



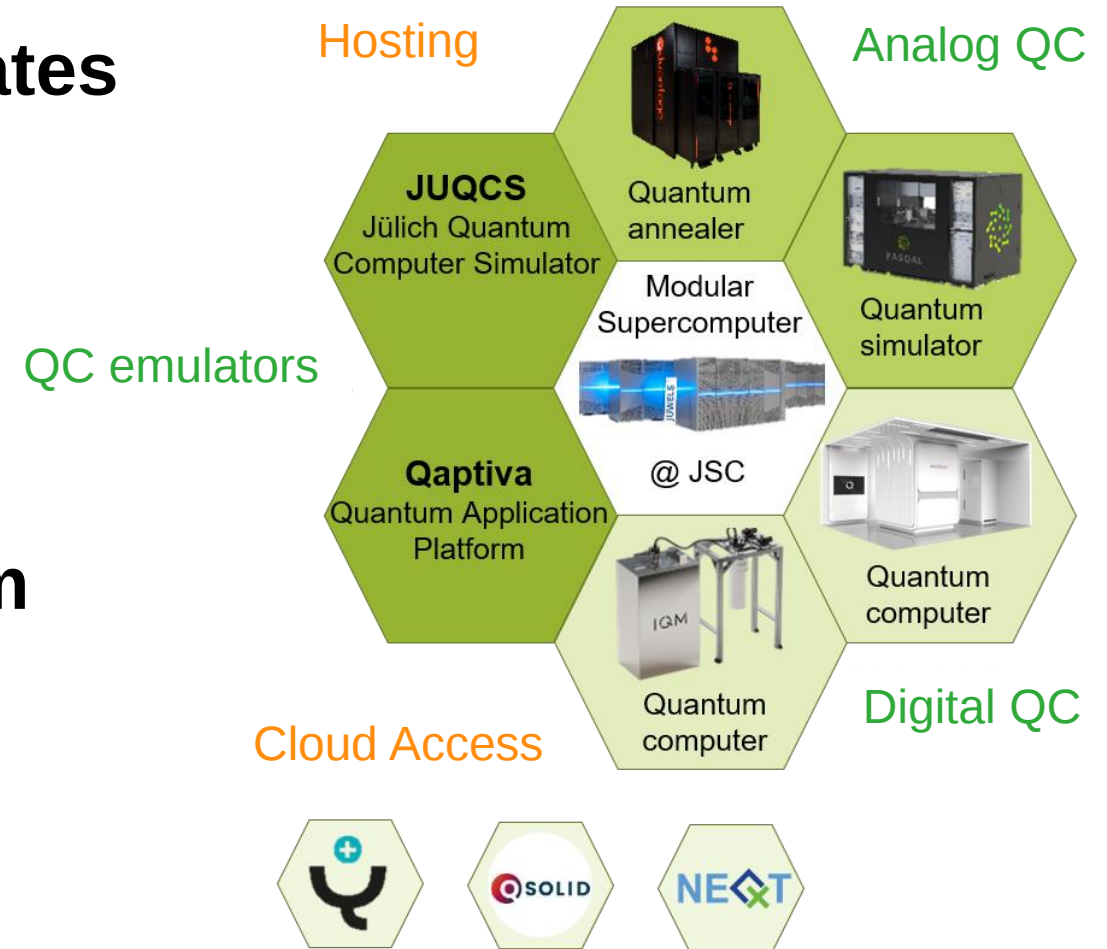
INTRODUCTION TO GATE-BASED QUANTUM COMPUTING

JUNIQ Summer School on Quantum Computing 2024

26 AUGUST 2024 | DR. DENNIS WILLSCH

CONTENTS

1. Quantum Bits and Quantum Gates
2. Programming and Simulating Quantum Circuits
3. Applications:
 1. Quantum Fourier Transform
 2. Quantum Adder
 3. Quantum Approximate Optimization Algorithm



QUANTUM BITS AND QUANTUM GATES

A single qubit

➤ Definition of a single qubit

$$|\psi\rangle = \psi_0 |0\rangle + \psi_1 |1\rangle = \begin{pmatrix} \psi_0 \\ \psi_1 \end{pmatrix}$$

Computational basis states:

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

Digital computer: **Bit**

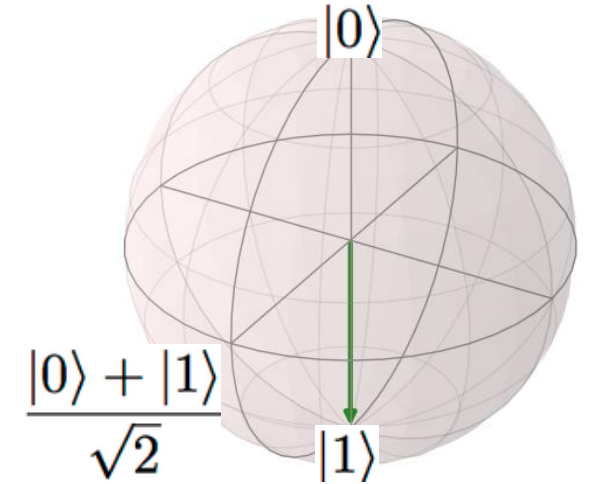
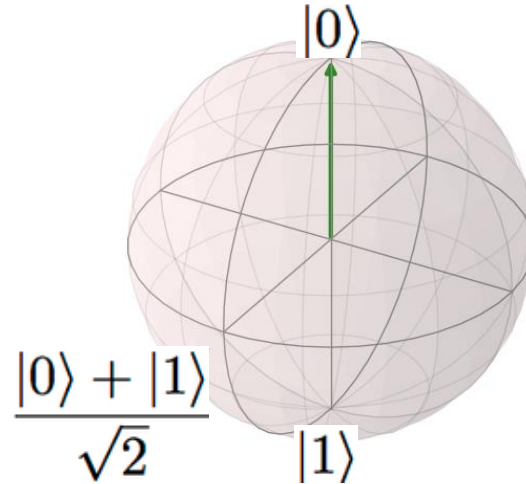


0



1

Quantum computer: **Quantum Bit = Qubit**



QUANTUM BITS AND QUANTUM GATES

A single qubit

➤ Definition of a single qubit

$$|\psi\rangle = \psi_0 |0\rangle + \psi_1 |1\rangle = \begin{pmatrix} \psi_0 \\ \psi_1 \end{pmatrix}$$

Normalization:

$$\langle\psi|\psi\rangle = |\psi_0|^2 + |\psi_1|^2 = 1$$
$$\langle\psi| = (\psi_0^*, \psi_1^*)$$

➤ Representation on the Bloch sphere

$$|\psi\rangle = \cos \frac{\vartheta}{2} |0\rangle + e^{i\varphi} \sin \frac{\vartheta}{2} |1\rangle = \begin{pmatrix} \cos \frac{\vartheta}{2} \\ e^{i\varphi} \sin \frac{\vartheta}{2} \end{pmatrix}$$

Global phase:

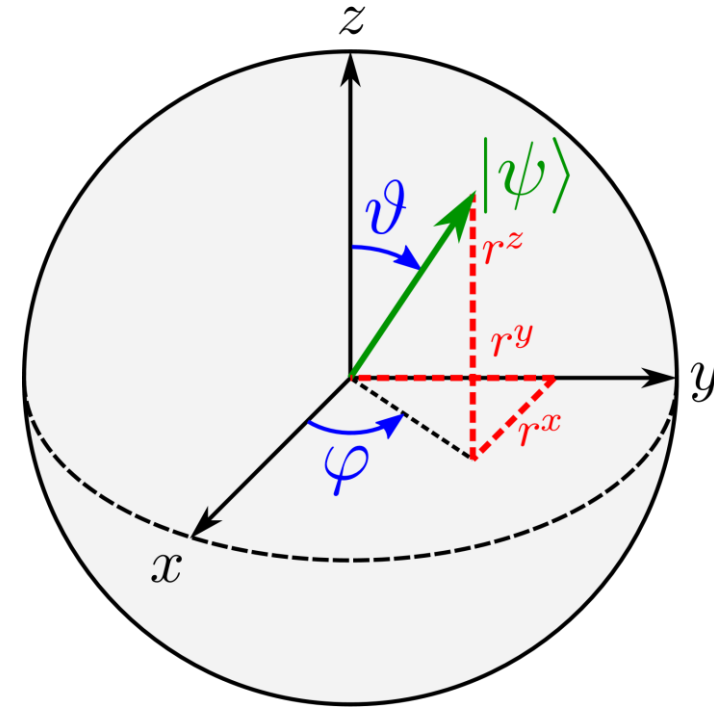
$$|\psi\rangle \equiv e^{i\Phi} |\psi\rangle$$

➤ Evaluation of the Cartesian coordinates

$$\vec{r} = \begin{pmatrix} r^x \\ r^y \\ r^z \end{pmatrix} = \begin{pmatrix} \langle\psi|\sigma^x|\psi\rangle \\ \langle\psi|\sigma^y|\psi\rangle \\ \langle\psi|\sigma^z|\psi\rangle \end{pmatrix} = \begin{pmatrix} \sin \vartheta \cos \varphi \\ \sin \vartheta \sin \varphi \\ \cos \vartheta \end{pmatrix}$$

Pauli matrices:

$$\sigma^x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad \sigma^y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \quad \sigma^z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$



QUANTUM BITS AND QUANTUM GATES

Gates as rotations on the Bloch sphere

➤ Rotations around the axes of the Bloch sphere

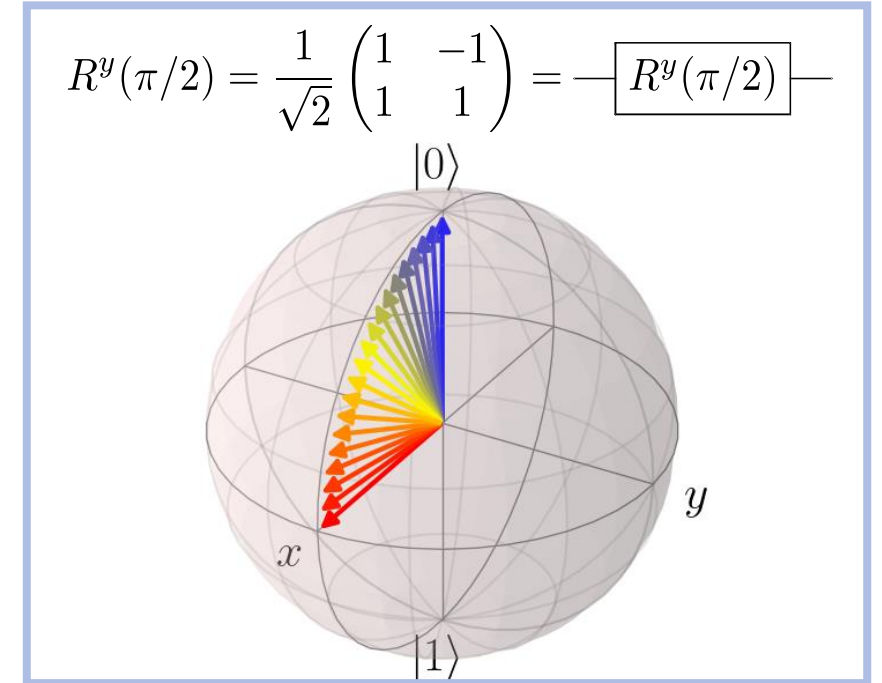
$$R^x(\theta) = e^{-i\theta\sigma^x/2} = \begin{pmatrix} \cos(\theta/2) & -i\sin(\theta/2) \\ -i\sin(\theta/2) & \cos(\theta/2) \end{pmatrix}$$

$$R^y(\theta) = e^{-i\theta\sigma^y/2} = \begin{pmatrix} \cos(\theta/2) & -\sin(\theta/2) \\ \sin(\theta/2) & \cos(\theta/2) \end{pmatrix}$$

$$R^z(\theta) = e^{-i\theta\sigma^z/2} = \begin{pmatrix} \exp(-i\theta/2) & 0 \\ 0 & \exp(i\theta/2) \end{pmatrix}$$

➤ General rotation gate

$$R^{\vec{n}}(\theta) = e^{-i\theta\vec{n}\cdot\vec{\sigma}/2} = \cos\frac{\theta}{2} I - i\sin\frac{\theta}{2} \vec{n} \cdot \vec{\sigma}$$



Standard gate set:

$$X = \sigma^x$$

$$Y = \sigma^y$$

$$Z = \sigma^z$$

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

$$S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$$

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

QUANTUM BITS AND QUANTUM GATES

Multiple qubits

➤ Multi-qubit state

$$|\psi\rangle = \sum_{j=0}^{2^n-1} \psi_j |j\rangle = \psi_0 |0\cdots 00\rangle + \psi_1 |0\cdots 01\rangle + \cdots + \psi_{2^n-1} |1\cdots 11\rangle = \begin{pmatrix} \psi_0 \\ \psi_1 \\ \vdots \\ \psi_{2^n-1} \end{pmatrix}$$

➤ Important two-qubit gates

$$\text{CNOT} = \text{CX} = \begin{matrix} & |00\rangle & |01\rangle & |10\rangle & |11\rangle \\ \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} & = & \begin{array}{c} \bullet \\ | \\ \oplus \end{array} \end{matrix}$$

$$\text{CZ} = \begin{matrix} & |00\rangle & |01\rangle & |10\rangle & |11\rangle \\ \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} & = & \begin{array}{c} \bullet \\ | \\ \bullet \end{array} \end{matrix}$$

➤ General controlled gates

$$CU_{i_1 i_2} |q_0 \cdots q_{n-1}\rangle = \begin{cases} |q_0 \cdots q_{n-1}\rangle & (\text{if } q_{i_1} = 0) \\ |q_0 \cdots q_{i_2-1}\rangle (U |q_{i_2}\rangle) |q_{i_2+1} \cdots q_{n-1}\rangle & (\text{if } q_{i_1} = 1) \end{cases}$$

Computational basis states:

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

Tensor product

$$|q_0 q_1\rangle = |q_0\rangle \otimes |q_1\rangle \quad A \otimes B = \begin{pmatrix} a_{00}B & a_{01}B & \cdots \\ a_{10}B & a_{11}B & \\ \vdots & & \ddots \end{pmatrix}$$

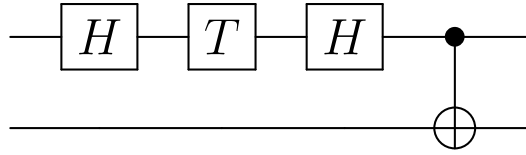
$$CU = \begin{pmatrix} I & 0 \\ 0 & U \end{pmatrix} = \begin{array}{c} \bullet \\ | \\ \boxed{U} \end{array}$$

$$\text{SWAP} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{array}{c} \times \\ | \\ \times \end{array}$$

PROGRAMMING AND SIMULATING QUANTUM CIRCUITS

Input: _____

Programm =
Circuit of
Quantum Gates



Result of this
program

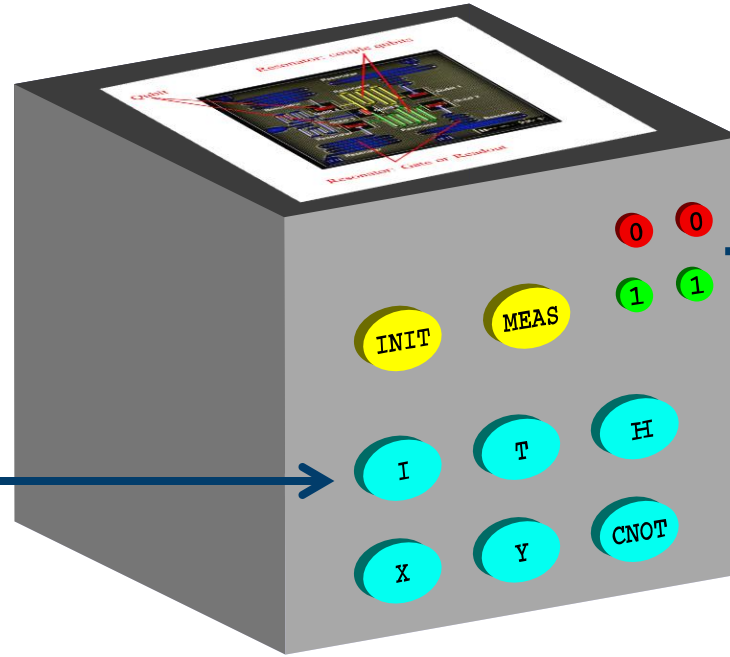
Option 1:
Compute the
unitary matrix

$$\text{CNOT} (H \otimes I) (T \otimes I) (H \otimes I) =$$

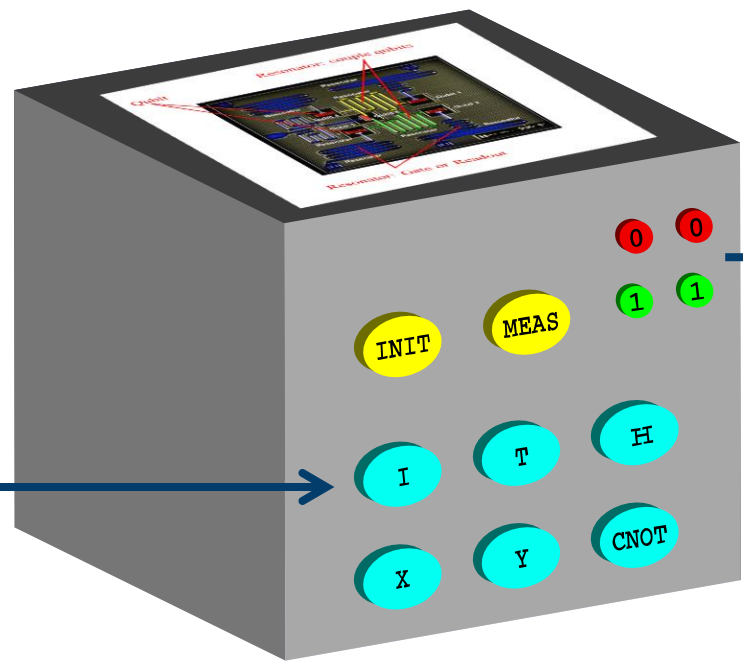
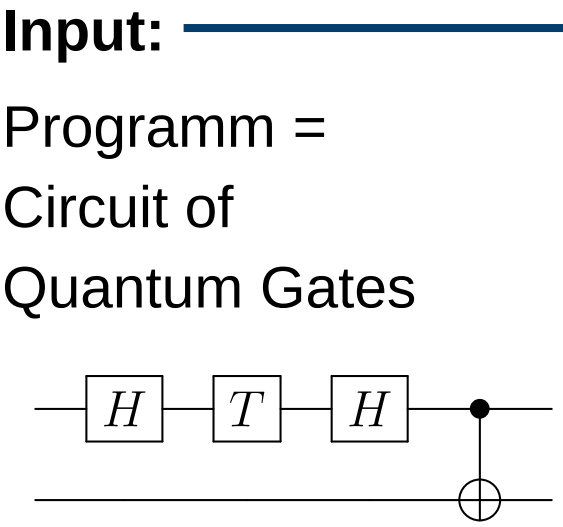
$$\begin{pmatrix} \frac{1+e^{i\pi/4}}{2} & 0 & \frac{1-e^{i\pi/4}}{2} & 0 \\ 0 & \frac{1+e^{i\pi/4}}{2} & 0 & \frac{1-e^{i\pi/4}}{2} \\ 0 & \frac{1-e^{i\pi/4}}{2} & 0 & \frac{1+e^{i\pi/4}}{2} \\ \frac{1-e^{i\pi/4}}{2} & 0 & \frac{1+e^{i\pi/4}}{2} & 0 \end{pmatrix}$$

$$\sqrt{0.85 \dots} |00\rangle + \sqrt{0.15 \dots} |11\rangle$$

Output: Bitstrings
„00“, „01“, „10“, „11“



PROGRAMMING AND SIMULATING QUANTUM CIRCUITS



Output: Bitstrings
„00“, „01“, „10“, „11“

Result of this
program

Option 2: Track
the unitary
transformations

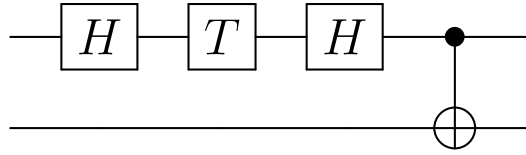
$$\sqrt{0.85 \dots} |00\rangle + \sqrt{0.15 \dots} |11\rangle$$

$$\begin{aligned}
 |00\rangle &\xrightarrow{H \otimes I} \frac{1}{\sqrt{2}} |00\rangle + \frac{1}{\sqrt{2}} |10\rangle \xrightarrow{T \otimes I} \frac{1}{\sqrt{2}} |00\rangle + \frac{e^{i\pi/4}}{\sqrt{2}} |10\rangle \\
 &\xrightarrow{H \otimes I} \frac{1 + e^{i\pi/4}}{\sqrt{2}} |00\rangle + \frac{1 - e^{i\pi/4}}{\sqrt{2}} |10\rangle \\
 &\xrightarrow{\text{CNOT}} \frac{1 + e^{i\pi/4}}{\sqrt{2}} |00\rangle + \frac{1 - e^{i\pi/4}}{\sqrt{2}} |11\rangle
 \end{aligned}$$

PROGRAMMING AND SIMULATING QUANTUM CIRCUITS

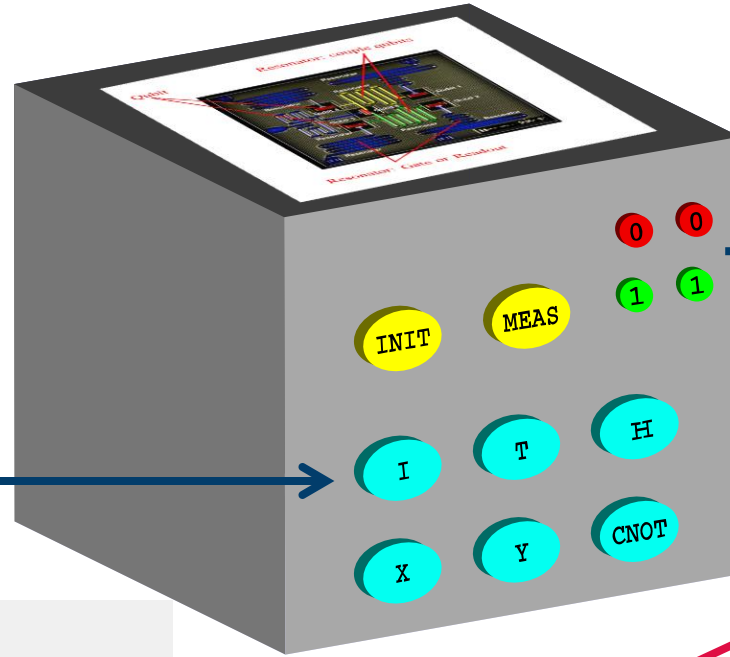
Input:

Programm =
Circuit of
Quantum Gates



Option 3: Simulate the circuit

```
1 import qiskit
2
3 circuit = qiskit.QuantumCircuit(2)
4 circuit.h(0)
5 circuit.t(0)
6 circuit.h(0)
7 circuit.cx(0,1)
8
9 from juqcs import Juqcs
```



Output: Bitstrings
„00“, „01“, „10“, „11“

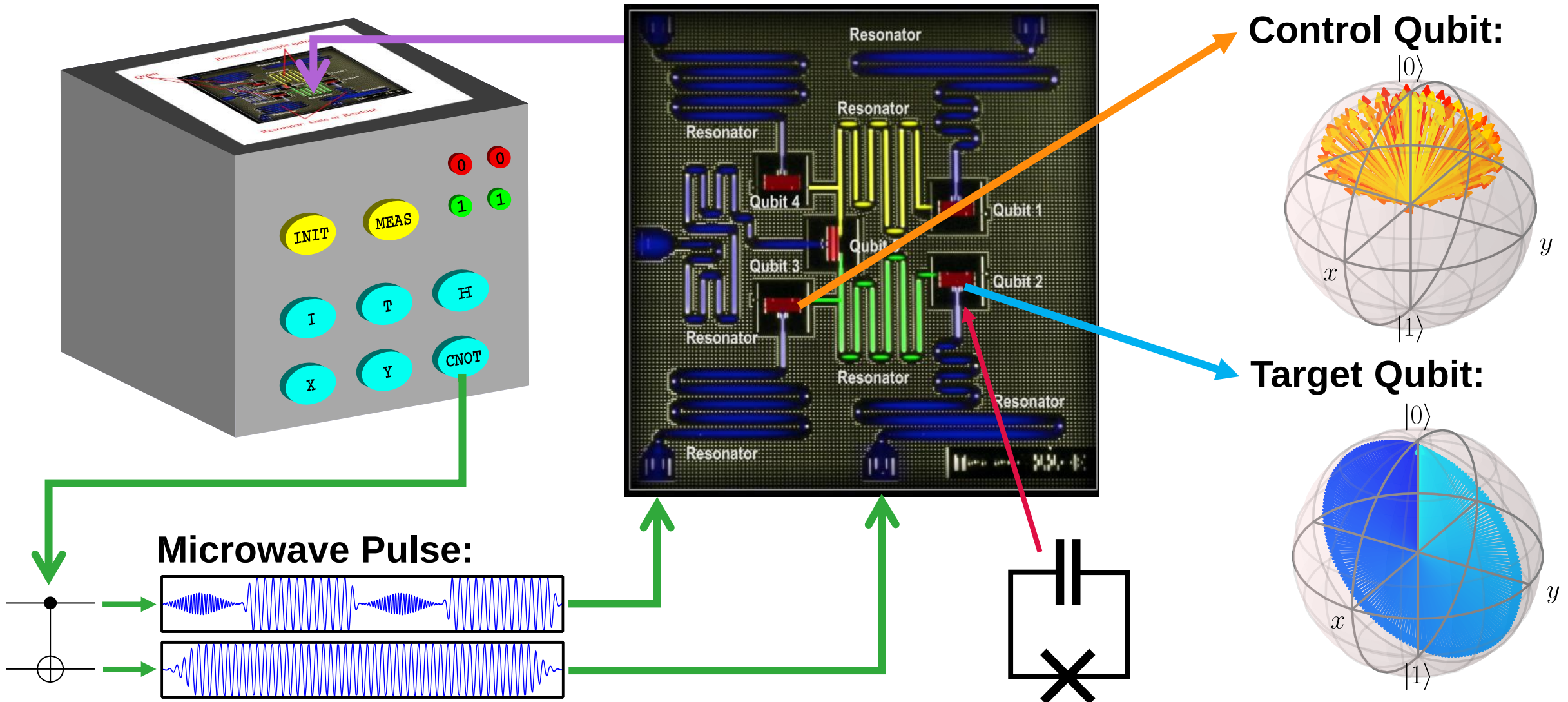
$$\sqrt{0.85 \dots} |00\rangle + \sqrt{0.15 \dots} |11\rangle$$

JUQCS
(Jülich Universal
Quantum Computer
Simulator)

Python Package qiskit-juqcs: <https://pypi.org/project/qiskit-juqcs/>

Measure **00** with 85 %
and **11** with 15 %

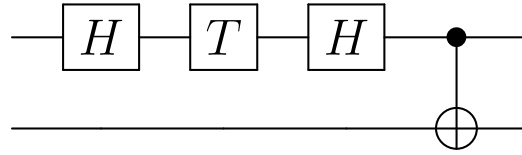
INTERLUDE: SUPERCONDUCTING QUANTUM COMPUTERS



PROGRAMMING AND SIMULATING QUANTUM CIRCUITS

Simulation with Qiskit Aer

qasm_simulator



statevector_simulator

```
import qiskit
circuit = qiskit.QuantumCircuit(2)
circuit.h(0)
circuit.t(0)
circuit.h(0)
circuit.cx(0,1)
circuit.measure_all()

backend = qiskit.Aer.get_backend('qasm_simulator')
result = backend.run(circuit,
                     shots=1000).result()
result.get_counts()
```

{'00': 846, '11': 154}

Measure **00** with 85 %
and **11** with 15 %

```
import qiskit
circuit = qiskit.QuantumCircuit(2)
circuit.h(0)
circuit.t(0)
circuit.h(0)
circuit.cx(0,1)

backend = qiskit.Aer.get_backend('statevector_simulator')
result = backend.run(circuit.reverse_bits()).result()

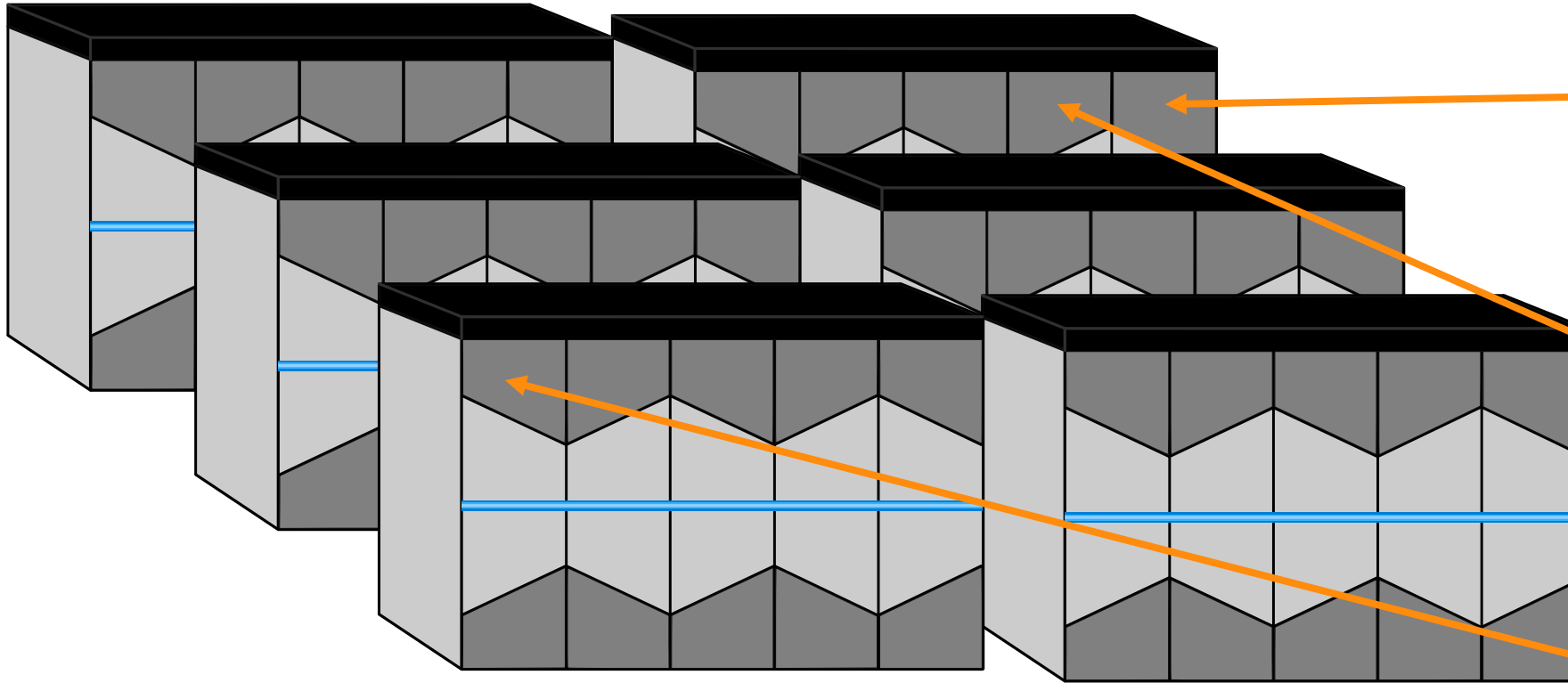
result.get_statevector()
```

Statevector([0.85355339+0.35355339j, 0. +0.j, 0. +0.j, 0.14644661-0.35355339j],

$$\sqrt{0.85} \dots |00\rangle + \sqrt{0.15} \dots |11\rangle$$

SIMULATING QUANTUM CIRCUITS

Simulation with JUQCS (large-scale simulations on a supercomputer)



GPU #0 (Node #0)

$$\begin{pmatrix} \psi_{000000000000\ 0\cdots 0} \\ \vdots \\ \psi_{000000000000\ 1\cdots 1} \end{pmatrix}$$

GPU #4 (Node #1)

$$\begin{pmatrix} \psi_{000000000001\ 0\cdots 0} \\ \vdots \\ \psi_{000000000001\ 1\cdots 1} \end{pmatrix}$$

GPU #2047

$$\begin{pmatrix} \psi_{111111111111\ 0\cdots 0} \\ \vdots \\ \psi_{111111111111\ 1\cdots 1} \end{pmatrix}$$

- **Video:** Simulation of Quantum Computers on GPUs
<https://youtu.be/JUSEYCKh2ac>
- **JUNIQ:** Possible to simulate around 40 qubits (45 with JUQCS-G)
<https://juniq.fz-juelich.de>

QUANTUM FOURIER TRANSFORM

An operation to move registers into phases and back

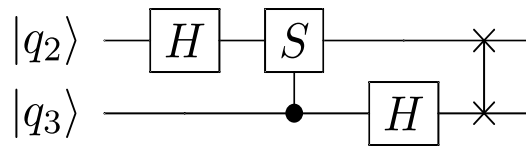
- The formal definition of the QFT is:

$$|q_0 q_1 q_2 \dots\rangle \xrightarrow{\text{QFT}} \sum_{q'_0 q'_1 q'_2 \dots} e^{2\pi i (q_0 q_1 q_2 \dots) (q'_0 q'_1 q'_2 \dots) / 2^N} |q'_0 q'_1 q'_2 \dots\rangle$$

- Intuition:

$$|j\rangle \leftrightarrow \sum_{j'} e^{2\pi i j j' / 2^n} |j'\rangle$$

- Implementation for two qubits:



- Verification:

$$\text{SWAP} (I \otimes H) CS (H \otimes I) |q_2 q_3\rangle \propto \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & i & -1 & -i \\ 1 & -1 & 1 & -1 \\ 1 & -i & -1 & i \end{pmatrix} |q_2 q_3\rangle = \sum_{q'_2 q'_3} e^{2\pi i (q_2 q_3) (q'_2 q'_3) / 4} |q'_2 q'_3\rangle$$

$$\begin{aligned} & e^{2\pi i (q_2 q_3) 0/4} |0\rangle \\ & + e^{2\pi i (q_2 q_3) 1/4} |1\rangle \\ & + e^{2\pi i (q_2 q_3) 2/4} |2\rangle \\ & + e^{2\pi i (q_2 q_3) 3/4} |3\rangle \end{aligned}$$

QUANTUM ADDER

A nice application of the QFT

- We are looking for a quantum circuit to implement the following unitary operation

$$|q_0 q_1\rangle |q_2 q_3\rangle \mapsto |q_0 q_1\rangle |q_0 q_1 + q_2 q_3\rangle$$

- Such a modulo-4 adder would also work on superpositions

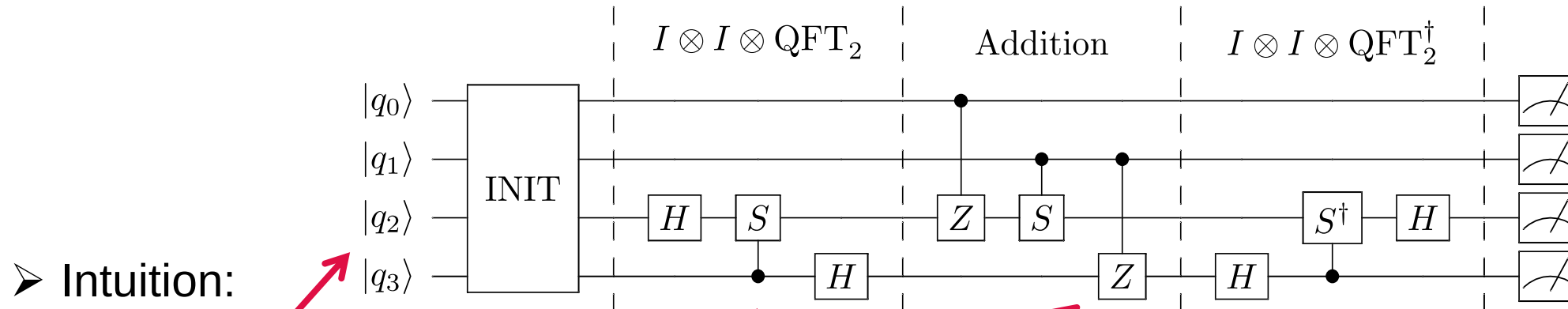
- Examples:

$$\begin{array}{ll}
 |2\rangle |1\rangle & \mapsto |2\rangle |3\rangle, \\
 |2\rangle \frac{|0\rangle + |1\rangle}{\sqrt{2}} & \mapsto |2\rangle \frac{|2\rangle + |3\rangle}{\sqrt{2}}, \\
 |2\rangle \frac{|0\rangle + |1\rangle + |2\rangle}{\sqrt{3}} & \mapsto |2\rangle \frac{|2\rangle + |3\rangle + |0\rangle}{\sqrt{3}}, \\
 \frac{|0\rangle + |1\rangle}{\sqrt{2}} \frac{|2\rangle + |3\rangle + |0\rangle}{\sqrt{3}} & \mapsto ?
 \end{array}$$

QUANTUM ADDER

A nice application of the QFT

- Idea: → The QFT converts between registers and phases
 - By using bitwise phase shifts, we can implement the addition in the exponent
 - This is automatically modulo 4
- A circuit to realize this idea is



$$|j\rangle|k\rangle \mapsto |j\rangle \left(\sum_{k'} e^{2\pi i k k' / 2^n} |k'\rangle \right) = \sum_{k'} e^{2\pi i k k' / 2^n} |j\rangle|k'\rangle \mapsto \sum_{k'} e^{2\pi i (j+k) k' / 2^n} |j\rangle|k'\rangle \mapsto |j\rangle|j+k\rangle$$

QUANTUM APPROXIMATE OPTIMIZATION ALGORITHM

A very brief overview of the QAOA

- Say we want to find the minimum of the function

$$E(q_0, \dots, q_{n-1}) = \sum_{i,j} q_i Q_{ij} q_j \quad \Leftrightarrow$$

“QUBO” (q = 0,1)

$$E(s_0, \dots, s_{n-1}) = \sum h_i s_i + \sum_{i < j} J_{ij} s_i s_j$$

“Ising” (s = +1,-1)

- Combinatorial optimization problem
→ discrete optimization is hard!

- Idea: With a quantum circuit, we can create a superposition

$$|\psi\rangle = \psi_0 |0 \dots 00\rangle + \dots + \psi_{q_0^* \dots q_{n-1}^*} |q_0^* \dots q_{n-1}^*\rangle + \dots + \psi_{2^n-1} |1 \dots 11\rangle$$

Solution to the problem

- Goal: Enhance this term!

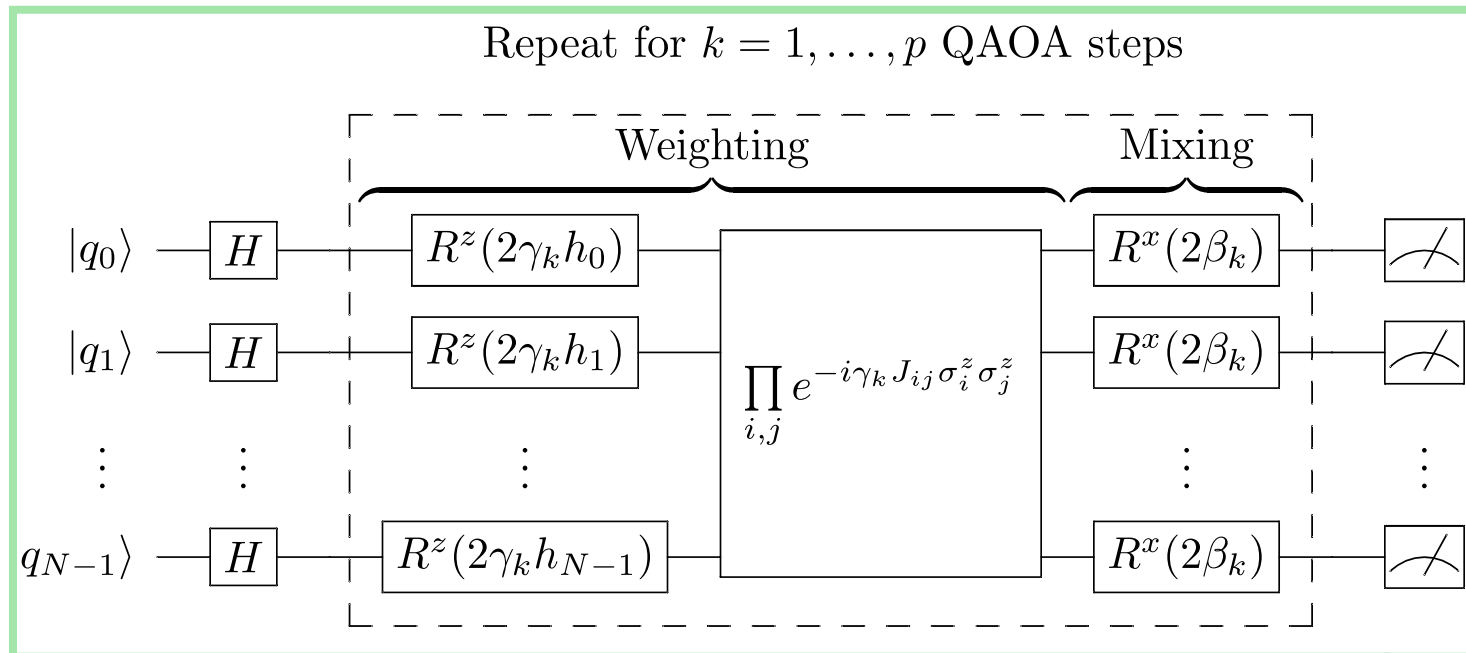
QUANTUM APPROXIMATE OPTIMIZATION ALGORITHM

A very brief overview of the QAOA

- Goal: Enhance this term!

$$|\psi\rangle = \psi_0 |0 \cdots 00\rangle + \cdots + \boxed{\psi_{q_0^* \cdots q_{n-1}^*} |q_0^* \cdots q_{n-1}^*\rangle} + \cdots + \psi_{2^n-1} |1 \cdots 11\rangle$$

- Make the quantum circuit dependent on real parameters $(\beta_1, \dots, \beta_p, \gamma_1, \dots, \gamma_p)$
- Hope: these $2p$ parameters are easier searchable than the original discrete variables



QUANTUM APPROXIMATE OPTIMIZATION ALGORITHM

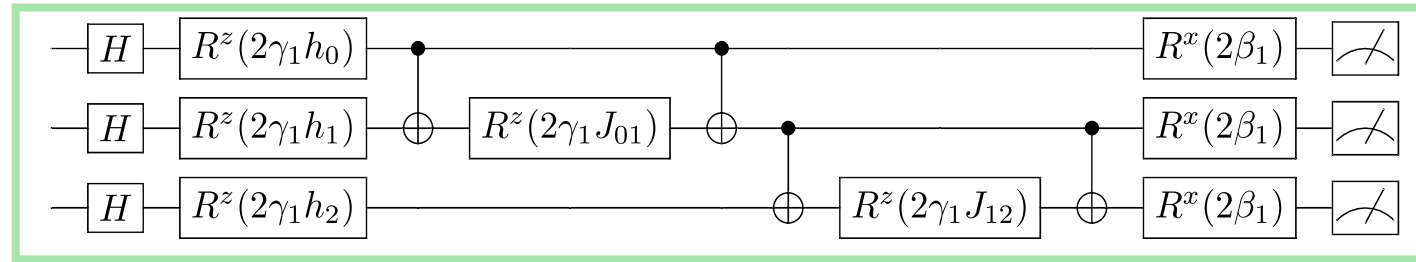
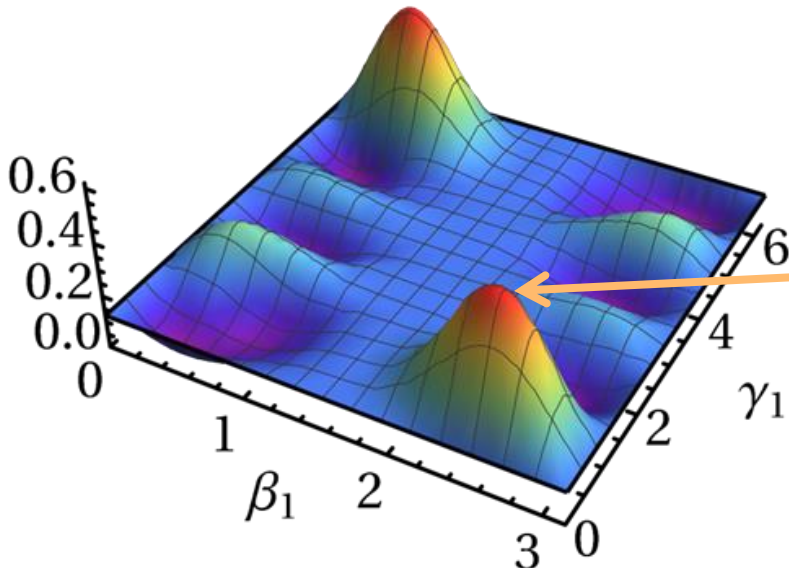
A very brief overview of the QAOA

- Goal: Enhance this term!

$$|\psi\rangle = \psi_0 |0 \cdots 00\rangle + \cdots + \boxed{\psi_{q_0^* \cdots q_{n-1}^*} |q_0^* \cdots q_{n-1}^*\rangle} + \cdots + \psi_{2^n-1} |1 \cdots 11\rangle$$

- Look at p=1 QAOA step

$$\text{Success probability } |\psi_{q_0^* \cdots q_{n-1}^*}|^2$$



There are regions where the success probability is enhanced!
How to find them?

- Will be a central topic of this School ☺



THANK YOU FOR YOUR ATTENTION

➤ Further information:

- **Programming Quantum Computers:**
- **JUQCS:** Video
- **JUQCS:** Paper

<https://arxiv.org/pdf/2201.02051.pdf>

<https://youtu.be/JUSEYCKh2ac>

<https://doi.org/10.1016/j.cpc.2022.108411>