)

$$R_x(\theta) = \begin{pmatrix} \cos\frac{\theta}{2} & -i\sin\frac{\theta}{2} \\ -i\sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{pmatrix}$$

$$R_y(\theta) = \begin{pmatrix} \cos\frac{\theta}{2} & -\sin\frac{\theta}{2} \\ \sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{pmatrix}$$

$$R_z(\theta) = \begin{pmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{pmatrix}$$



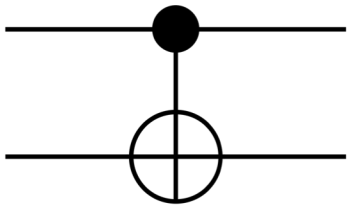
$$\mathrm{H} = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

$$\mathrm{H}\,|0\rangle = \mathrm{H}\begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

$$\mathrm{H}\,|1\rangle = \mathrm{H}\begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$$

(Normalization factors omitted)

#5: Quantum Gates (Two Qubit)



$$\mathrm{CNOT}\,|0x\rangle = |0x\rangle$$

$$\mathrm{CNOT}\,|1x\rangle = |1\bar{x}\rangle$$

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

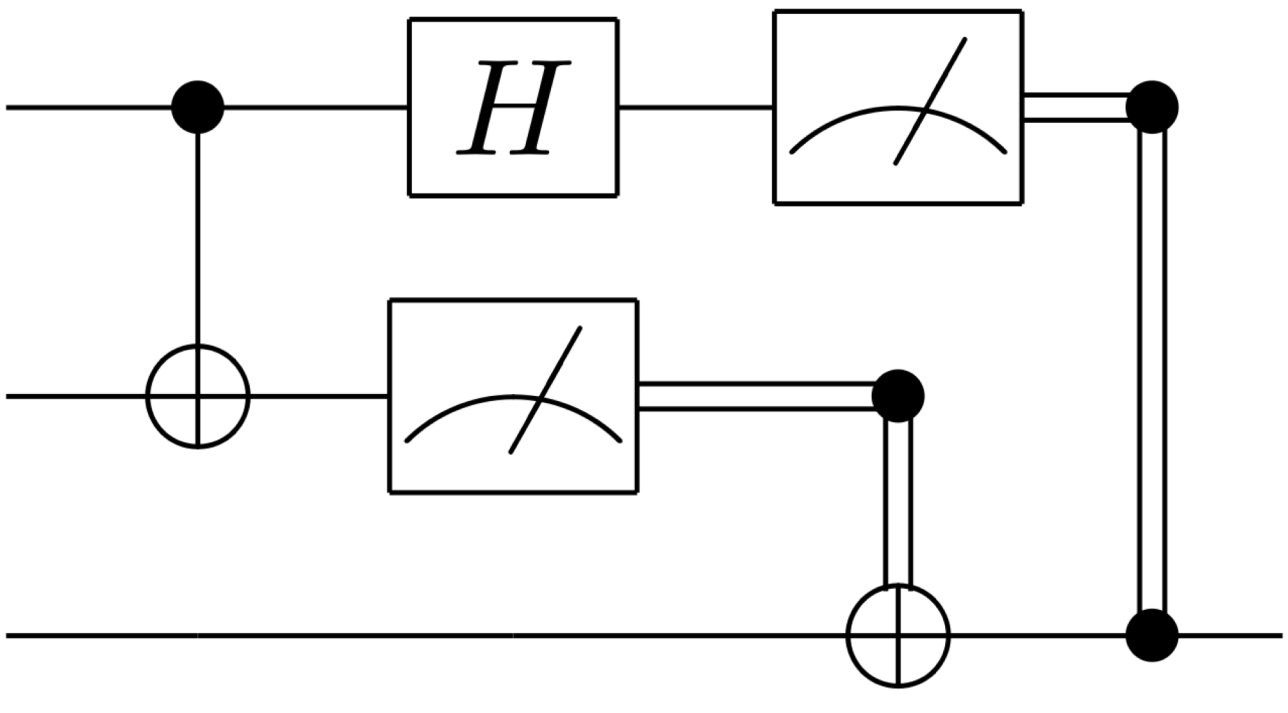
$$\mathrm{CNOT}\underbrace{(|0\rangle + |1\rangle)\,|0\rangle}_{\text{Separable State}} = \mathrm{CNOT}\begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \end{pmatrix} = \underbrace{|00\rangle + |11\rangle}_{\text{Entangled State}}$$

(Normalization factors omitted)

#6: Example: Quantum Teleportation

$$a\,|0\rangle + b\,|1\rangle = \begin{pmatrix} a \\ b \end{pmatrix}$$

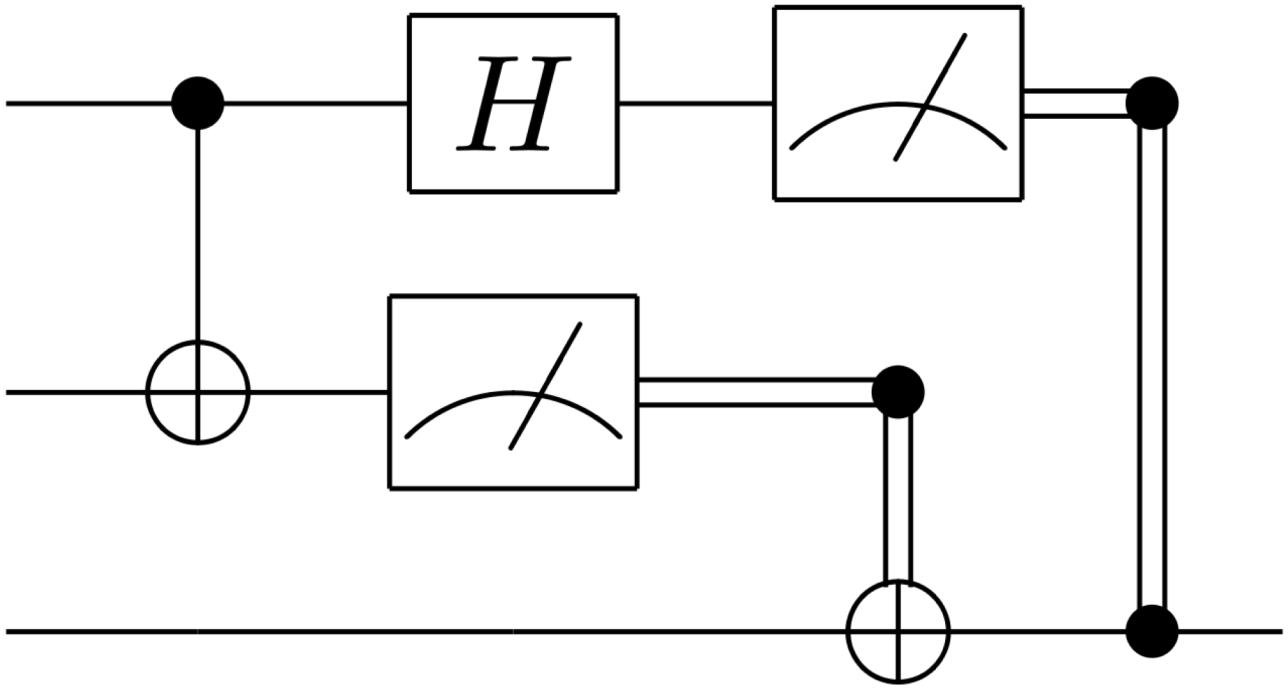
$$|00\rangle + |11\rangle = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$



(Normalization factors omitted)

#6: Example: Quantum Teleportation

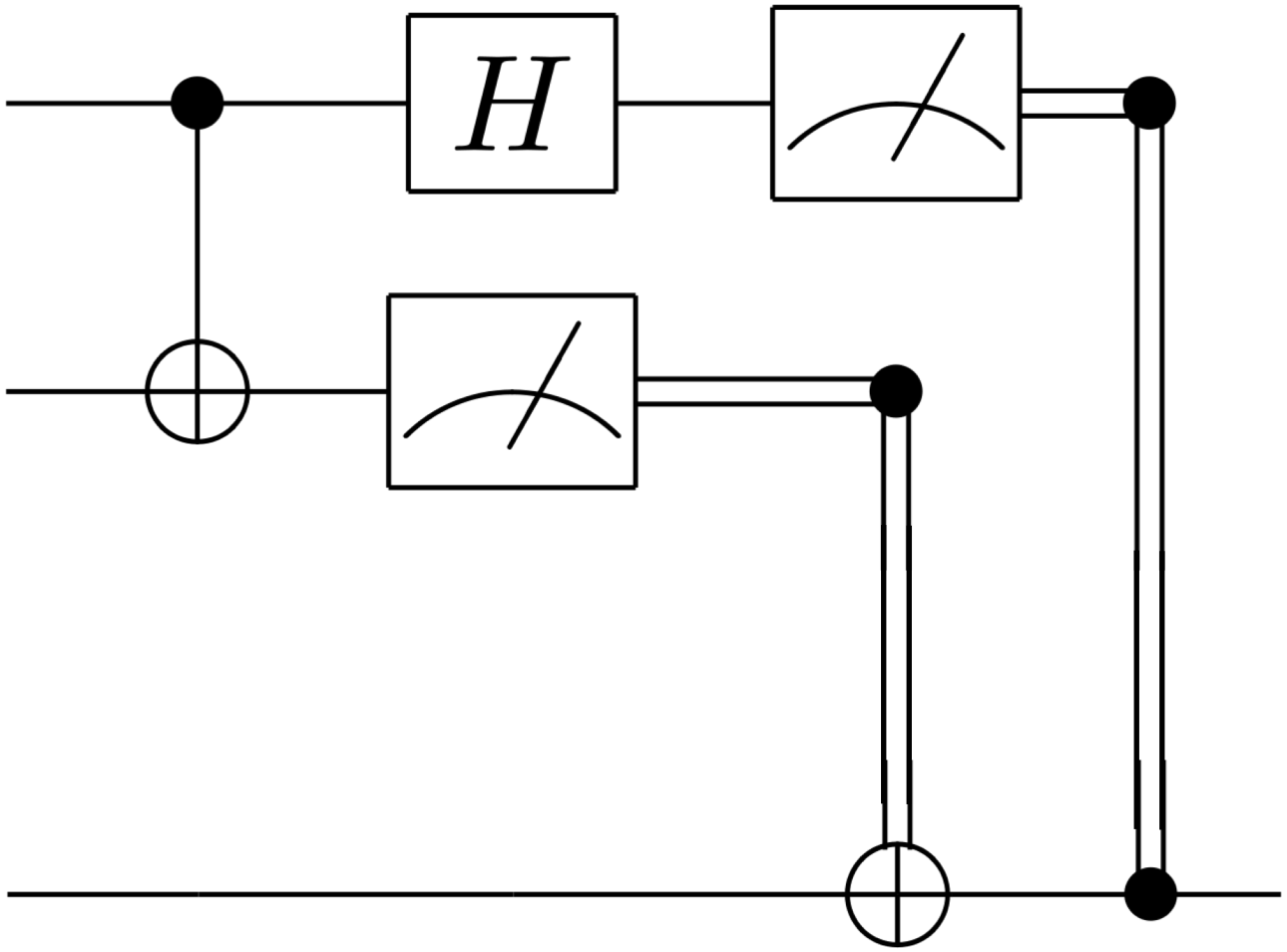
$$a\,|000\rangle + a\,|011\rangle +$$
$$b\,|100\rangle + b\,|111\rangle$$



(Normalization factors omitted)

#6: Example: Quantum Teleportation

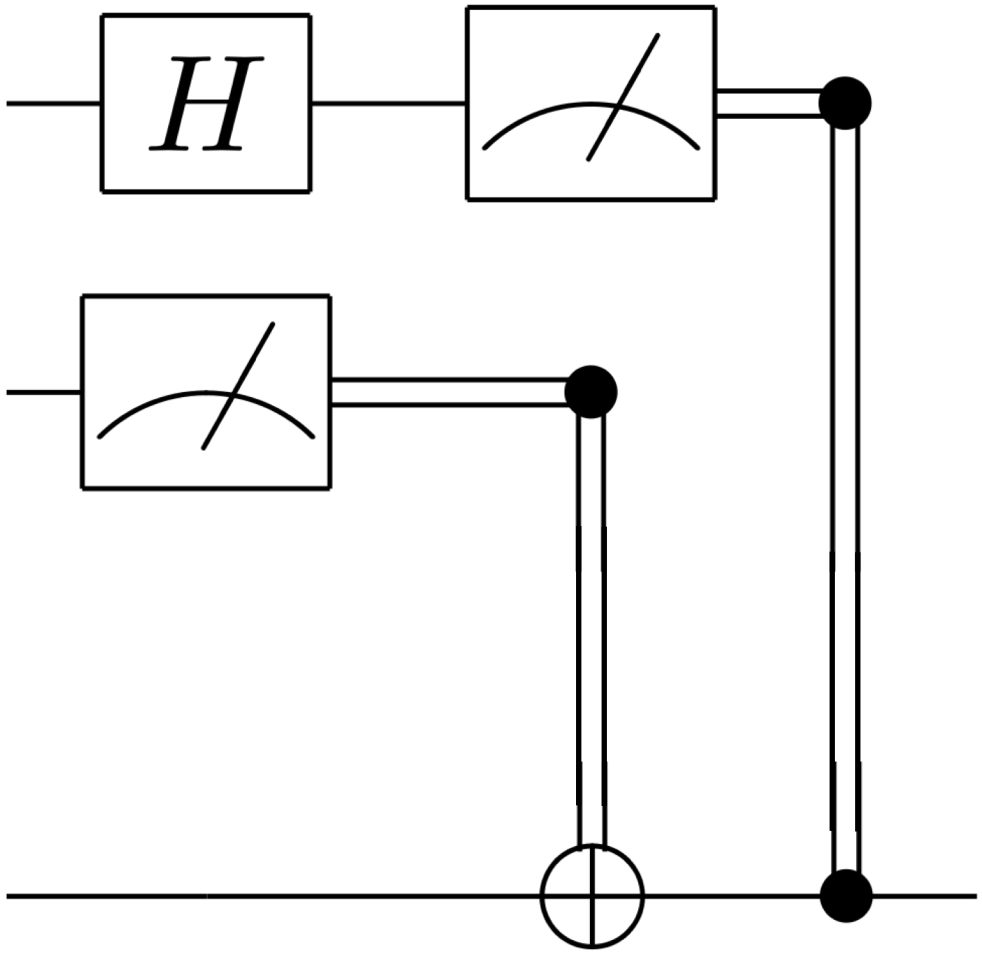
$$a\,|000\rangle + a\,|011\rangle +$$
$$b\,|100\rangle + b\,|111\rangle$$



(Normalization factors omitted)

#6: Example: Quantum Teleportation

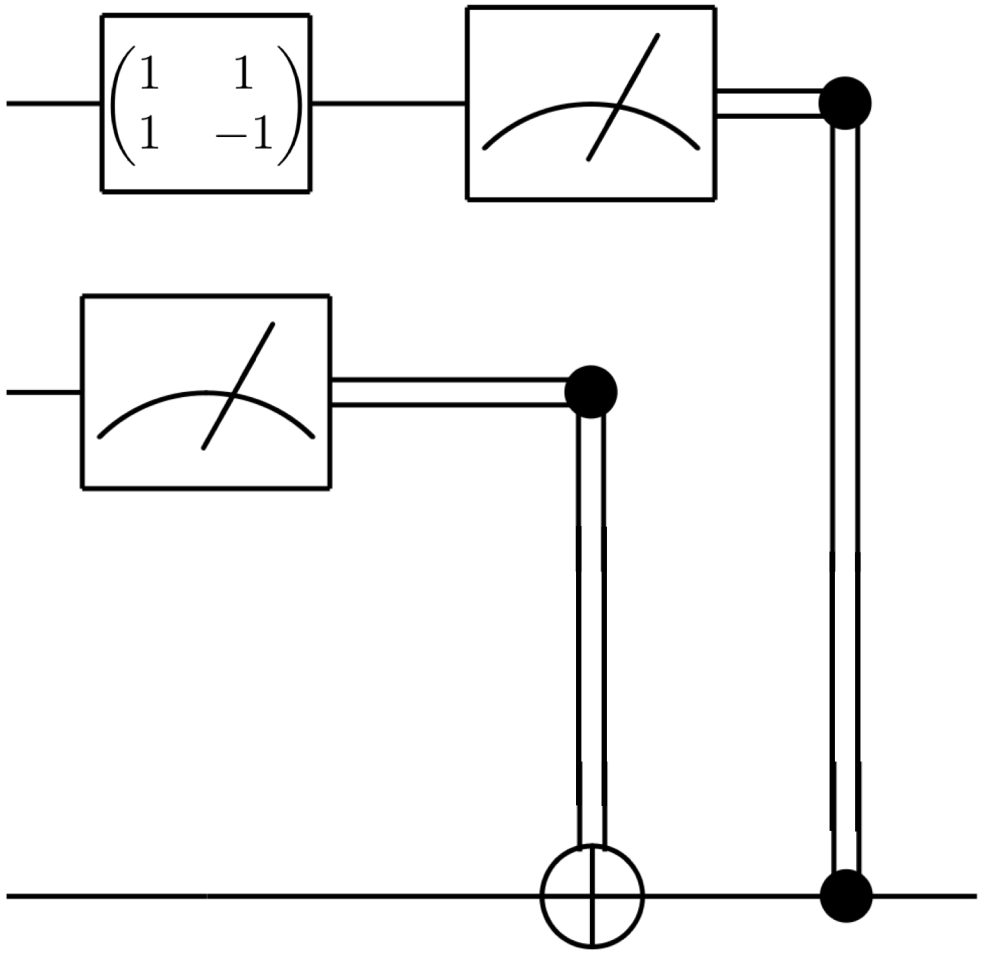
$$a\,|000\rangle + a\,|011\rangle + b\,|110\rangle + b\,|101\rangle$$



(Normalization factors omitted)

#6: Example: Quantum Teleportation

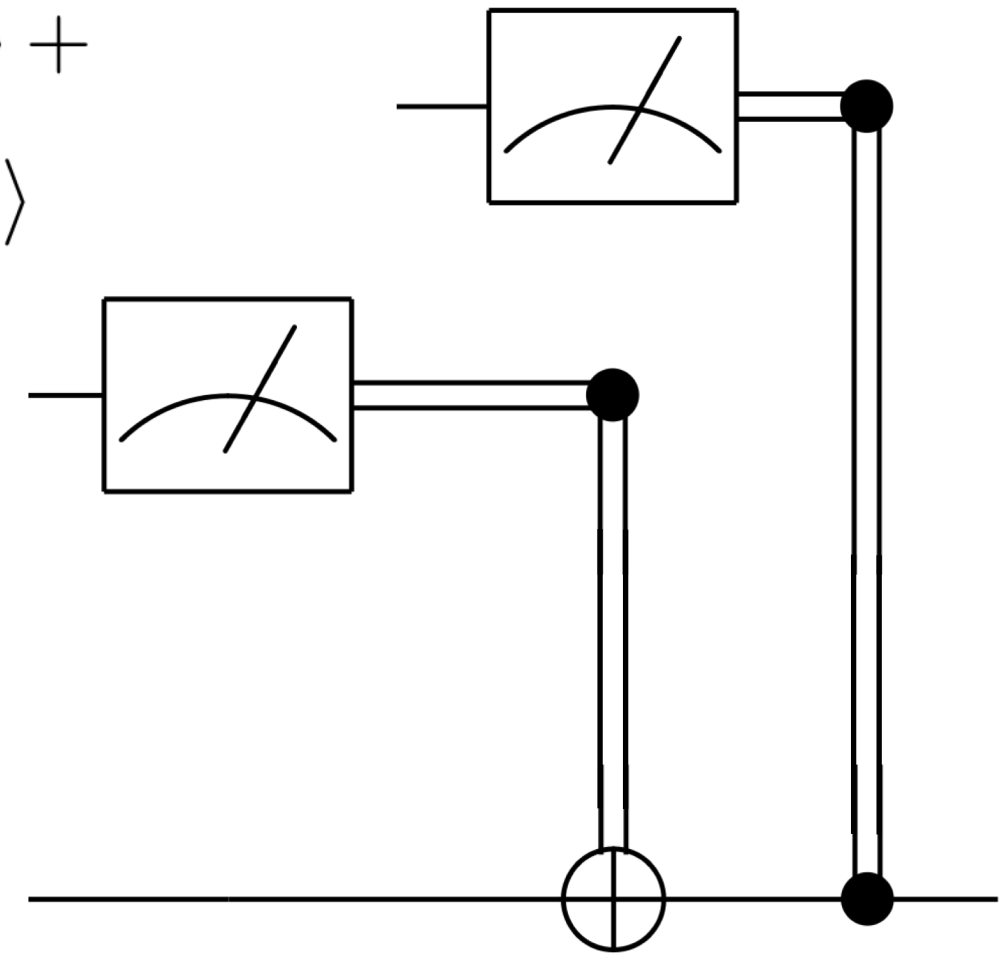
$$a\,|000\rangle + a\,|011\rangle + b\,|110\rangle + b\,|101\rangle$$



(Normalization factors omitted)

#6: Example: Quantum Teleportation

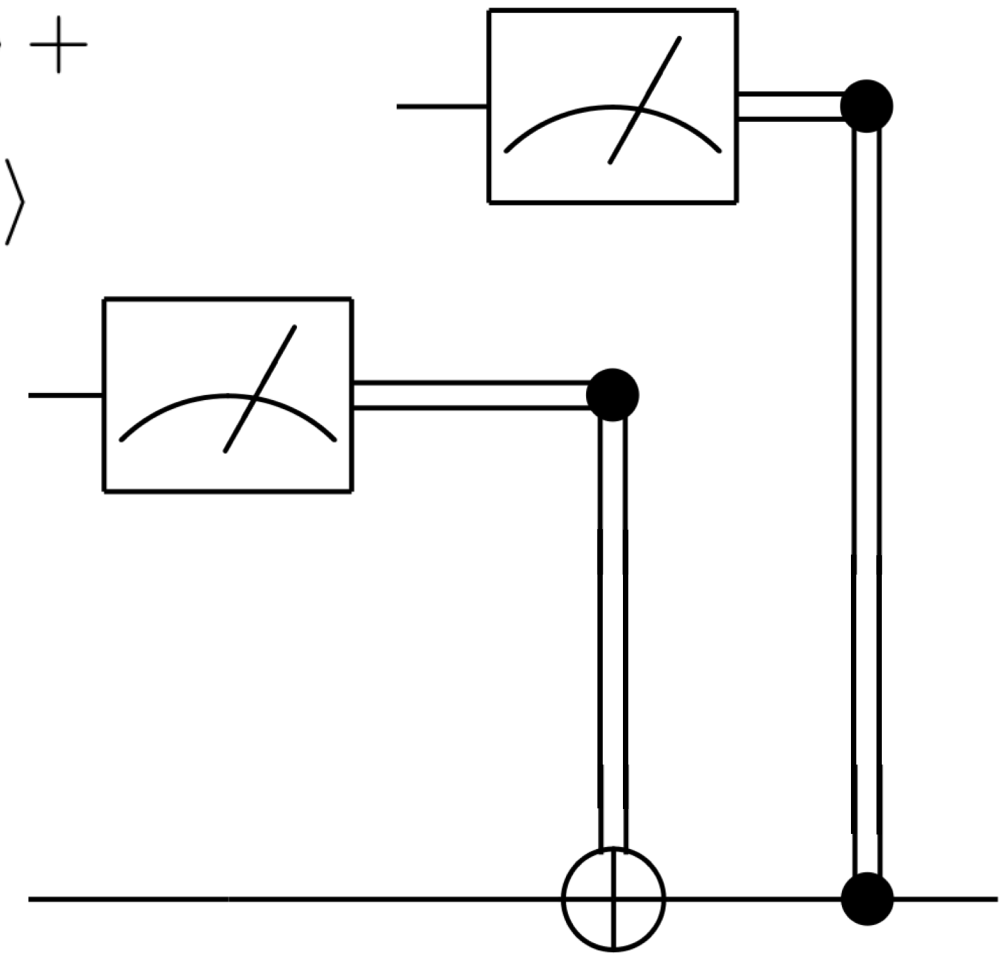
$$a\,|000\rangle + a\,|100\rangle + a\,|011\rangle + a\,|111\rangle +$$
$$b\,|010\rangle - b\,|110\rangle + b\,|001\rangle - b\,|101\rangle$$



(Normalization factors omitted)

#6: Example: Quantum Teleportation

$$a\,|000\rangle + a\,|100\rangle + a\,|011\rangle + a\,|111\rangle +$$
$$b\,|010\rangle - b\,|110\rangle + b\,|001\rangle - b\,|101\rangle$$

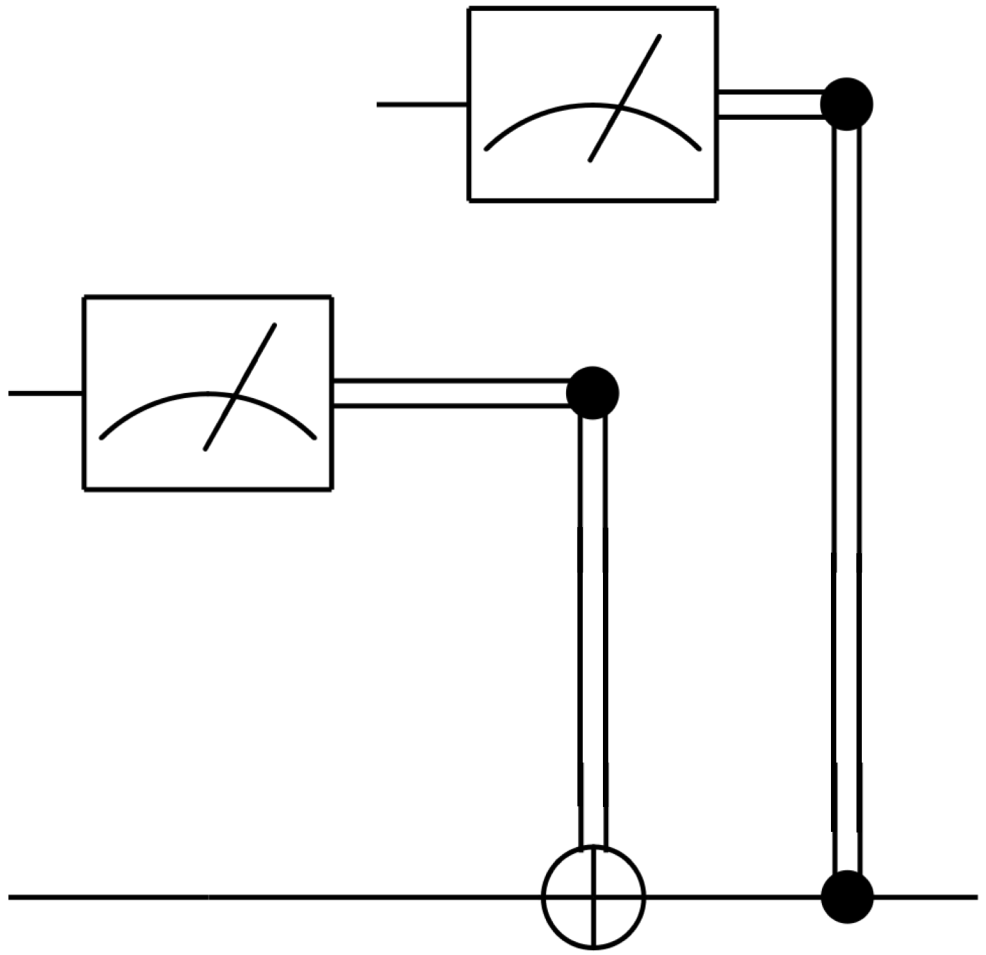


(Normalization factors omitted)

#6: Example: Quantum Teleportation

$$a\,|000\rangle +$$

$$b\,|001\rangle$$

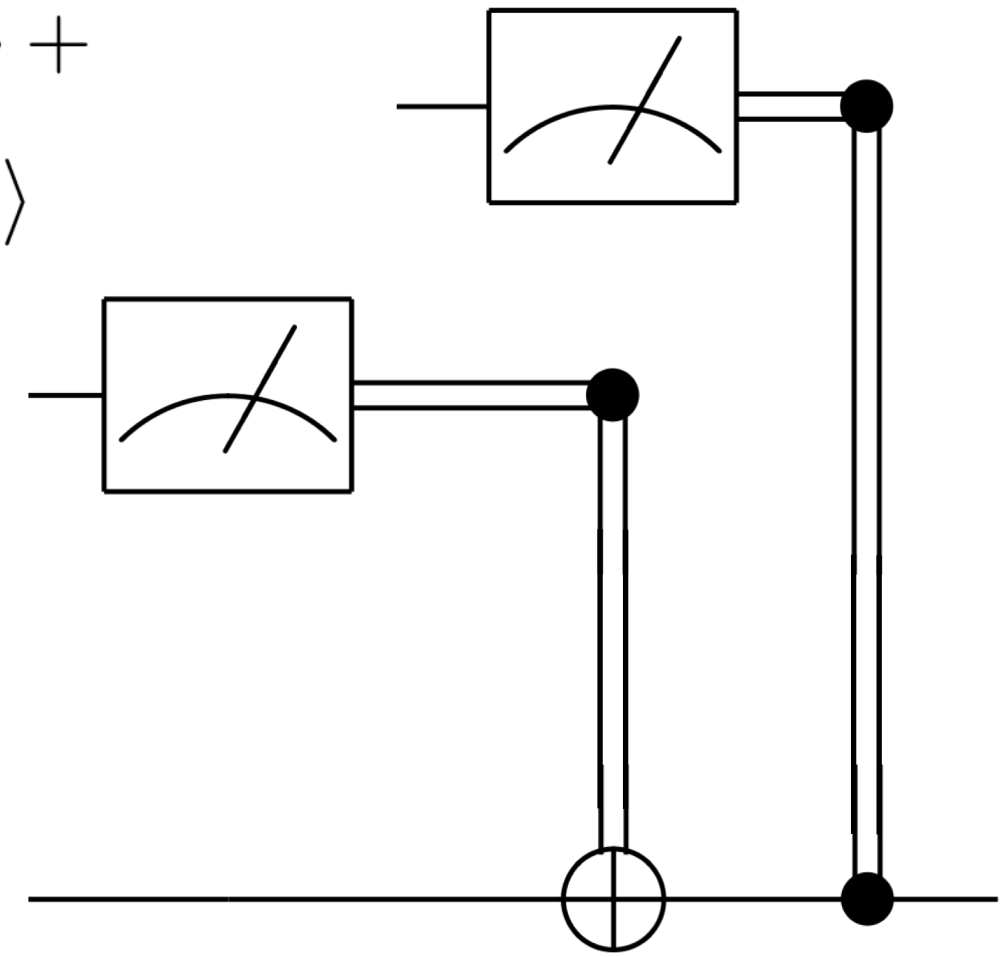


$$= |00\rangle\,(a\,|0\rangle + b\,|1\rangle)$$

(Normalization factors omitted)

#6: Example: Quantum Teleportation

$$a\,|000\rangle + a\,|100\rangle + a\,|011\rangle + a\,|111\rangle +$$
$$b\,|010\rangle - b\,|110\rangle + b\,|001\rangle - b\,|101\rangle$$

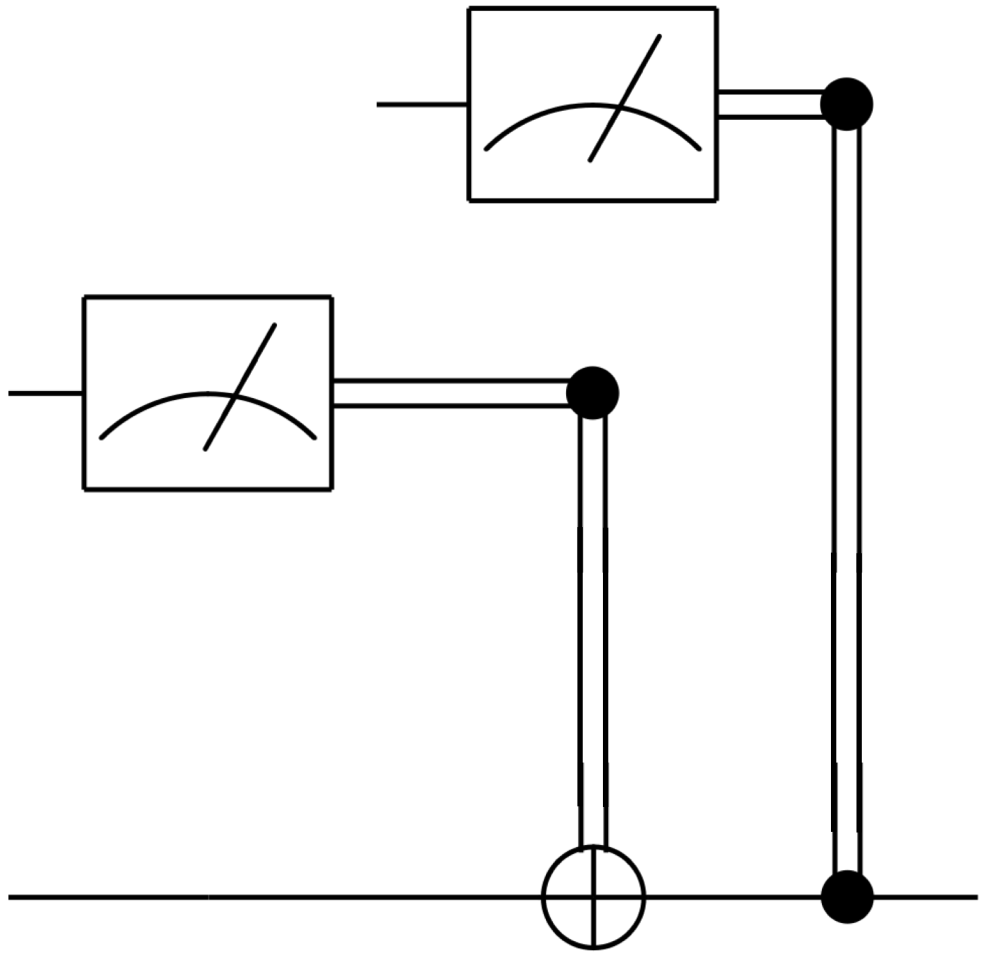


(Normalization factors omitted)

#6: Example: Quantum Teleportation

$$a\,|011\rangle +$$

$$b\,|010\rangle$$



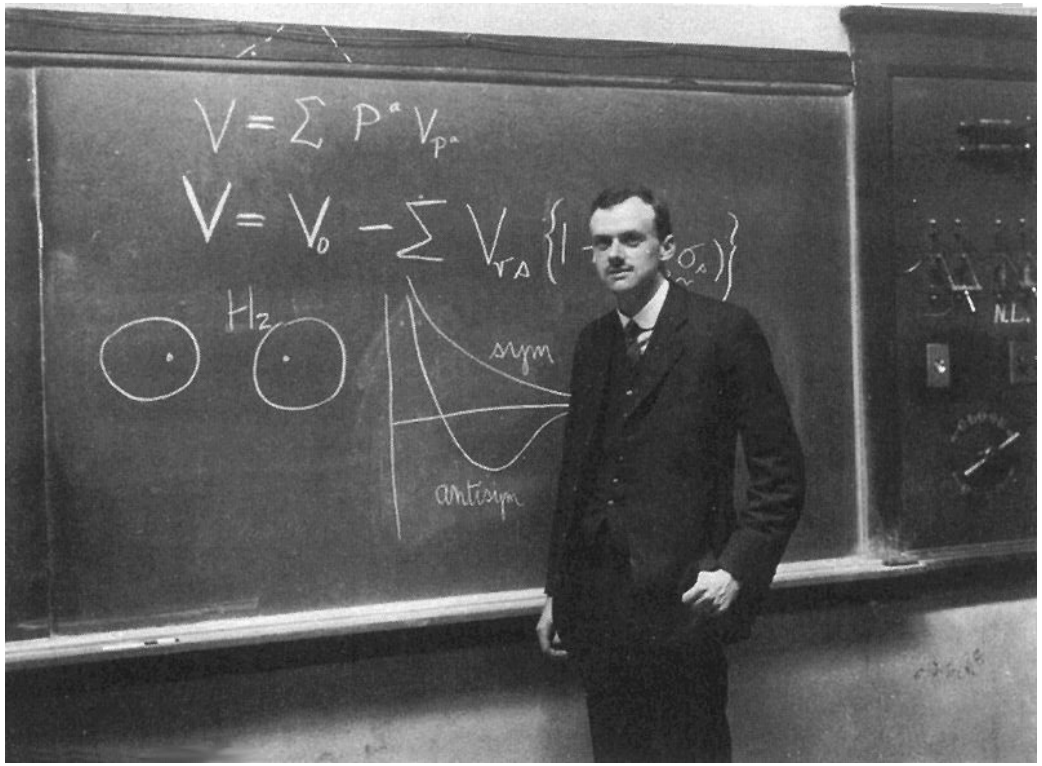
$$= |01\rangle\,(a\,|1\rangle + b\,|0\rangle)$$

(Normalization factors omitted)

Summary

- States are represented by vectors, gates are matrices
- Quantum states can be in superposition
  - Gives rise to potential quantum speedup. N qubits occupy 2^N state space.
- CNOT gate is an entangling gate

Chemistry Applications



The underlying physical laws necessary for the mathematical theory of a large part of physics and the whole of chemistry are thus completely known, and the difficulty is only that the exact application of these laws leads to equations much too complicated to be soluble.

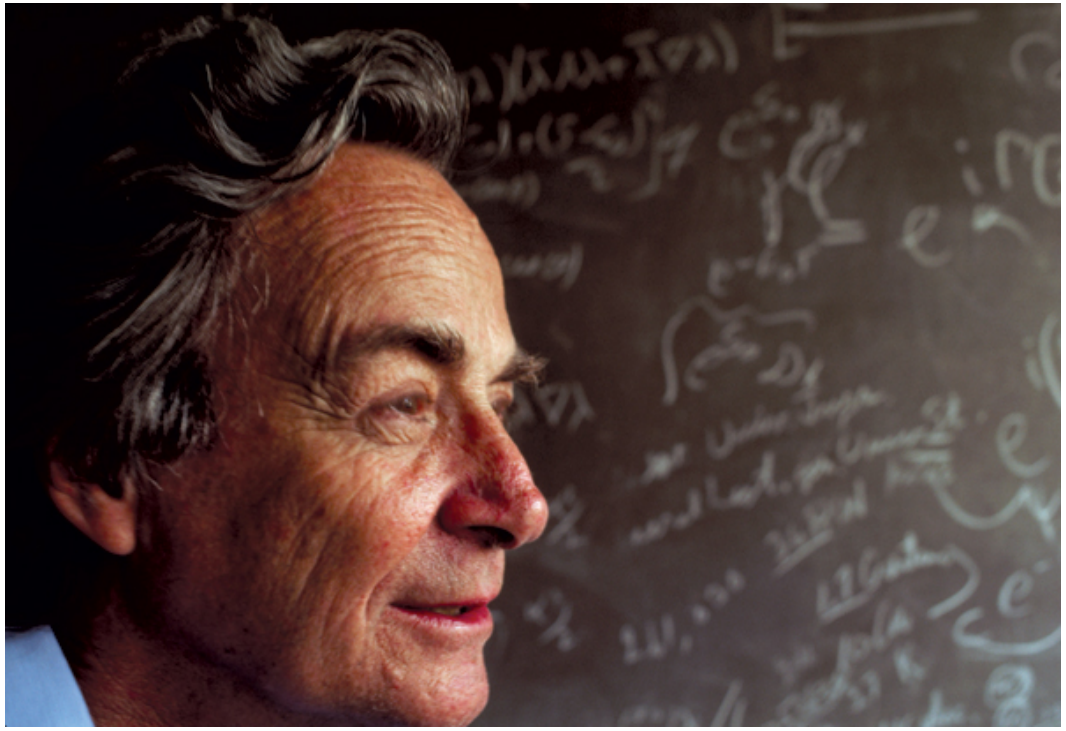
P.A.M. Dirac, Proc. R. Soc. A **123**, 714 (1929)

Content via Ken Brown (Duke)

Ground State Determination

- What assignment of electrons to “addresses” minimizes the energy?
- Equivalently: find lowest eigenvalue of Hamiltonian matrix
  - Electrons have kinetic energy
  - Electrons repel each other
  - Electrons are attracted to nuclei
- For n electrons & m addresses, m!/(n!)(m-n!) possible “classical” configurations

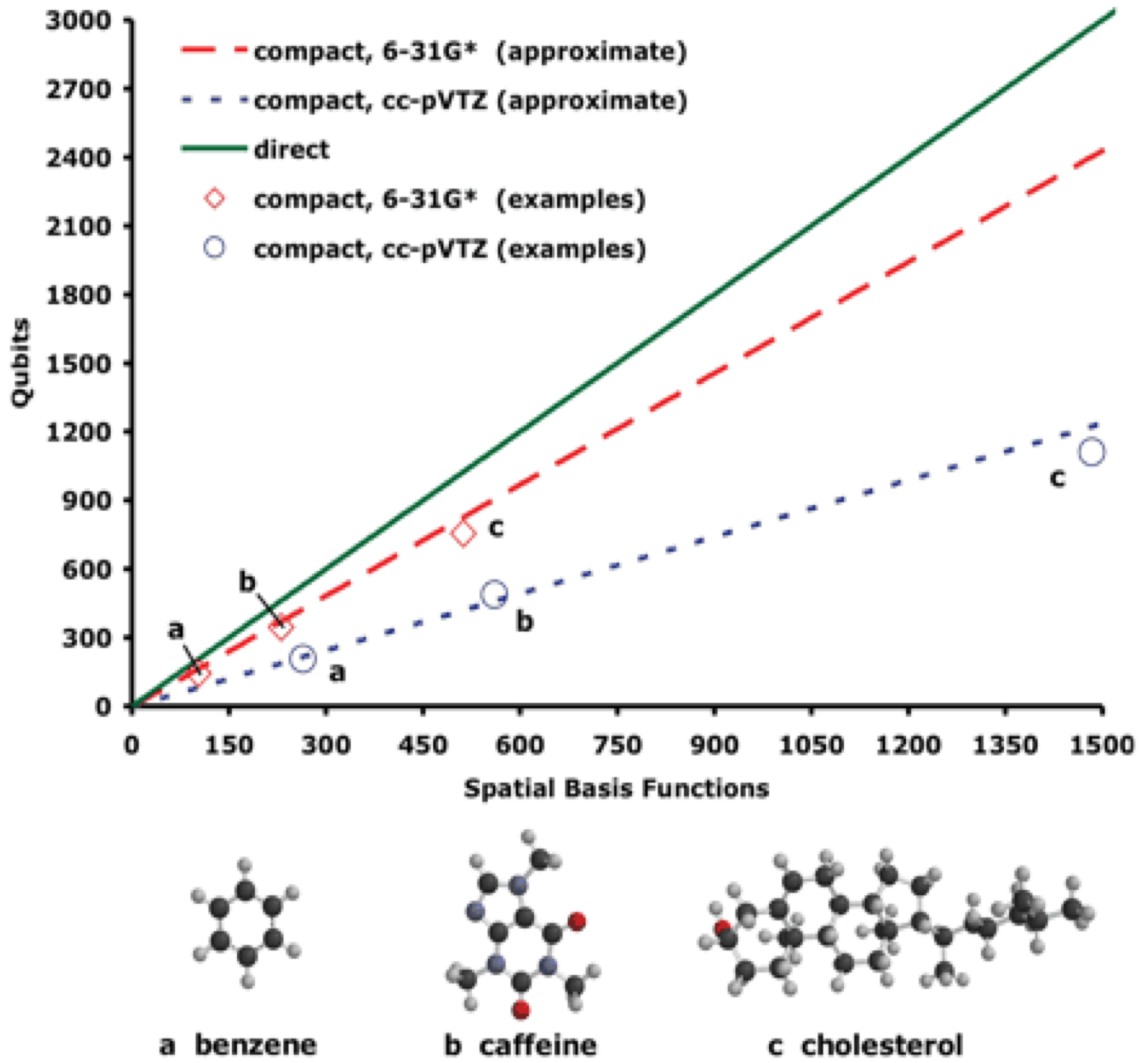
Disruptive Idea



"Let the computer itself be built of quantum mechanical elements which obey quantum mechanical laws."

R. Feynman, *Int. J. Theor. Phys.* **21**, 467 (1982)

Chemistry on QC is Tractable



D.S. Abrams and S. Lloyd,
Phys. Rev. Lett. **83**, 5162 (1999).

A. Aspuru-Guzik, A. Dutoi,
P. Love, and M. Head-Gordon
Science **309**, 1704 (2005).

VQE: Variational Quantum Eigensolver

- Guess, check, repeat

1. Guess ground state vector as $\Psi(\theta_1, \ldots, \theta_N)$ (“ansatz”)
2. Measure energy, $< H >$
3. Pass result to classical optimizer, pick next $\theta_1, \ldots, \theta_N$

- Demo: open VQE_Demo.ipnyb

VQE Demo: He-H+

Setup

Open VQE_Demo.ipnyb

In [ ]:
```
import Qconfig
from qiskit import register

token_sum = 0
for x in Qconfig.APItoken:
    token_sum = int(x, 16) + token_sum

if token_sum == 887:
    print('ISCA-2018 tutorial APItoken setup: Success')
    register(Qconfig.APItoken)
else:
    print('ISCA-2018 tutorial APItoken setup: Fail')
```

Background

Goal: find the lowest eigenvalue/vector of matrix H

Quantum Part

1. Prepare an ansatz (trial input qubit state), parametrized by N angles.
2. Measure $\langle H \rangle$ term-by-term

Classical Part

Choose new values for the N angles, with goal of minimizing $\langle H \rangle$.

Repeat

Demo: He-H+

Set H to Hamiltonian.

Lowest eigenvalue of H is **ground state energy**.

Demo: iteration of the quantum part.

Quantum Part

1. Prepare an ansatz (trial input qubit state), parametrized by N angles.

2. Measure $\langle H \rangle$ term-by-term

You guess angles!

Ansatz Preparation

Typical choices:

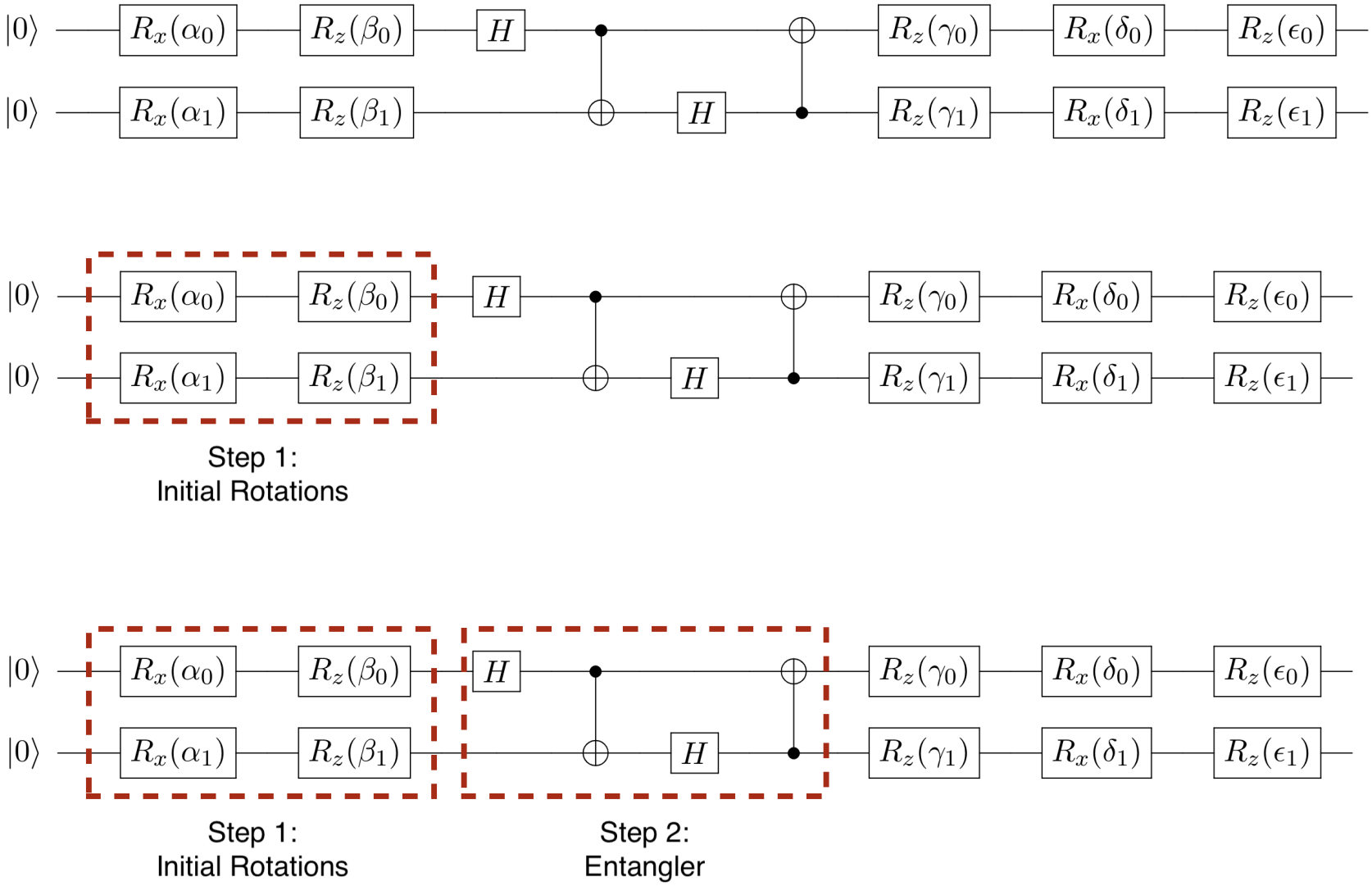
- problem ansatz (e.g. UCC "gold standard")
- hardware ansatz

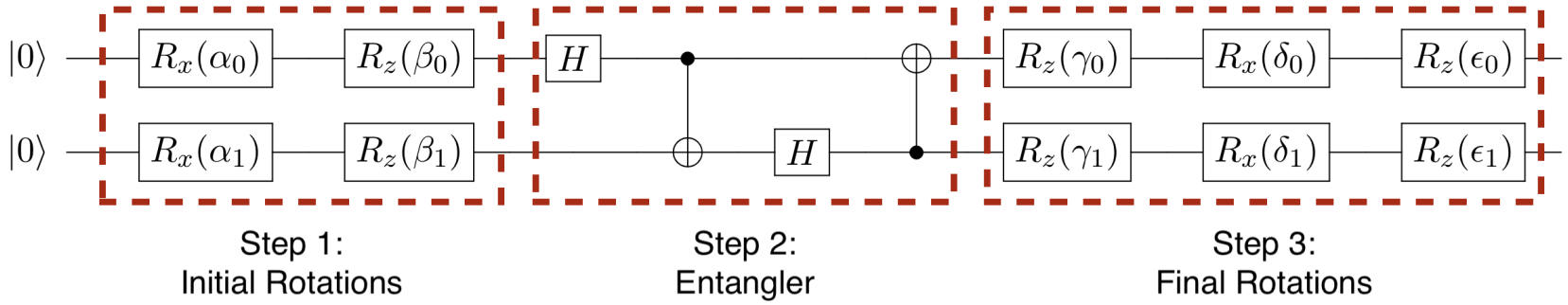
Our choice: **hardware ansatz**

[Accounting for Errors in Quantum Algorithms via Individual Error Reduction, M. Otten and S. Gray]

[Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets, A. Kandala et al.]

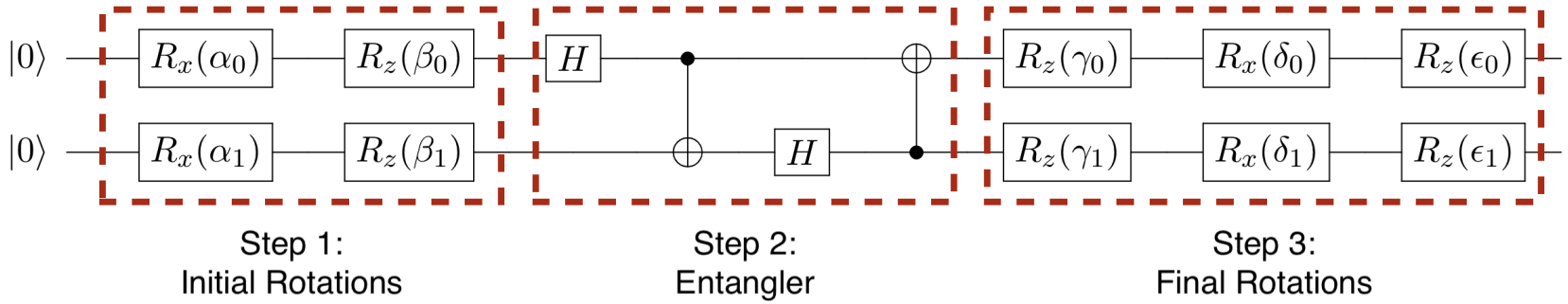
"entangling ansatz"


$|0\rangle$
$|0\rangle$
$R_x(\alpha_0)$
$R_x(\alpha_1)$
$R_z(\beta_0)$
$R_z(\beta_1)$
H
H
$R_z(\gamma_0)$
$R_z(\gamma_1)$
$R_x(\delta_0)$
$R_x(\delta_1)$
$R_z(\epsilon_0)$
$R_z(\epsilon_1)$
Step 1:
Initial Rotations
Step 1:
Initial Rotations
Step 2:
Entangler



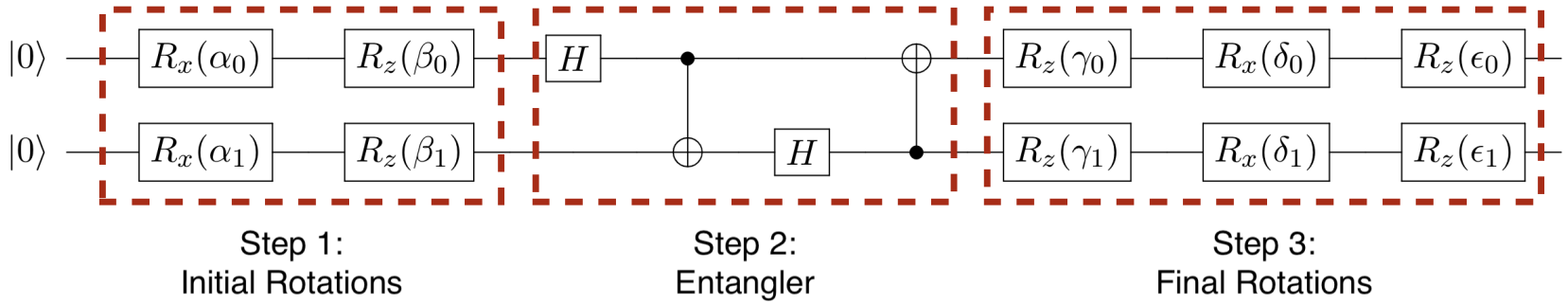
In [3]:

```
scaffold_code = """

...

int main() {
  qbit reg[2];
  cbit result[2];

  prepareAnsatz(reg);
  measure(reg, result);

  return 0;
}
"""
```


$|0\rangle$
$|0\rangle$
$R_x(\alpha_0)$
$R_z(\beta_0)$
$R_x(\alpha_1)$
$R_z(\beta_1)$
H
H
$R_z(\gamma_0)$
$R_x(\delta_0)$
$R_z(\epsilon_0)$
$R_z(\gamma_1)$
$R_x(\delta_1)$
$R_z(\epsilon_1)$
Step 1:
Initial Rotations
Step 2:
Entangler
Step 3:
Final Rotations

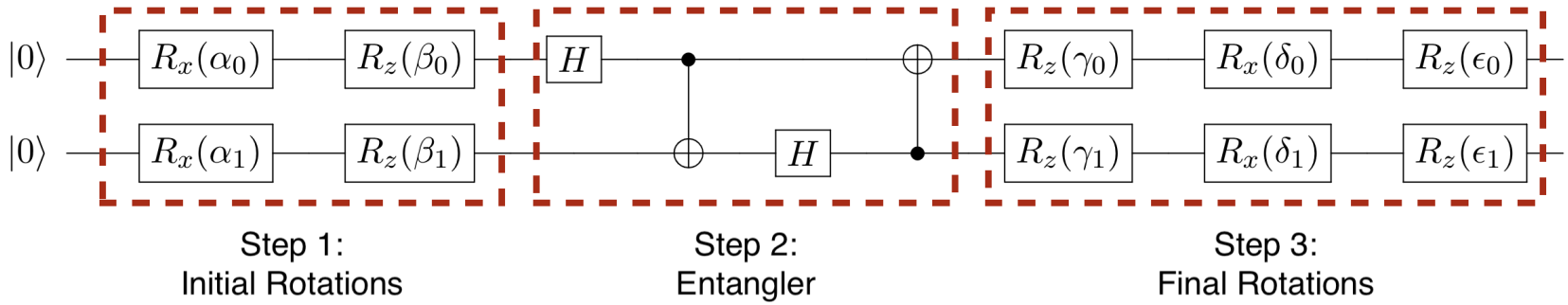
In [ ]:

```
scaffold_code = """
...
module initialRotations(qbit reg[2]) {
  ...
}

module entangler(qbit reg[2]) {
  // IMPLEMENT THIS
}

module finalRotations(qbit reg[2]) {
  ...
}

module prepareAnsatz(qbit reg[2]) {
  initialRotations(reg);
  entangler(reg);
  finalRotations(reg);
}
...
"""
```



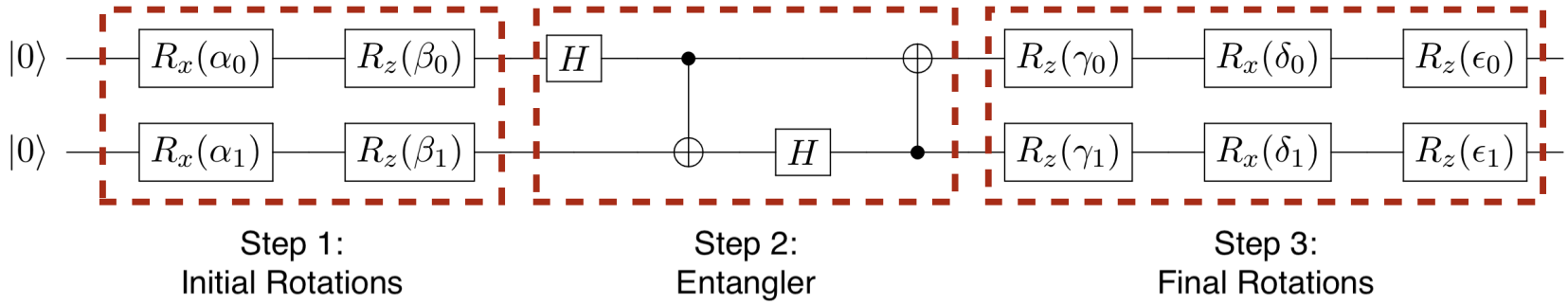
In [4]:

```
scaffold_code = """

...

module initialRotations(qbit reg[2]) {
  Rx(reg[0], alpha0);
  Rx(reg[1], alpha1);

  Rz(reg[0], beta0);
  Rz(reg[1], beta1);
}

...

"""
```



In [ ]:

```
scaffold_code = """

...

module entangler(qbit reg[2]) {
  H(reg[0]);
  CNOT(reg[1], reg[0]);

  H(reg[1]);
  CNOT(reg[0], reg[1]);
}

...

"""
```



In [ ]:

```
scaffold_code = """

...

module finalRotations(qbit reg[2]) {
  Rz(reg[0], gamma0);
  Rz(reg[1], gamma1);

  Rx(reg[0], delta0);
  Rx(reg[1], delta1);

  Rz(reg[0], epsilon0);
  Rz(reg[1], epsilon1);
}

...

"""
```

Measuring $\langle H \rangle$

For Hydrohelium (He-H+) with bond distance of 90pm, Hamiltonian (energy) is:

$$H = -3.851II - 0.229I\sigma_x - 1.047I\sigma_z - 0.229\sigma_x I + 0.261\sigma_x\sigma_x + 0.229\sigma_x\sigma_z - 1.0467\sigma$$

[A variational eigenvalue solver on a photonic quantum processor, A. Peruzzo et al.]

Consider $\sigma_z I$ term.

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

Eigenvalues are:

- +1 for $|00\rangle$ and $|01\rangle$
- -1 for $|10\rangle$ and $|11\rangle$

Prescription for measuring $\sigma_z I$:

- Simply measure qubits
  - if we measure $|00\rangle$ or $|01\rangle$, outcome is +1

- if we measure $|10\rangle$ or $|11\rangle$, outcome is -1

We want to minimize $-1.0467\sigma_z I$, so +1 is better

In [9]:

```
scaffold_code = """

...

module measure(qbit reg[2], cbit result[2]) {
  result[0] = MeasZ(reg[0]);
  result[1] = MeasZ(reg[1]);
}

...

"""
```

In [9]:

```
scaffold_code = """
const double alpha0 = 0.0, beta0 = 0.0, gamma0 = 0.0, delta0 = 0.0, epsilon0 = 0.0;
const double alpha1 = 0.0, beta1 = 0.0, gamma1 = 0.0, delta1 = 0.0, epsilon1 = 0.0;

module initialRotations(qbit reg[2]) {
  Rx(reg[0], alpha0);
  Rx(reg[1], alpha1);

  Rz(reg[0], beta0);
  Rz(reg[1], beta1);
}

module entangler(qbit reg[2]) {
  H(reg[0]);
  CNOT(reg[0], reg[1]);

  H(reg[1]);
  CNOT(reg[1], reg[0]);
}

module finalRotations(qbit reg[2]) {
  Rz(reg[0], gamma0);
  Rz(reg[1], gamma1);

  Rx(reg[0], delta0);
  Rx(reg[1], delta1);

  Rz(reg[0], epsilon0);
  Rz(reg[1], epsilon1);
```

```
}

module prepareAnsatz(qbit reg[2]) {
  initialRotations(reg);
  entangler(reg);
  finalRotations(reg);
}

module measure(qbit reg[2], cbit result[2]) {
  result[0] = MeasZ(reg[0]);
  result[1] = MeasZ(reg[1]);
}

int main() {
  qbit reg[2];
  cbit result[2];

  prepareAnsatz(reg);
  measure(reg, result);

  return 0;
}
"""
```

In [10]:

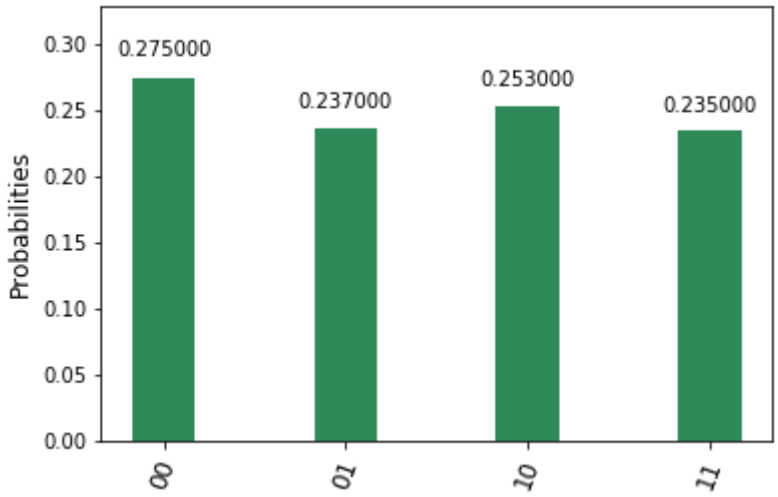
```
# Compile the Scaffold to OpenQASM
from scaffcc_interface import ScaffCC
openqasm = ScaffCC(scaffold_code, disable_rotation_decomposition=True).get_openqasm
()
print(openqasm)
```

```
OPENQASM 2.0;
include "qelib1.inc";
qreg reg[2];
creg result[2];
h reg[0];
rz(0.000000e+00) reg[0];
h reg[0];
h reg[1];
rz(0.000000e+00) reg[1];
h reg[1];
rz(0.000000e+00) reg[0];
rz(0.000000e+00) reg[1];
h reg[0];
cx reg[0],reg[1];
h reg[1];
cx reg[1],reg[0];
rz(0.000000e+00) reg[0];
rz(0.000000e+00) reg[1];
h reg[0];
rz(0.000000e+00) reg[0];
h reg[0];
```

```
h reg[1];
rz(0.000000e+00) reg[1];
h reg[1];
rz(0.000000e+00) reg[0];
rz(0.000000e+00) reg[1];
measure reg[0] -> result[0];
measure reg[1] -> result[1];
```

In [11]:

```
%matplotlib inline
from qiskit import load_qasm_string, available_backends, execute, get_backend
vqe_circ = load_qasm_string(openqasm)

num_shots = 1000
result = execute(vqe_circ, backend='local_qasm_simulator', shots=num_shots).result()

counts = result.get_counts()
from qiskit.tools.visualization import plot_histogram
plot_histogram(counts)

expected_value = (counts['00'] + counts['01'] - counts['10'] - counts['11']) / num_s
hots
print('Expected value for sigma_z I is : %s' % expected_value)
```




```
Expected value for sigma_z I is : 0.024
```

Pick your own angles! Largest expected value ($\sigma_z I$ has negative coefficient) wins.

For other terms of Hamiltonian, Pauli Measurement

Pauli Measurement	U
$Z \otimes \mathbf{1}$	$\mathbf{1} \otimes \mathbf{1}$
$X \otimes \mathbf{1}$	$H \otimes \mathbf{1}$
$Y \otimes \mathbf{1}$	$HS^\dagger \otimes \mathbf{1}$
$\mathbf{1} \otimes Z$	SWAP
$\mathbf{1} \otimes X$	$(H \otimes \mathbf{1})\,\mathrm{SWAP}$
$\mathbf{1} \otimes Y$	$(HS^\dagger \otimes \mathbf{1})\,\mathrm{SWAP}$
$Z \otimes Z$	CNOT_{10}
$X \otimes Z$	$\mathrm{CNOT}_{10}(H \otimes \mathbf{1})$
$Y \otimes Z$	$\mathrm{CNOT}_{10}(HS^\dagger \otimes \mathbf{1})$
$Z \otimes X$	$\mathrm{CNOT}_{10}(\mathbf{1} \otimes H)$
$X \otimes X$	$\mathrm{CNOT}_{10}(H \otimes H)$

[https://docs.microsoft.com/en-us/quantum/quantum-concepts-7-paulimeasurements?view=qsharp-preview (https://docs.microsoft.com/en-us/quantum/quantum-concepts-7-paulimeasurements?view=qsharp-preview)]