



Quantum practice

Ana Díaz Muñoz

May, 2024

Who I am?



Ana Díaz Muñoz



IBM Quantum Composer

IBM Quantum Learning

HomeCatalogComposerLab

PhD experiment - Bell state *Saved*

FileEditView

Visualizations seed806

Setup and run

Operations

Left alignmentInspect

Search

H⊕⊗⊗⊗⊗⊗I

TST[†]S[†]P

RZz|0>⋮●if

√X√X[†]YRXRYRXX

RZZURCCXRC3X

q[0]

q[1]

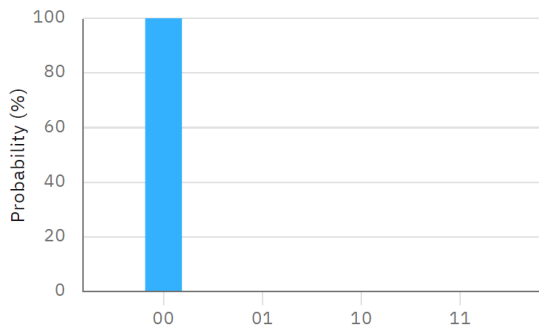
c2

Qiskit

Read only

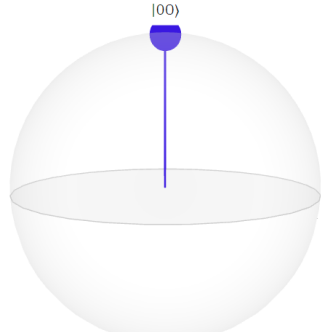
```
1 from qiskit import QuantumRegister,
2   ClassicalRegister, QuantumCircuit
3
4 qreg_q = QuantumRegister(2, 'q')
5 creg_c = ClassicalRegister(2, 'c')
6 circuit = QuantumCircuit(qreg_q, creg_c)
7
8
```

Probabilities



Computational basis states	Probability (%)
00	100
01	0
10	0
11	0

Q-sphere



π/2

π

3π/2

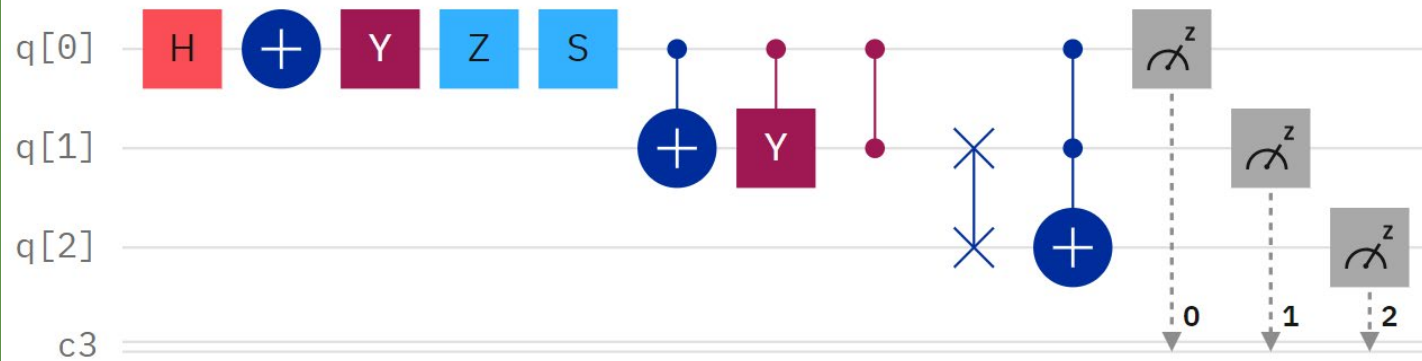
0

Phase

☒ State☐ Phase angle

Qiskit algorithms

Quantum circuit



Quantum algorithm

```
1 from qiskit import QuantumRegister,
  ClassicalRegister, QuantumCircuit
2 from numpy import pi
3
4 qreg_q = QuantumRegister(3, 'q')
5 creg_c = ClassicalRegister(3, 'c')
6 circuit = QuantumCircuit(qreg_q, creg_c)
7
8 circuit.h(qreg_q[0])
9 circuit.x(qreg_q[0])
10 circuit.y(qreg_q[0])
11 circuit.z(qreg_q[0])
12 circuit.s(qreg_q[0])
13 circuit.cx(qreg_q[0], qreg_q[1])
14 circuit.cy(qreg_q[0], qreg_q[1])
15 circuit.cz(qreg_q[0], qreg_q[1])
16 circuit.swap(qreg_q[1], qreg_q[2])
17 circuit.ccx(qreg_q[0], qreg_q[1], qreg_q[2])
18 circuit.measure(qreg_q[0], creg_c[0])
19 circuit.measure(qreg_q[1], creg_c[1])
20 circuit.measure(qreg_q[2], creg_c[2])
```

It's your turn

<https://algassert.com/quirk#>

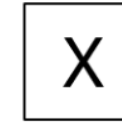
```
1 from qiskit import QuantumRegister, ClassicalRegister, QuantumCircuit
2 from numpy import pi
3
4 qreg_q = QuantumRegister(2, 'q')
5 creg_c = ClassicalRegister(2, 'c')
6 circuit = QuantumCircuit(qreg_q, creg_c)
7
8 circuit.h(qreg_q[1])
9 circuit.swap(qreg_q[0], qreg_q[1])
10 circuit.s(qreg_q[0])
11 circuit.cz(qreg_q[0], qreg_q[1])
12 circuit.measure(qreg_q[0], creg_c[0])
13 circuit.measure(qreg_q[1], creg_c[1])
```

Quantum practice

- .h(target)



- .x(target)



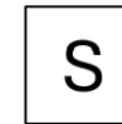
- .y(target)



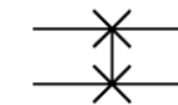
- .z(target)



- .s(target)



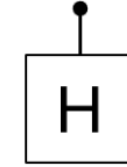
- .swap(target, target)



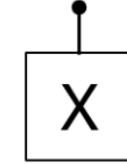
- .measure(target qubit, target bit)



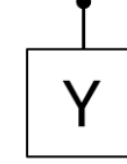
- .ch(control, target)



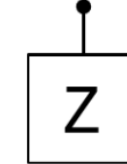
- .cx(control, target)



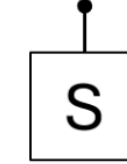
- .cy(control, target)



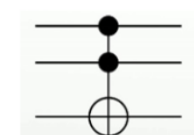
- .cz(control, target)



- .cs(control, target)



- .ccx(control, control, target)





Bell state

<https://forms.gle/jBGmktXhktXDUQfY6>



Example 2

<https://forms.gle/qPT1TBwPzBYRxQ9c9>



Thank you for your attention

Ana Díaz Muñoz

May, 2024



Check your email