QUANTINUUM

Quantum Circuit Compilation and Classical Control with TKET: Part I

Presented by:

Callum Macpherson & Lewis Wright

19/09/2023

Nice to meet you!

Callum Macpherson, MPhys

Quantum software: Technical support & outreach

Quantum optics & atomic physics

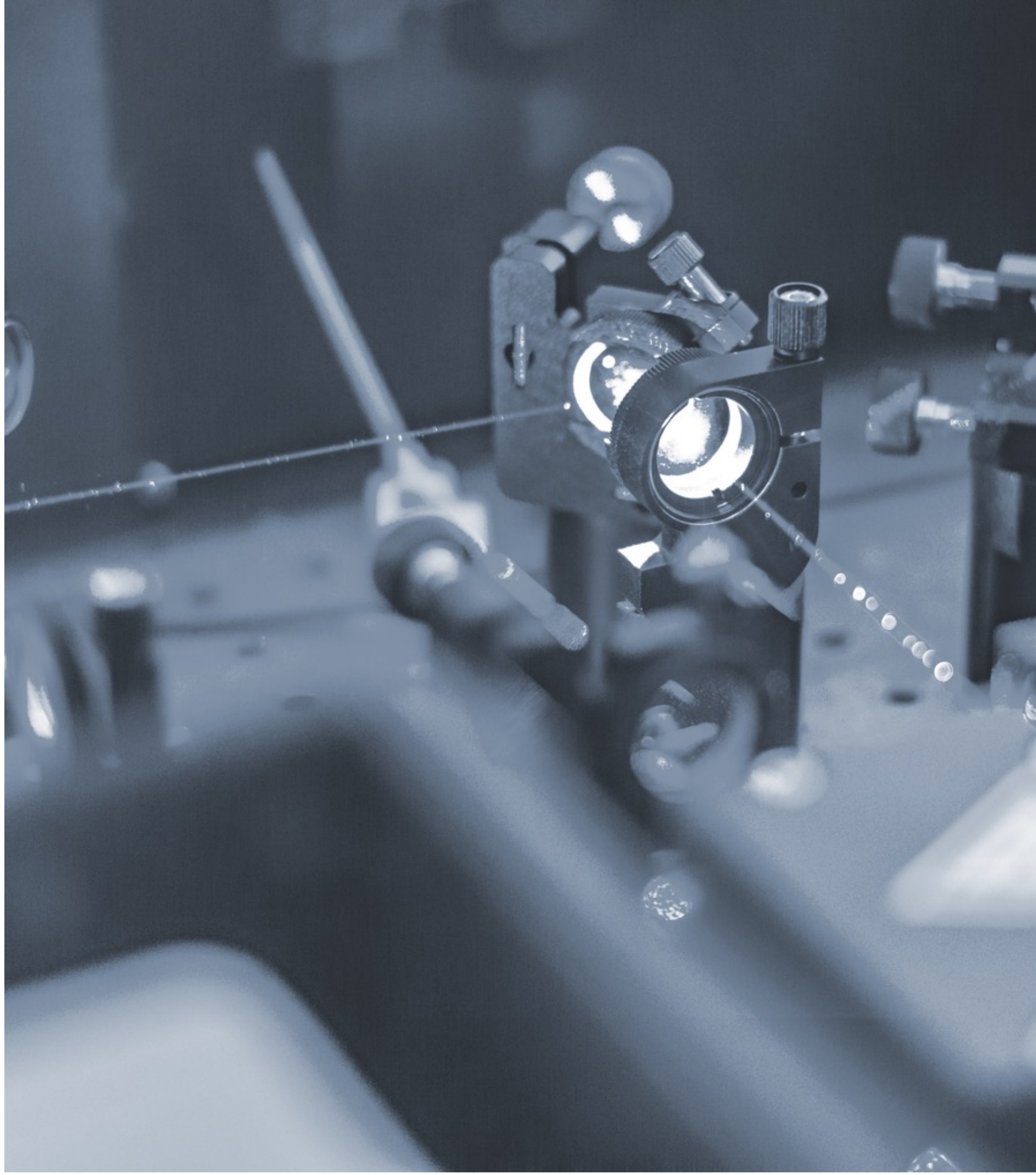
Lewis Wright, PhD

Quantum algorithms scientist

Physically motivated quantum algorithms & tensor network methods

Agenda: Part I

- Tutorial 1: Introduction slides (10 minutes)
    - Quantum software.
    - What is TKET?
    - Quantum compilation
- TKET 101: Basic concepts (40 minutes)
    - Constructing circuits
    - Backends
    - New features
- Practical application: PDE solver (40 minutes)
    - Converting Parameterised Quantum Circuit to TKET
    - Circuit compilation
    - Converting to different native gatesets.
    - Noisy simulations.



Tutorial 1 & 2 notebooks:

Quantum Software

- General purpose SDKs - qiskit, Cirq, pytket*
- Quantum Programming languages/high level languages - Q#, Silq, Quipper
- Compiler - TKET, qiskit, BSQKit
- Online services - AWS Bracket
- Quantum Error Correction/Mitigation- Qermit, others
- Application libraries - e.g. InQuanto, pennylane
- Simulators e.g. Qulacs, Stim

Qiskit

SILQ

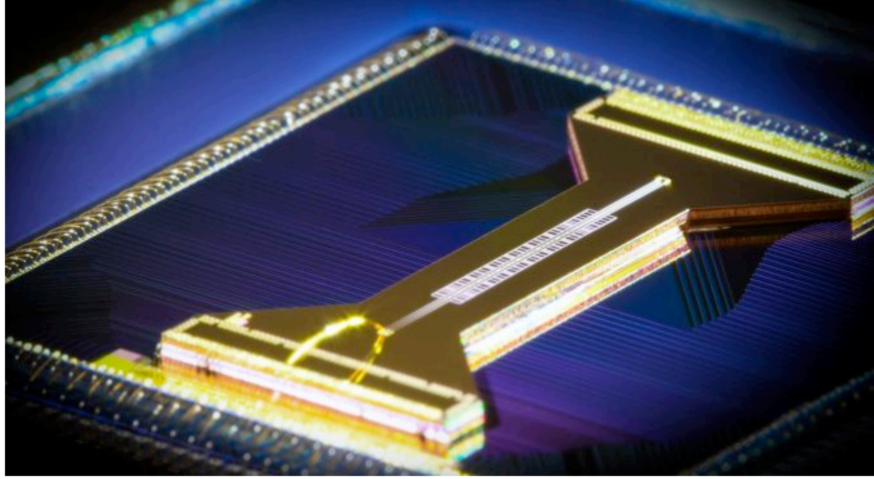
TKET

PENNYLANE

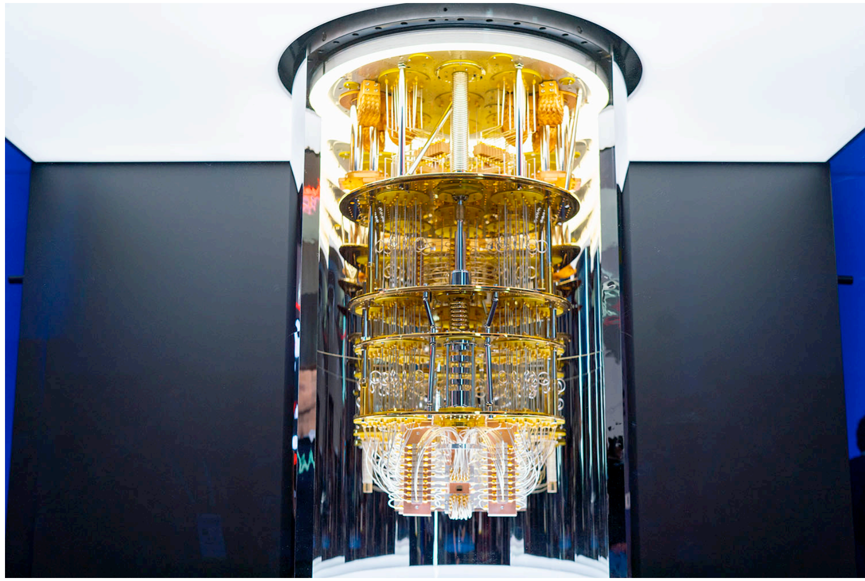
INQUANTO™ Computational Chemistry

Quantum Hardware

- Trapped ions - Quantinuum, IONQ, AQT
- Superconductors - IBM, Google, Rigetti, IQM
- Photonics - PsiQuantum, Quandela…
- Neutral atoms - Pasqal, Infleqtion…
- Others - Semiconductors, topological qubits…



H-series Ion traps



Superconducting circuits- IBM

Current Challenges with Quantum computing

- Not enough qubits for many of the exciting applications
- The qubits we do have are subject to complex noise (hard to model)
- Quantum error correction at an early stage experimentally
- **Low-level details greatly influence performance** - gate count/depth, connectivity

What is TKET?

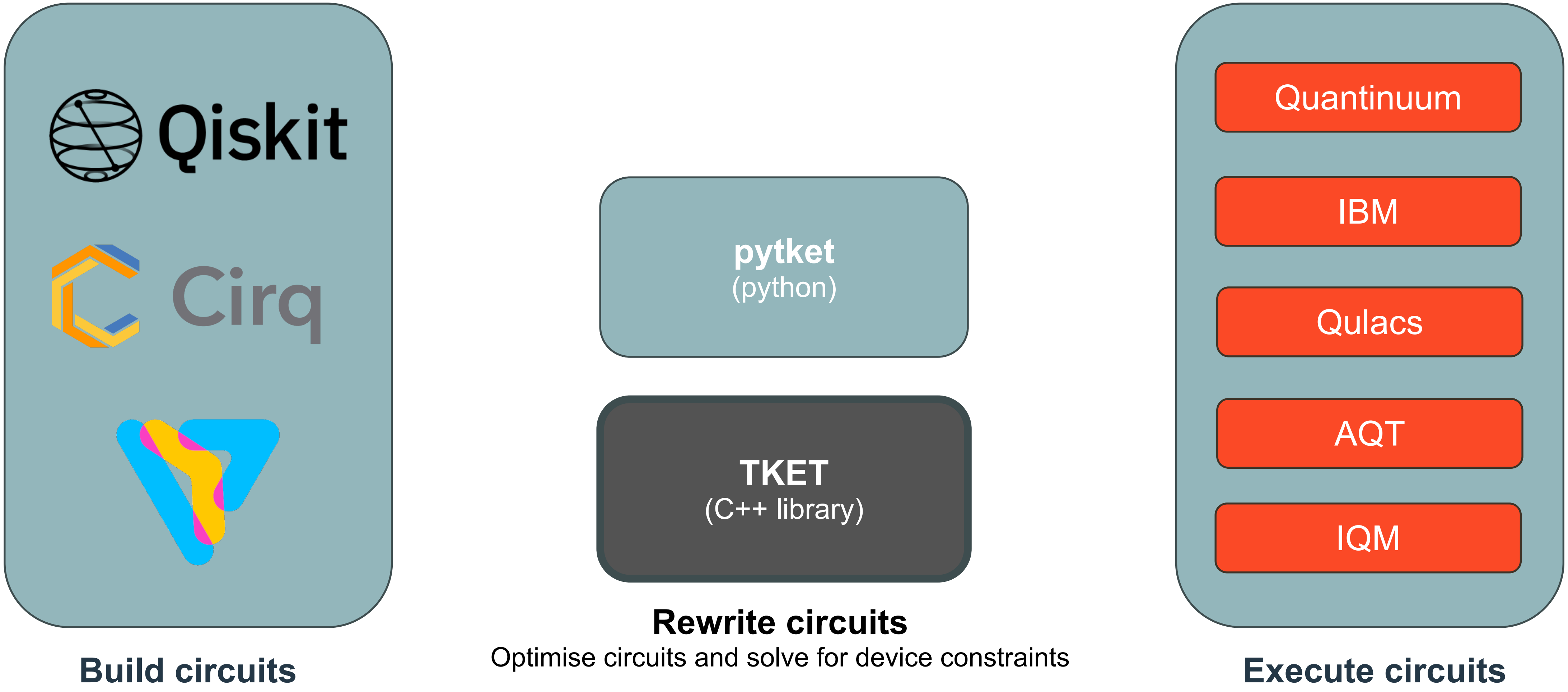
TKET is a quantum software library developed by Quantinuum:

- A high performance quantum compiler
- Open source! https://github.com/CQCL/tket
- “Hardware agnostic” - Targets a range of devices and simulators
- Works with popular libraries - Qiskit, Cirq, Braket, Pennylane + more

```
$ pip install pytket
```

TKET Architecture

Note: Cloud access through Microsoft Azure and AWS Braket is also available



Qiskit

Cirq

pytket
(python)

TKET
(C++ library)

Quantinuum

IBM

Qulacs

AQT

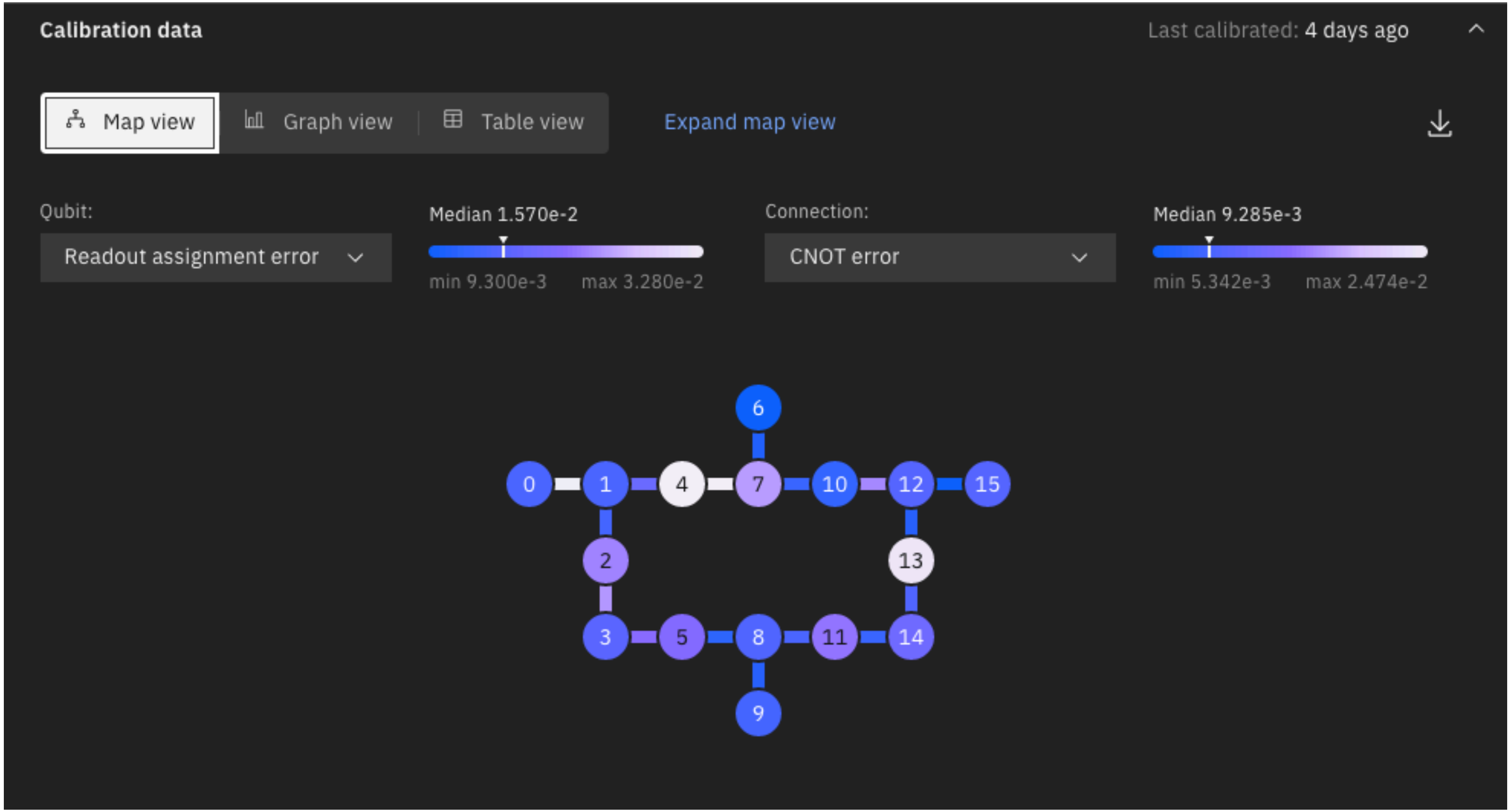
IQM

Build circuits

Rewrite circuits
Optimise circuits and solve for device constraints

Execute circuits

A Real Quantum device



Calibration data

Last calibrated: 4 days ago

Qubit: Readout assignment error

Median 1.570e-2

min 9.300e-3 max 3.280e-2

Connection: CNOT error

Median 9.285e-3

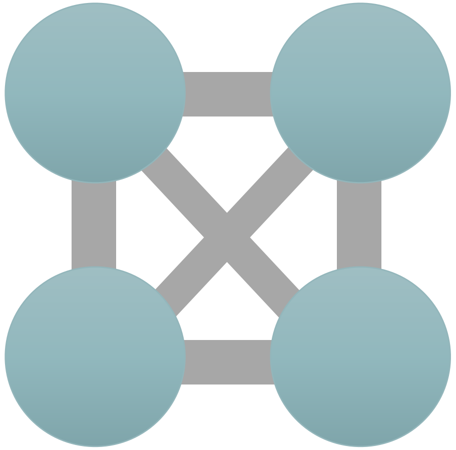
min 5.342e-3 max 2.474e-2

Source: IBM Quantum

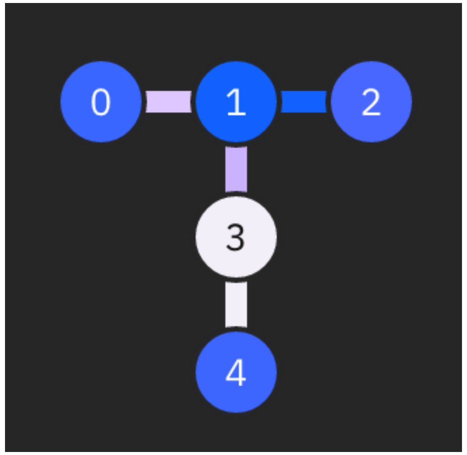
Quantum Compilation I

Target device: IBMQ Belem

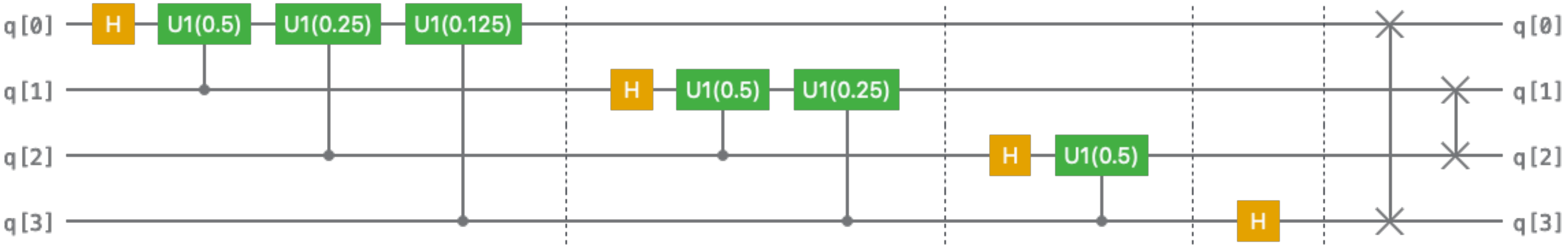
- Nearest neighbour interaction only
- Limited gate set {X, SX, Rz, CNOT}
- CNOT error



Complete connectivity graph



Belem qubit topology

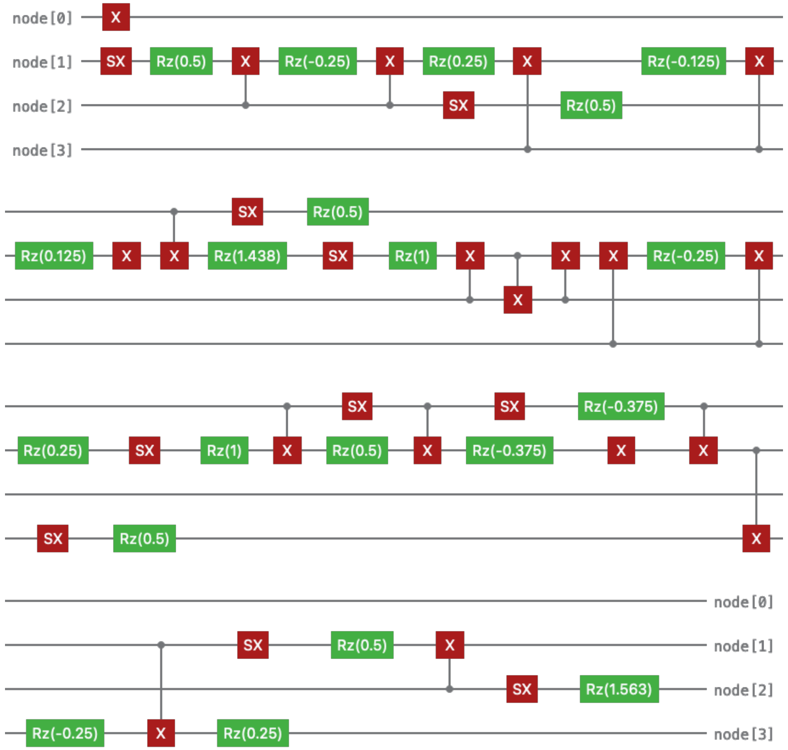


Quantum Fourier transform circuit (hardware independent)

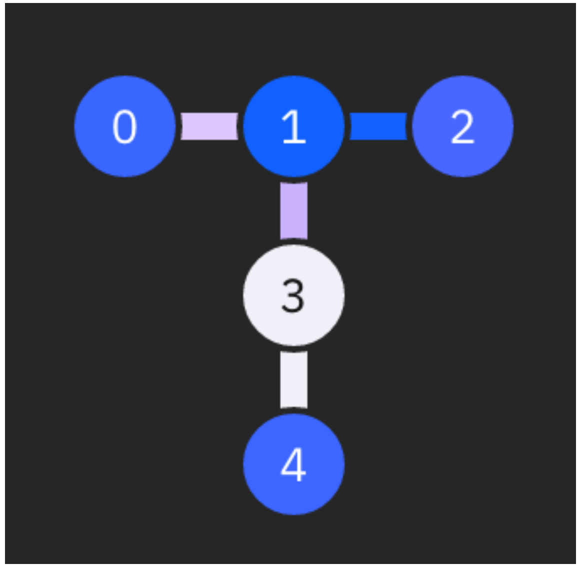
Quantum Compilation II

- Circuit is in IBM native gateset

- Each qubit is assigned to a physical node of the device

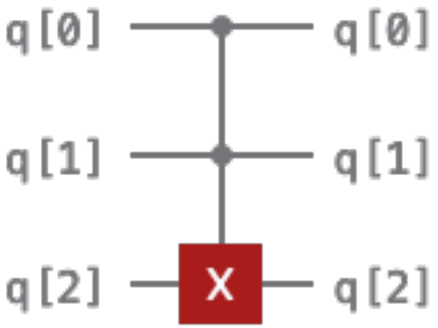


Compiled quantum Fourier transform with native gates {X, SX, Rz, CNOT}

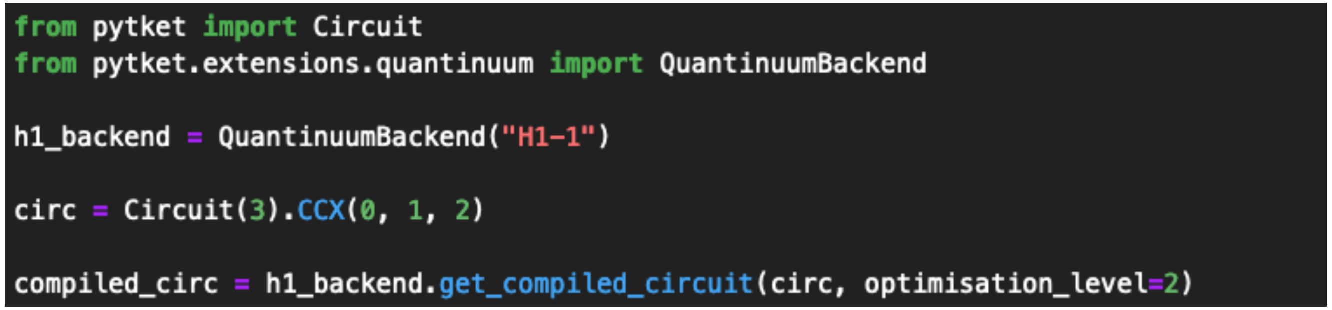


Belem qubit topology

Quantum Compilation III (CCX gate)



CCX gate



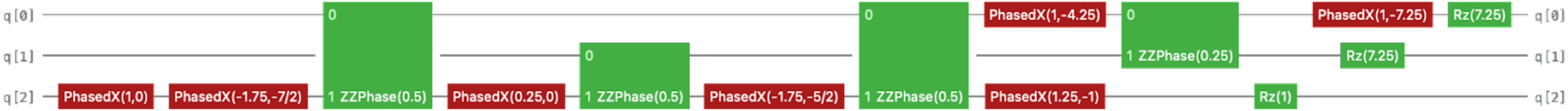
```
from pytket import Circuit
from pytket.extensions.quantinuum import QuantinuumBackend

h1_backend = QuantinuumBackend("H1-1")

circ = Circuit(3).CCX(0, 1, 2)

compiled_circ = h1_backend.get_compiled_circuit(circ, optimisation_level=2)
```

pytket code



CCX gate compiled to Quantinuum’s H-Series gate set

TKET 101: Basic Concepts Notebook

QUANTINUUM