Kuantum Programlama ve Kuantum Yazılım

MindFusion Days Sunumu

Kuantum Bilişim Konuşalım

DATA SCIENCE EARTH

Başlıklar

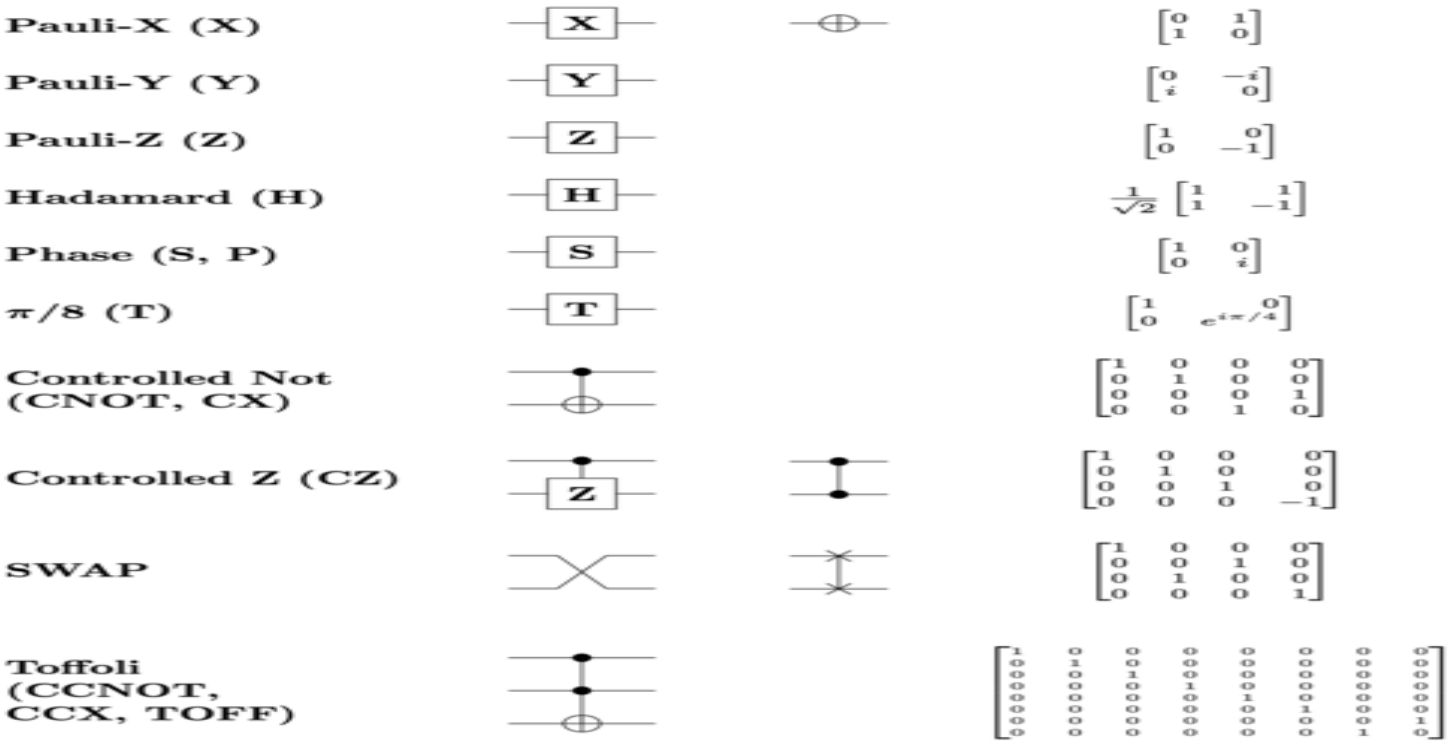
- Kuantum Programlama Nedir?
- Kuantum Yazılım Nedir?
- Kuantum Programlama Dilleri
- Klasik Programlama ile Farkları
- Kütüphaneler ve Diller
- QWorld/QTurkey Bronze Workshop
- Kuantum Programlama ile Neler Yapılabilir?

Sosyal

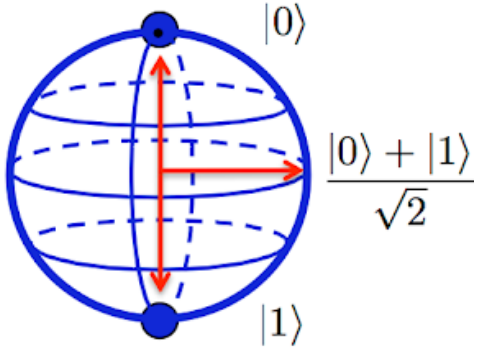
- Twitter: @dtunacs
- Medium: @dogukantuna
- LinkedIn: Doğukan Tuna

Kuantum Programlama Nedir?

- **Kuantum programlama**, bir kuantum bilgisayar üzerinde çalışabilen kuantum programları olarak adlandırılan komut dizilerini programlama işlemidir.
- Kuantum algoritmalarının ifade edilmesini sağlar.
- Temeli kuantum mekaniğinin fenomenlerinden gelmektedir.
- Klasik programlamadan farklıdır.
- 2000'lerin başında ortaya çıkmıştır.

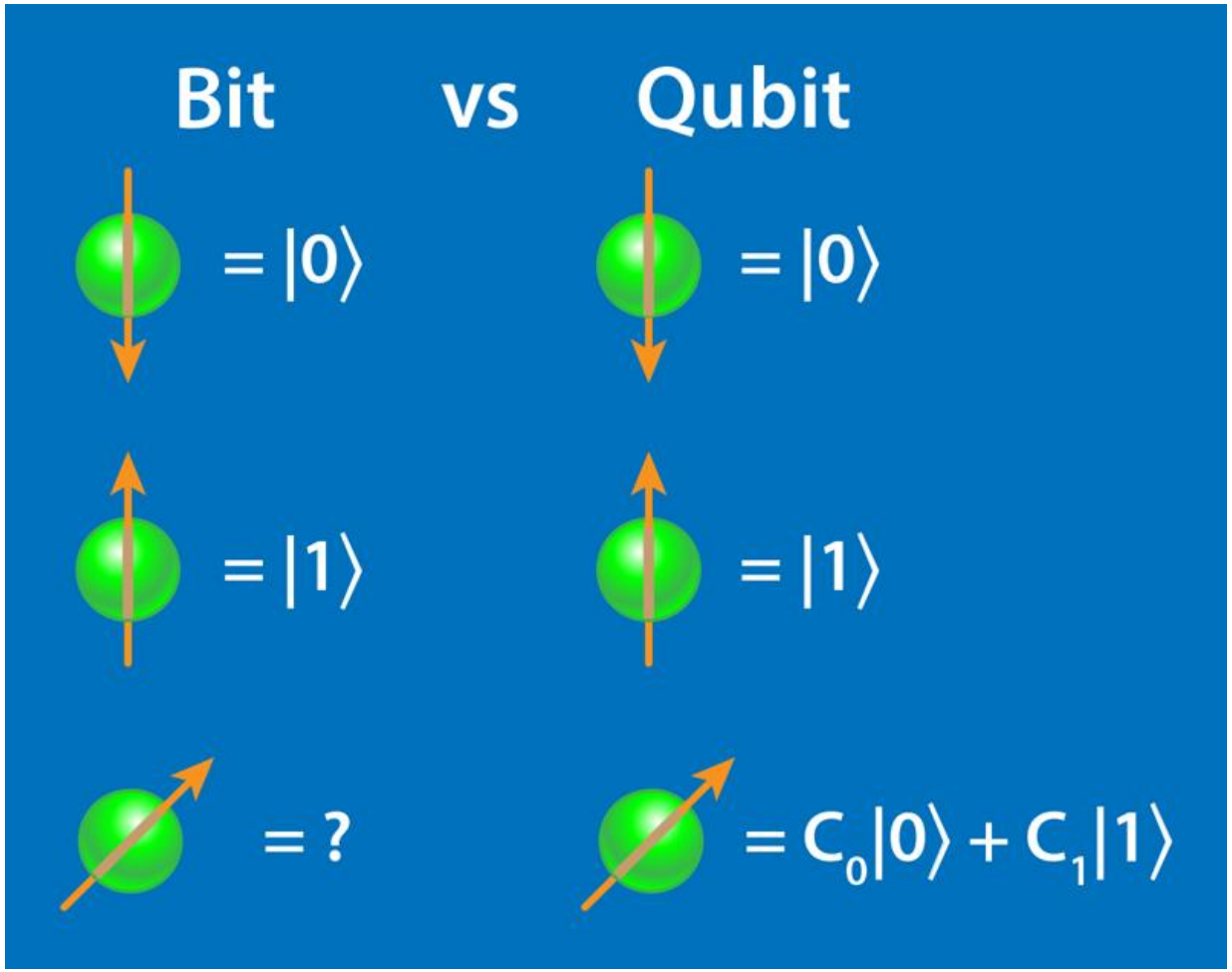


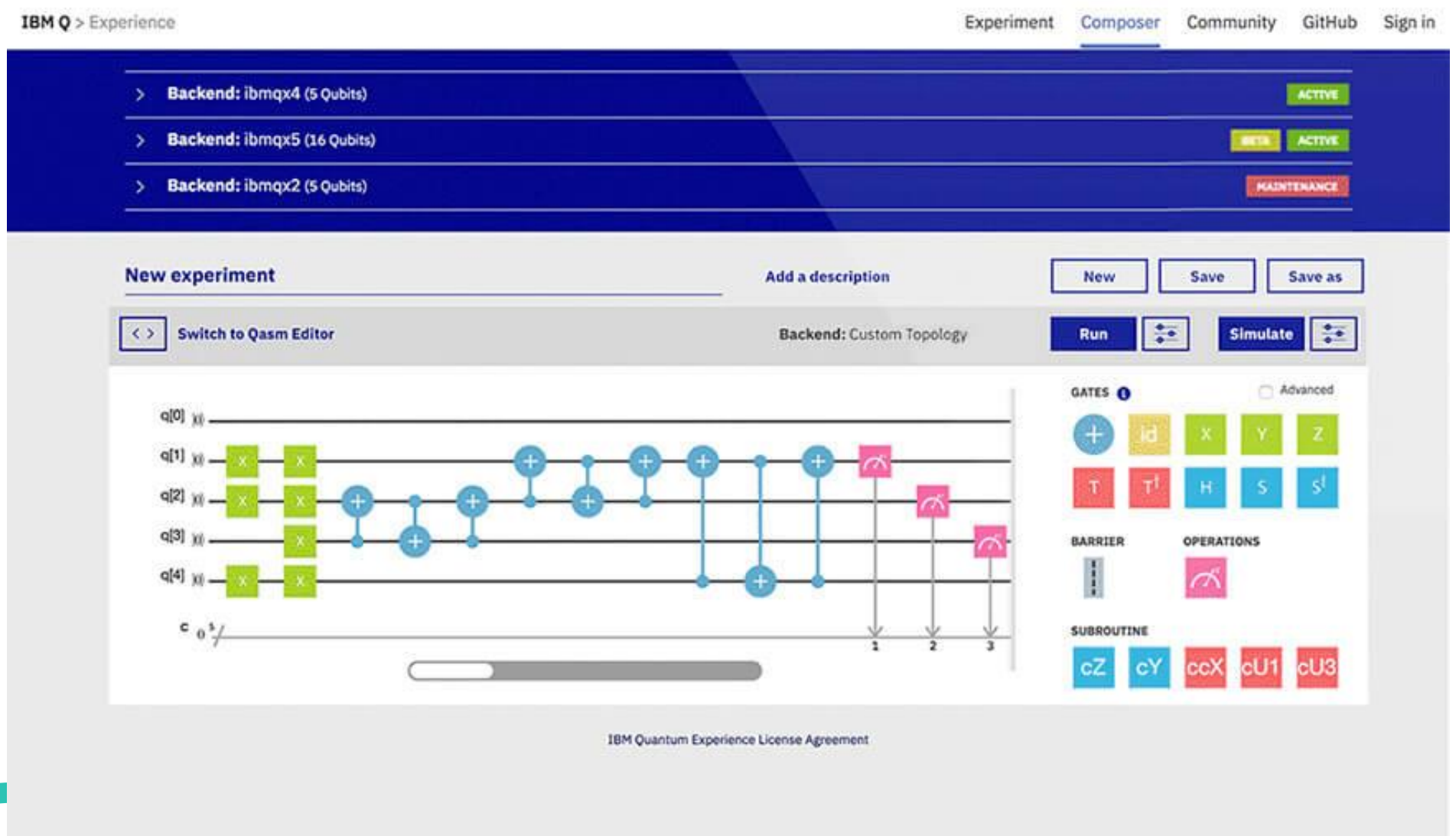
Klasik Bit

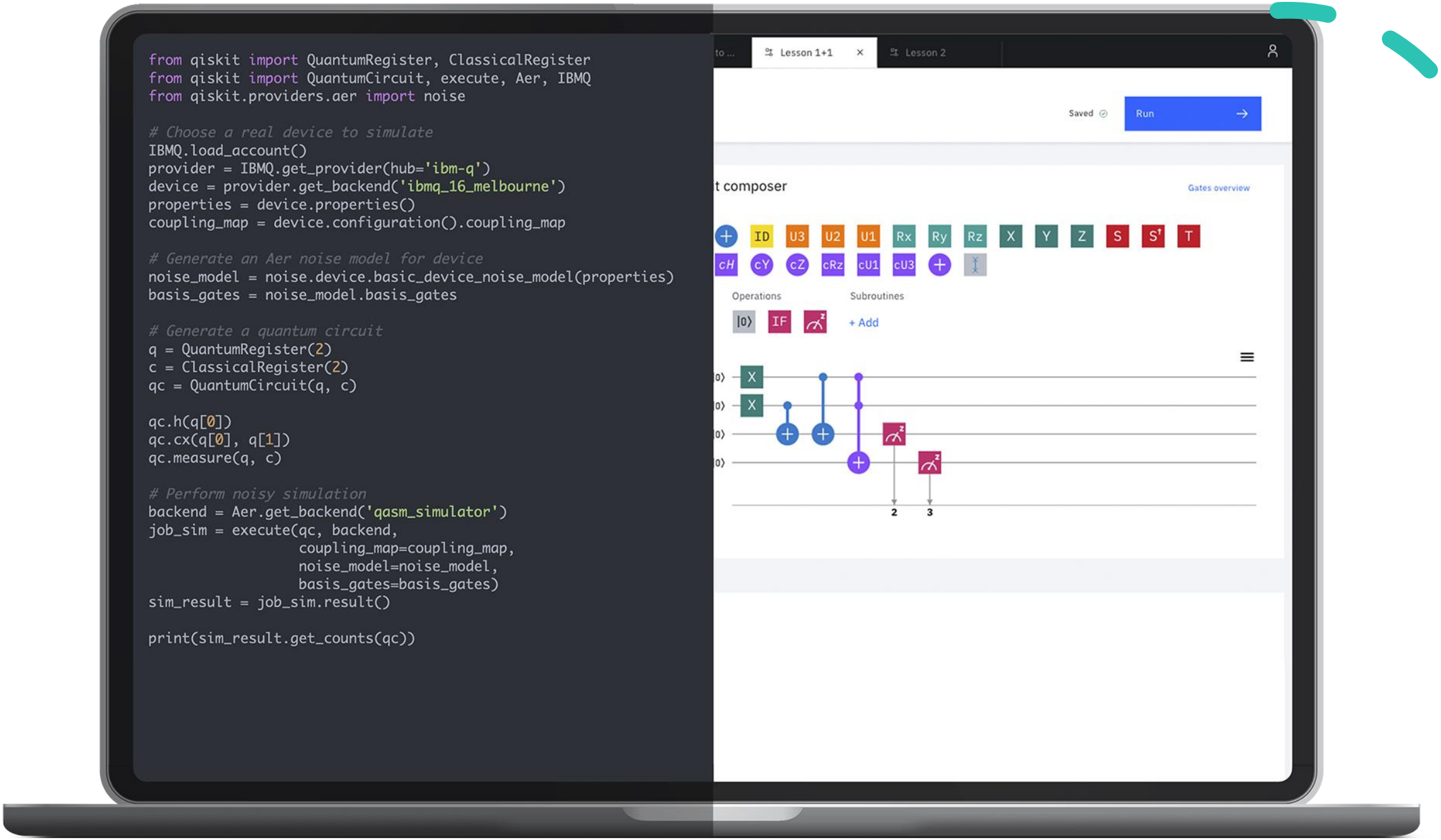


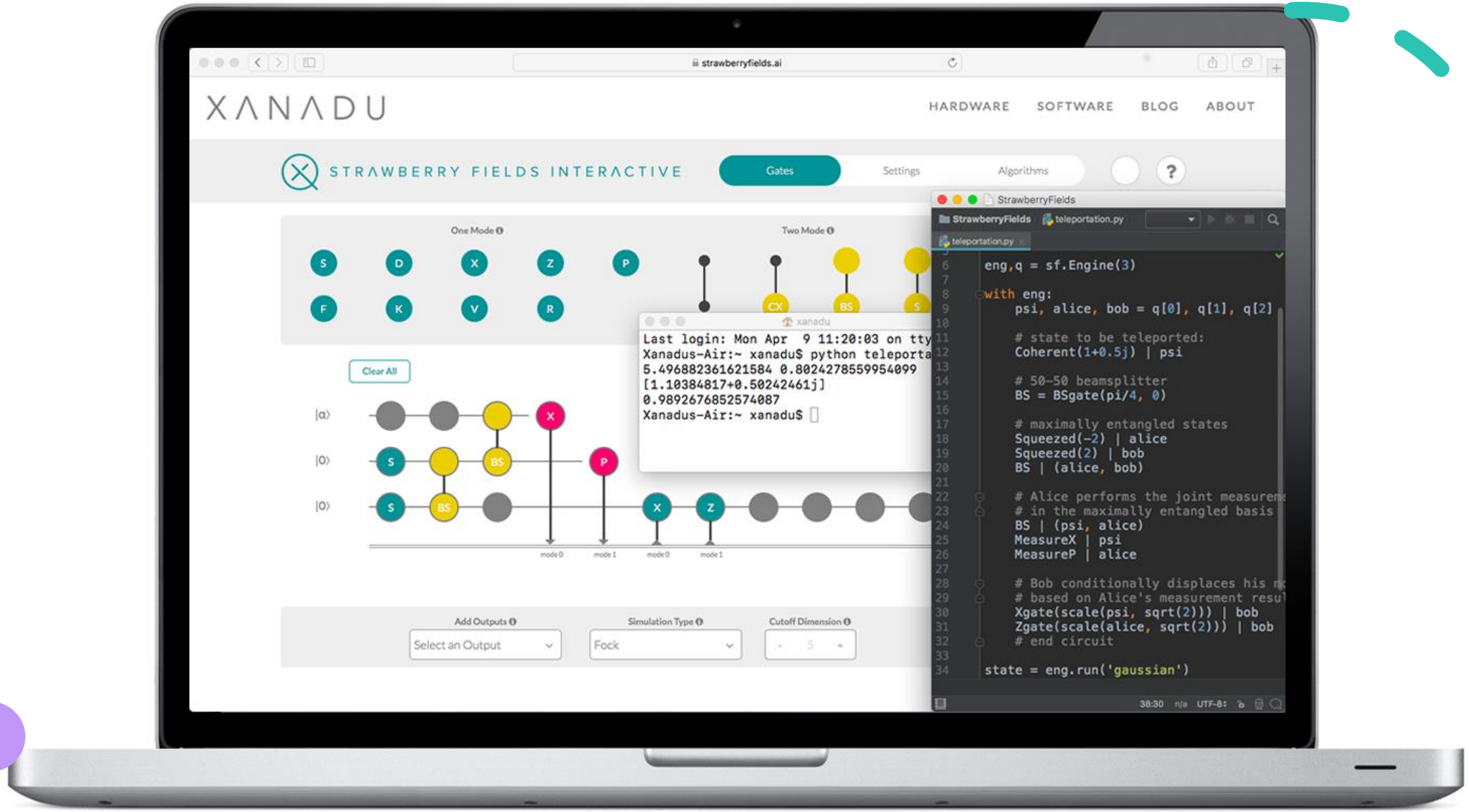
Kübit veya Kuantum Bit

- Klasik bitler ya 0 ya da 1 değerini alır. Ölçümden etkilenmez.
- Kübitler hem 0 hem 1 değerlerini alabilir. Ölçümden etkilenir.


Bit
vs
Qubit
= |0⟩
= |0⟩
= |1⟩
= |1⟩
= ?
= C0|0⟩ + C1|1⟩


IBM Q > Experience
Experiment
Composer
Community
GitHub
Sign in
Backend: ibmqx4 (5 Qubits)
ACTIVE
Backend: ibmqx5 (16 Qubits)
BETA
ACTIVE
Backend: ibmqx2 (5 Qubits)
MAINTENANCE
New experiment
Add a description
New
Save
Save as
Switch to Qasm Editor
Backend: Custom Topology
Run
Simulate
q[0]
q[1]
q[2]
q[3]
q[4]
c
GATES
Advanced
id
X
Y
Z
T
T†
H
S
S†
BARRIER
OPERATIONS
SUBROUTINE
cZ
cY
ccX
cU1
cU3
IBM Quantum Experience License Agreement


from qiskit import QuantumRegister, ClassicalRegister
from qiskit import QuantumCircuit, execute, Aer, IBMQ
from qiskit.providers.aer import noise
# Choose a real device to simulate
IBMQ.load_account()
provider = IBMQ.get_provider(hub='ibm-q')
device = provider.get_backend('ibmq_16_melbourne')
properties = device.properties()
coupling_map = device.configuration().coupling_map
# Generate an Aer noise model for device
noise_model = noise.device.basic_device_noise_model(properties)
basis_gates = noise_model.basis_gates
# Generate a quantum circuit
q = QuantumRegister(2)
c = ClassicalRegister(2)
qc = QuantumCircuit(q, c)
qc.h(q[0])
qc.cx(q[0], q[1])
qc.measure(q, c)
# Perform noisy simulation
backend = Aer.get_backend('qasm_simulator')
job_sim = execute(qc, backend,
coupling_map=coupling_map,
noise_model=noise_model,
basis_gates=basis_gates)
sim_result = job_sim.result()
print(sim_result.get_counts(qc))
Lesson 1+1
Lesson 2
Saved
Run
t composer
Gates overview
ID
U3
U2
U1
Rx
Ry
Rz
X
Y
Z
S
S†
T
cH
cY
cZ
cRz
cU1
cU3
Operations
Subroutines
|0⟩
IF
+ Add
X
X
2
3


strawberryfields.ai
XANADU
HARDWARE
SOFTWARE
BLOG
ABOUT
STRAWBERRY FIELDS INTERACTIVE
Gates
Settings
Algorithms
One Mode
Two Mode
S
D
X
Z
P
F
K
V
R
CX
BS
S
Clear All
|a⟩
|0⟩
|0⟩
X
S
BS
P
S
BS
X
Z
mode 0
mode 1
mode 0
mode 1
xanadu
Last login: Mon Apr 9 11:20:03 on tty
Xanadus-Air:~ xanadu$ python teleporta
5.496882361621584 0.8024278559954099
[1.10384817+0.50242461j]
0.9892676852574087
Xanadus-Air:~ xanadu$
Add Outputs
Select an Output
Simulation Type
Fock
Cutoff Dimension
5
StrawberryFields
StrawberryFields
teleportation.py
teleportation.py
eng,q = sf.Engine(3)
with eng:
psi, alice, bob = q[0], q[1], q[2]
state to be teleported:
Coherent(1+0.5j) | psi
# 50-50 beamsplitter
BS = BSgate(pi/4, 0)
# maximally entangled states
Squeezed(-2) | alice
Squeezed(2) | bob
BS | (alice, bob)
Alice performs the joint measurem
in the maximally entangled basis
BS | (psi, alice)
MeasureX | psi
MeasureP | alice
Bob conditionally displaces his m
based on Alice's measurement resu
Xgate(scale(psi, sqrt(2))) | bob
Zgate(scale(alice, sqrt(2))) | bob
# end circuit
state = eng.run('gaussian')
38:30
n/a
UTF-8:

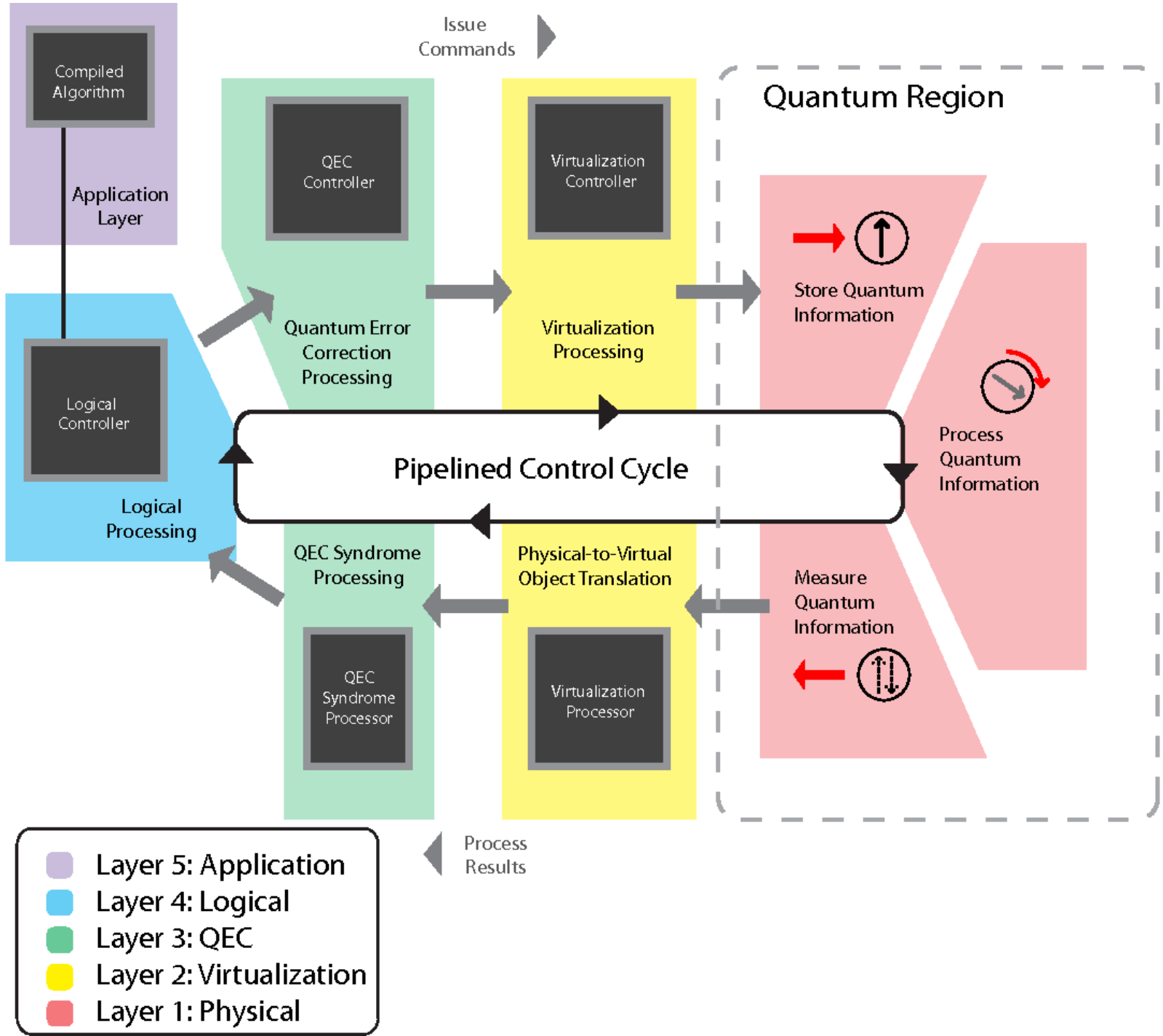
Kuantum Yazılım Nedir?

- **Kuantum algoritmalarını çalıştıran yazılımlar:** Kuantum yazılım geliştirme kitleri ve hesaplama platformları, son kullanıcılar için çözümler sağlar. Bunlar, son kullanıcıların kuantum algoritmaları geliştirmelerine ve test etmelerine yardımcı olur.

- Kuantum bilgisayarların çalışmasını sağlayan yazılımsal katmanlar: Kuantum bilgisayarlarda hatalar nedeniyle performans sorunları vardır ve bu tür hataları düzeltmek için hata düzeltme yazılımları oluşturulmuştur. Bir hata düzeltme yazılımı veya ürün yazılımı, kuantum bilgisayarların kararlılığını artıran düşük seviyeli bir programdır.

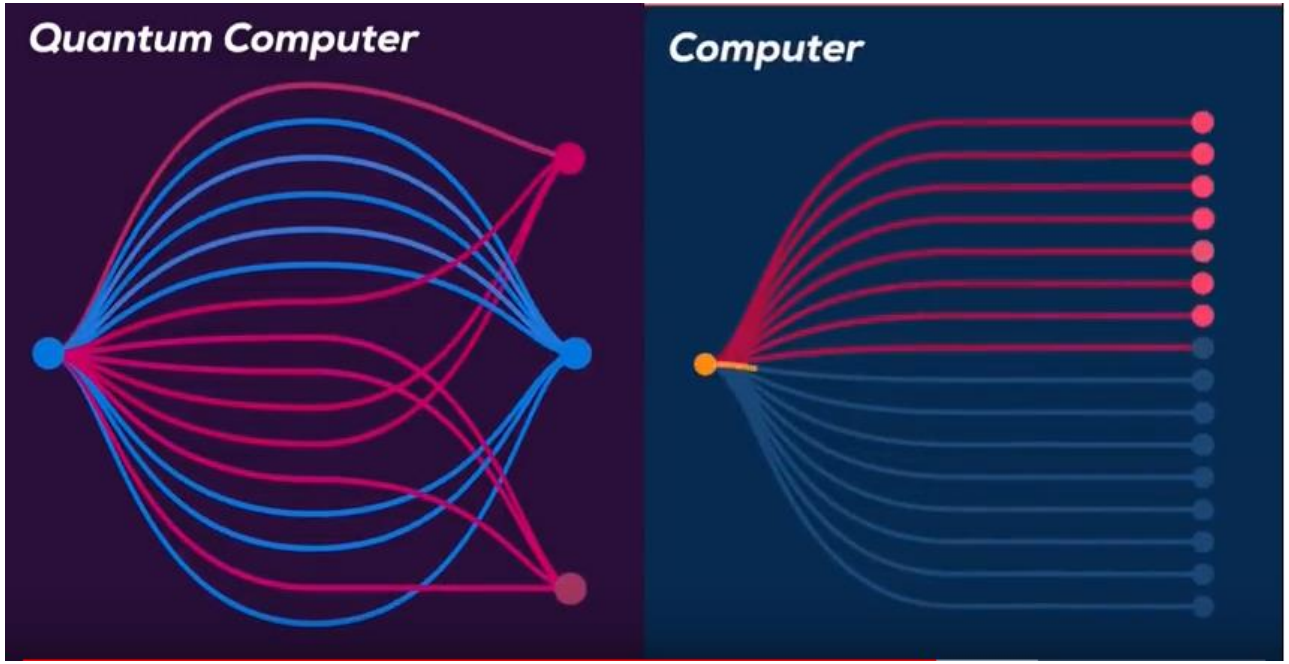
Kuantum bilgisayarların veya simülatörlerin çözebilmesi için bir problemin kuantum yazılım/program kullanılarak formüle edilmesi ve çevrilmesi gerekir. Farklı yazılım sistemlerinin çalışmasını sağlamak için farklı yöntemleri olabilir.

Quantum Dot Architecture



Klasik Programlama ile Farkları

- Programlama dilleri
- Fonksiyonalite
- Mimari



Kuantum Programlama Kütüphaneleri ve Dilleri


XANADU
D:wave
The Quantum Computing Company™
Microsoft
1QBit
Google
rigetti
IBM

Oyuncular

https://www.quantiki.org/wiki/quantum-programming-language

D-Wave	Ocean
Rigetti	Forest
IBM	IQ Experience/ Qiskit
Google	Cirq/ Quantum Playground
Microsoft	LIQUi |> / QDK
Zapata Computing	Orquestra
1QBit	1QBit SDK
Amazon	Braket SDK
Xanadu	Strawberry Fields/ Blackbird
Riverlane	DeltaFlow/ Anian
Qutech	Quantum Inspire
StrangeWorks	Quantum Computing Platform
QC Ware	Forge
Q-CTRL	Black Opal/ Boulder Opal/ Open Controls
Quantum Benchmark	True-Q

Qiskit
Elements for building a quantum future

Cirq

PENNY LANE

STRAWBERRY FIELDS

OpenFermion

TensorFlow Quantum

iskit

ardware

erra

ore, foundational tools for communicating with
ulators. Users can write quantum circuits,
e constraints with Terra. Its modular design
ons for quantum circuit optimizations and

Ignis

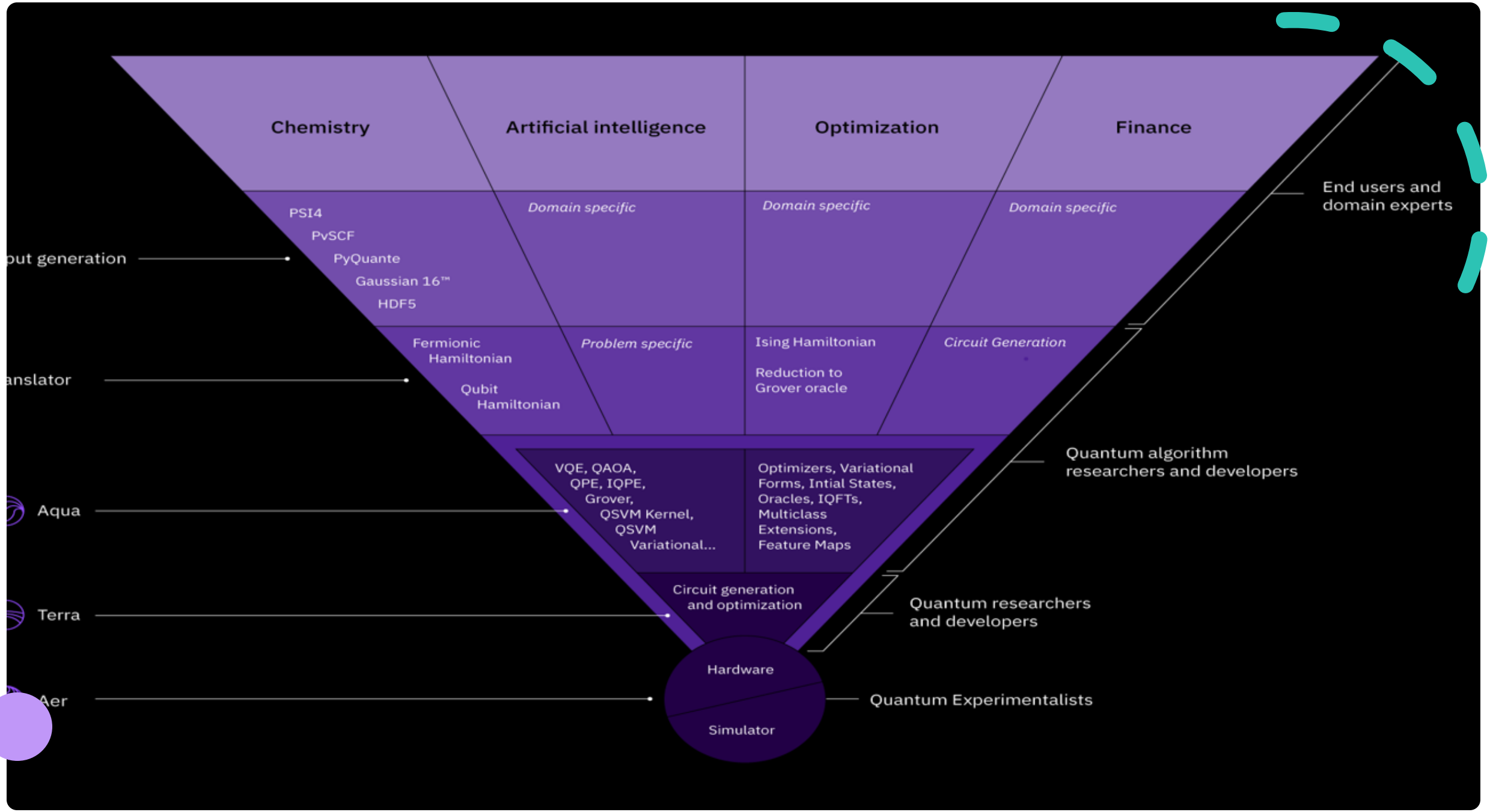
Controlling fire was a turning point in human evolution. Learning how to fix or control quantum errors will be a turning point in the evolution of quantum computing. Users can access better characterization of errors, improve gates, and compute in the presence of noise with Ignis. It is designed for researching and improving errors or noise in near-term quantum systems.

Aqua

Aqua is a modular and extensible library for experimenting with quantum algorithms on near-term devices. Users can build domain-specific applications, such as chemistry, AI and optimization with Aqua. It bridges quantum and classical computers by enabling classical programming to run on quantum devices.

Aer

Aer permeates all other Qiskit elements. Users can ac
quantum simulator and emulator research with Aer, w
better understand the limits of classical processors by
their ability to mimic quantum computation. Users car
current and near-term quantum computer functionalit

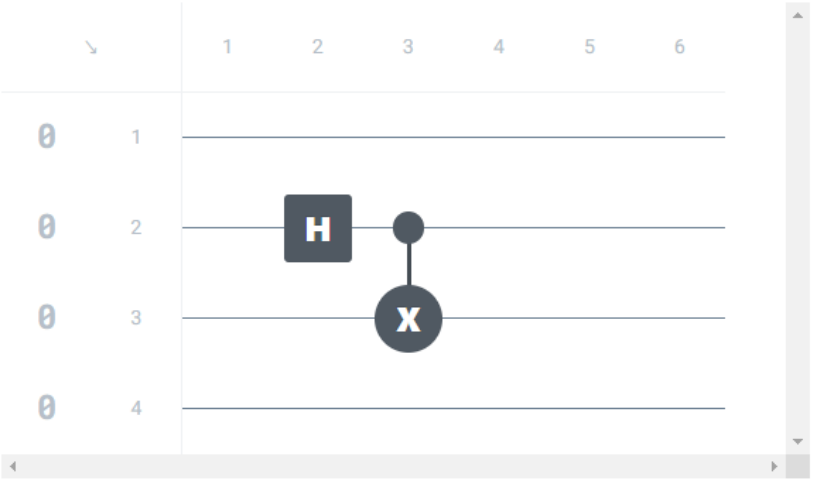

Chemistry
Artificial intelligence
Optimization
Finance
PSI4
PyQuante
Gaussian 16™
HDF5
Domain specific
Domain specific
Domain specific
Fermionic Hamiltonian
Qubit Hamiltonian
Problem specific
Ising Hamiltonian
Reduction to Grover oracle
Circuit Generation
VQE, QAOA, QPE, IQPE, Grover, QSVM Kernel, QSVM Variational...
Optimizers, Variational Forms, Intial States, Oracles, IQFTs, Multiclass Extensions, Feature Maps
Circuit generation and optimization
Hardware
Simulator
Aqua
Terra
Aer
End users and domain experts
Quantum algorithm researchers and developers
Quantum researchers and developers
Quantum Experimentalists

Quantum JavaScript Project (Q.js)

API documentation

Q is a quantum circuit simulator, drag-and-drop circuit editor, and powerful JavaScript library that runs right here in your web browser. There's nothing to install and nothing to configure, so jump right in and experiment. (Q recently celebrated our one-year anniversary. You can read the corresponding post on Medium, the discussion on Hacker News, and the thread on Reddit.)

Here's your first quantum circuit—a Bell state. It uses superposition and entanglement to calculate. (And here's how to make one yourself.) Tap and drag the tiles around to get a feel for the Q editor. It's easy to use on both desktop and mobile devices. Made a mistake? Just tap the Undo button.



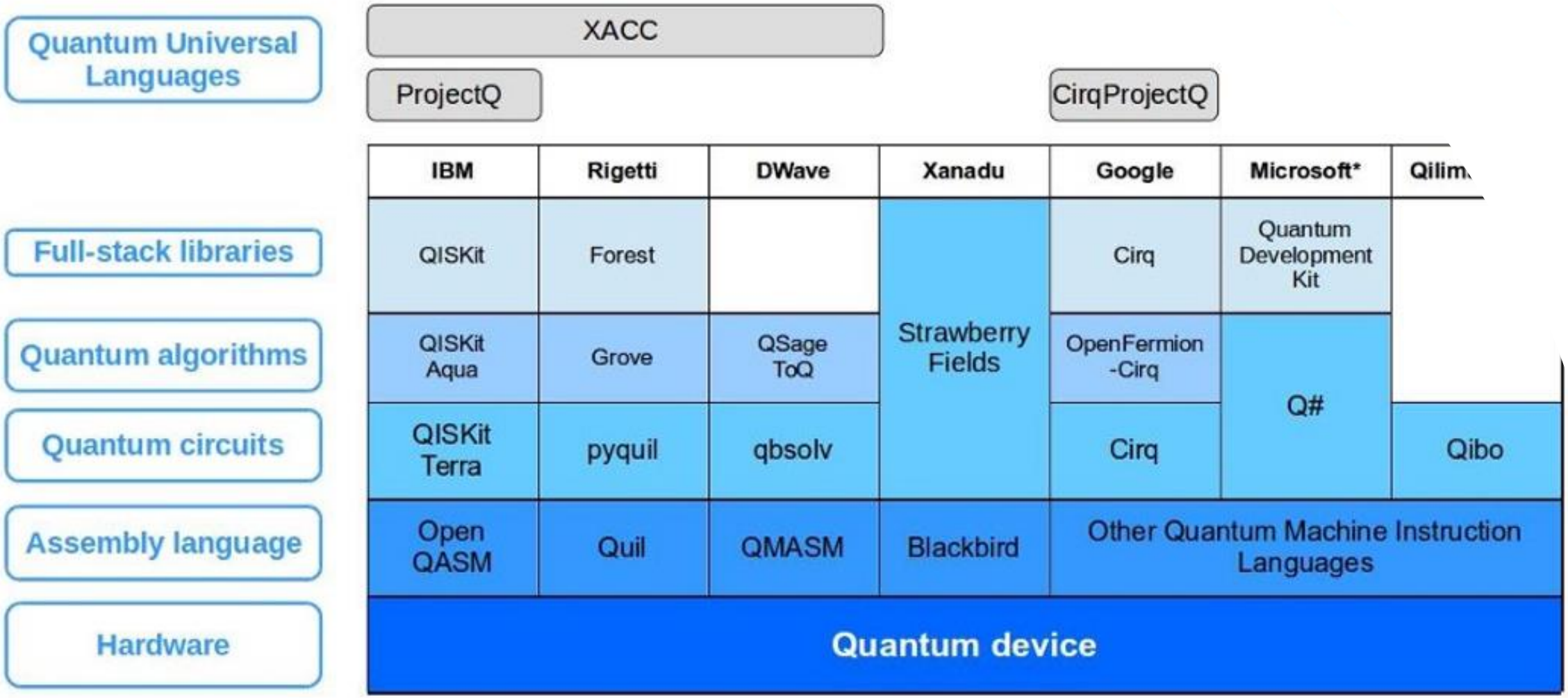
This circuit is accessible in your JavaScript console as `document.getElementById('example').circuit`

- **Quantum instruction sets**
- Quil - An instruction set architecture for quantum computing that first introduced a shared quantum/classical memory model.
  OpenQASM - The intermediate representation introduced by IBM for use with their Quantum Experience.
- **Quantum programming languages**
- **Imperative languages**
- **QCL**
- **Quantum pseudocode**
- **Q|SI>**
- **Q language** -
- **qGCL**
- **QMASM**

- **Functional languages**
- **QFC and QPL**
- **QML**
- **LIQUi|>**
- **Quantum lambda calculi**
- **Quipper**

- **Multi-Paradigm languages**
- **Silq**
- **Q# (Q Sharp)** -
- Strawberry Fields

Quantum Computing Programming Langu



* Hardware under development. Quantum programs are run on their own simulators.

"Quantum Language" is refered with no distinction both as a quantum equivalence of a programming language and as a library to write quantum programs supported by some well-known classical programming language.

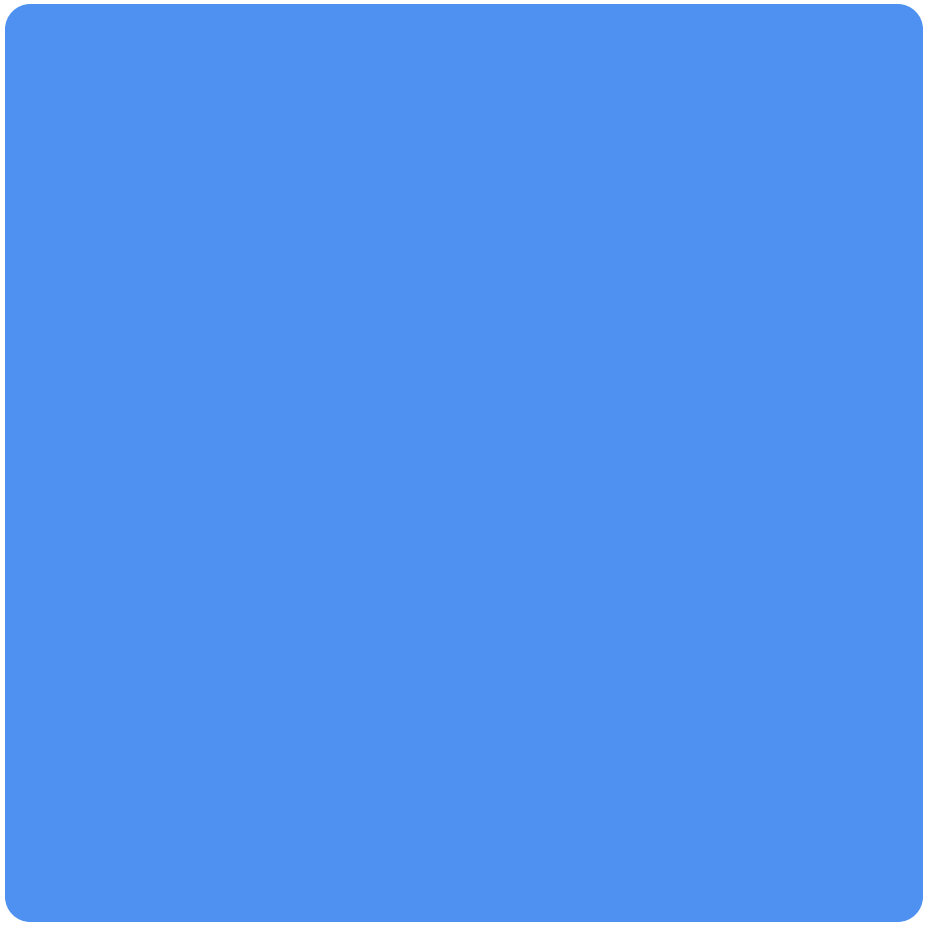
© Alba Cervera-Lierta for the QWA (2018)

Kaynak
Introduction to Quantum Mechanics, David J. Griffiths, Darrell F. Schroeter | (Phys)
Quantum Computing Since Democritus, Scott Aaronson | (QC)
Quantum Computing for Computer Scientists, Noson S. Yanofsky | (QC)
Quantum Computing for Everyone, Chris Bernhardt | (PopSci)
Programming Quantum Computers: Essential Algorithms and Code Samples, Eric R. Johnston, Nic Harrigan, Mercedes Gimeno-Segovia | (QC)
Quantum Computation and Quantum Information, Isaac Chuang, Michael Nielsen | (Phys)
The Quantum Moment: How Planck, Bohr, Einstein, and Heisenberg Taught Us to Love Uncertainty, Robert P. Crease, Alfred Scharff Goldhaber | (History)

QWorld/QTurkey Workshop

QWORLD

https://qworld.lu.lv/index.php/workshop-bronze/



- Bronze Material: https://github.com/KuantumTurkiye/bronze
- Eylül '20: https://qworld.lu.lv/index.php/qbronze36-qturkey-september-2020/
- Kasım '20: https://qworld.lu.lv/index.php/qbronze43-qturkey-november-2020/
- ….

Neler Yapılabilir?

Teşekkürler...