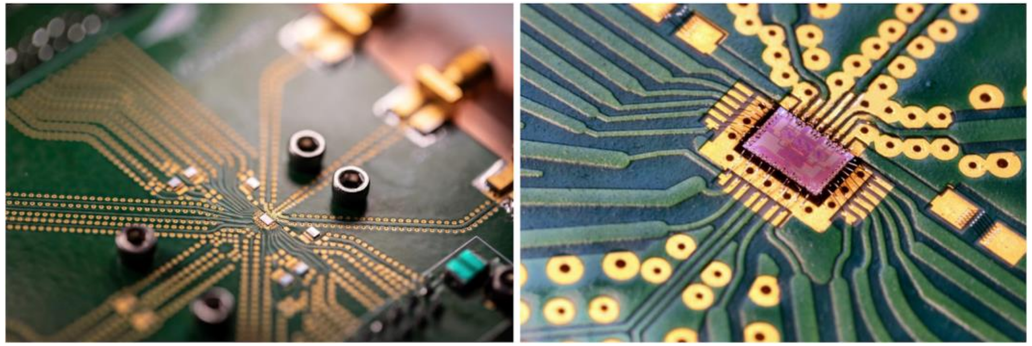
QUANTUM COMPUTING WITHOUT A QUANTUM COMPUTER

Dr Oliver Thomson Brown
EPCC, University of Edinburgh
o.brown@epcc.ed.ac.uk

Introduction

- Quantum computing is coming!
  - If you work at Rigetti / Google / IBM / Honeywell it's already here…

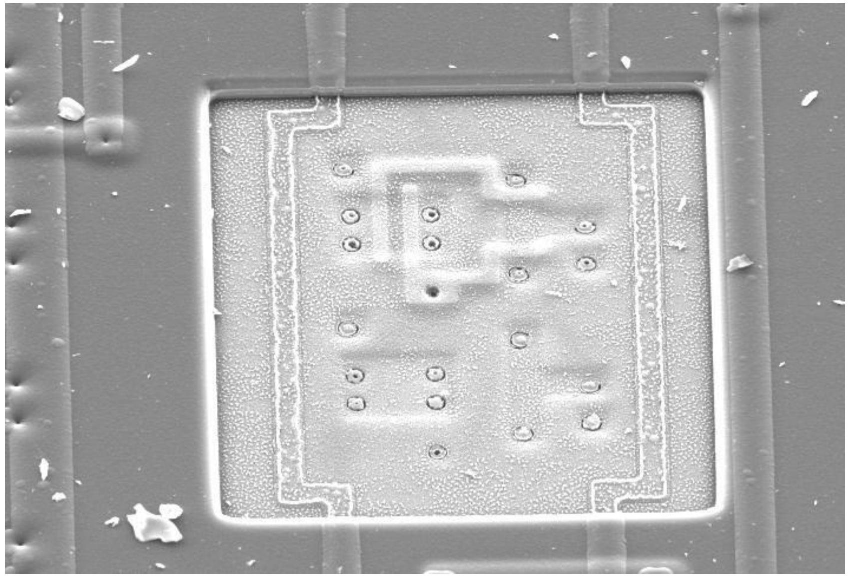


Google Sycamore quantum processor.
Source: https://ai.googleblog.com/2019/02/on-path-to-cryogenic-control-of-quantum.html

Cryostat surrounding Google Sycamore quantum processor.
Source: https://phys.org/news/2020-08-google-largest-chemical-simulation-quantum.html

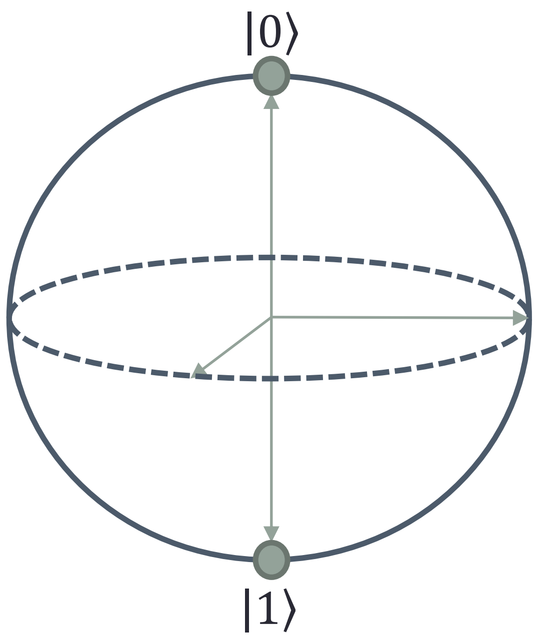
Classical Bits

- Programmers have an abstract model of the hardware on which their programs are executed.
  - Or rather, they have lots of them, depending on the high-level features of the hardware…
  - The result of 70 years worth of research and development!
- At the lowest level we have the concept of a ‘bit’.
  - We build datatypes on bits, and data structures on datatypes.
- As a software developer, **I don’t care how a bit is implemented.**



CMOS 2 input NOR gate with passivation and upper metal layers removed (3 micron CMOS).
Source: http://www2.eng.cam.ac.uk/~dmh/4b7/resource/mesp.htm

Quantum Bits



The Bloch sphere.

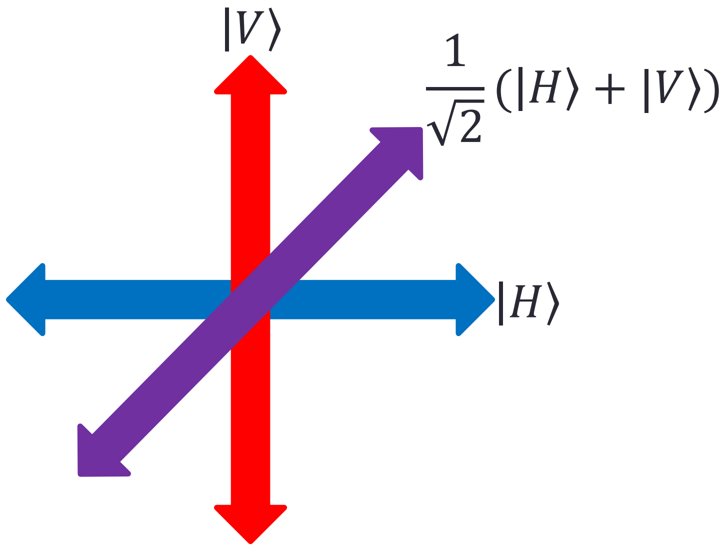
- A classical bit has two states, usually referred to as ‘0’ and ‘1’.
- A quantum bit (‘qubit’) has two states, usually referred to as $|0\rangle$ and $|1\rangle$. But…

Superposition Principle

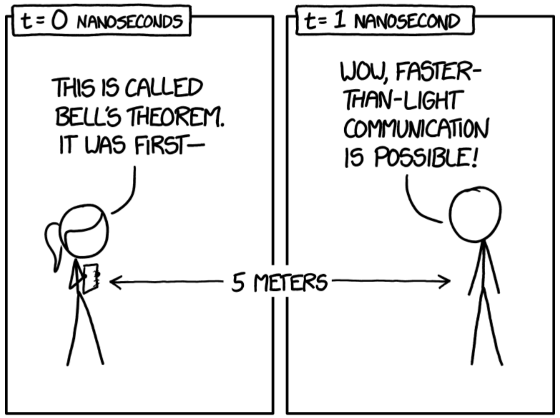
$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$$

Superposition

- How does that work?
  - Well it depends on the underlying physical implementation!
  - One example is polarised light, used in QKD.
- As quantum software developers, we try not to worry about it.
- Crucially, the superposition principle applies to many-body states as well as individual qubits.
  - $|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$



Entanglement



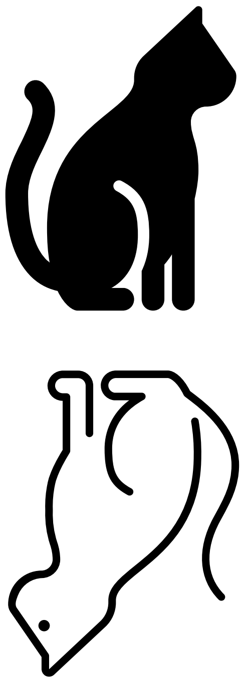
BELL'S SECOND THEOREM: MISUNDERSTANDINGS OF BELL'S THEOREM HAPPEN SO FAST THAT THEY VIOLATE LOCALITY.

Source: https://xkcd.com/1591/

- That state, $|\Phi^{+}\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$, has another interesting property. It is *maximally entangled*.
  - It is one of the 4 Bell states.
- Understanding entanglement would be a whole talk in itself…
- Key point for us is that entangled states **are not separable**.
  - $|\Phi^{+}\rangle \neq |A\rangle|B\rangle$

Measurement

- Measurement of quantum systems is another big topic.
- Result of a measurement on a single qubit is either $|0\rangle$ or $|1\rangle$.
  - The state $|\Psi\rangle = \alpha|0\rangle + \beta|1\rangle$ would be measured in state $|0\rangle$ or $|1\rangle$ with probability $|\alpha|^2$ and $|\beta|^2$ respectively.
  - Building up a more detailed picture of the original state requires repeated measurements.
- **Measurement changes the state of the system**.
  - It’s now in the state you measured it in.



Quantum Circuits

Operator	Gate(s)	Matrix
Pauli-X (X)	X	$\begin{bmatrix}0 & 1\\1 & 0\end{bmatrix}$
Pauli-Y (Y)	Y	$\begin{bmatrix}0 & -i\\i & 0\end{bmatrix}$
Pauli-Z (Z)	Z	$\begin{bmatrix}1 & 0\\0 & -1\end{bmatrix}$
Hadamard (H)	H	$\frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\1 & -1\end{bmatrix}$
Phase (S, P)	S	$\begin{bmatrix}1 & 0\\0 & i\end{bmatrix}$
$\pi/8$ (T)	T	$\begin{bmatrix}1 & 0\\0 & e^{i\pi/4}\end{bmatrix}$
Controlled Not (CNOT, CX)		$\begin{bmatrix}1 & 0 & 0 & 0\\0 & 1 & 0 & 0\\0 & 0 & 0 & 1\\0 & 0 & 1 & 0\end{bmatrix}$
Controlled Z (CZ)	Z	$\begin{bmatrix}1 & 0 & 0 & 0\\0 & 1 & 0 & 0\\0 & 0 & 1 & 0\\0 & 0 & 0 & -1\end{bmatrix}$
SWAP		$\begin{bmatrix}1 & 0 & 0 & 0\\0 & 0 & 1 & 0\\0 & 1 & 0 & 0\\0 & 0 & 0 & 1\end{bmatrix}$
Toffoli (CCNOT, CCX, TOFF)		$\begin{bmatrix}1 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\0 & 1 & 0 & 0 & 0 & 0 & 0 & 0\\0 & 0 & 1 & 0 & 0 & 0 & 0 & 0\\0 & 0 & 0 & 1 & 0 & 0 & 0 & 0\\0 & 0 & 0 & 0 & 1 & 0 & 0 & 0\\0 & 0 & 0 & 0 & 0 & 1 & 0 & 0\\0 & 0 & 0 & 0 & 0 & 0 & 0 & 1\\0 & 0 & 0 & 0 & 0 & 0 & 1 & 0\end{bmatrix}$

- How do we program a quantum computer then?
  - By writing a *quantum circuit*.
- We have some quantum register of qubits, and apply gates, like those on the left.
  - We may also have a classical register.
- OpenQASM quantum assembly language is in development, enabling imperative programming.
  - https://github.com/Qiskit/openqasm

Quantum Computer Emulation

- Qubits are represented by a vector of complex numbers.
  - Each element is the *complex amplitude* for that state.
  - Complex numbers are represented by two floating-point values.
  - Generally double precision is required.
  - One qubit needs two complex doubles.
    - $2 \times 2 \times 8 \text{ bytes} = 32 \text{ bytes}.$
- Single-qubit gates (also known as operators) are represented by 2×2 matrices.
  - Gates are applied to the qubit by calculating the matrix-vector product – the result is the new statevector.
  - Operators are also complex valued, so four complex doubles for a single-qubit gate.
    - $4 \times 16 \text{ bytes} = 64 \text{ bytes}$

$$|\Psi\rangle = \alpha|0\rangle + \beta|1\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}$$

$$\hat{O} = \begin{pmatrix} O_{00} & O_{01} \\ O_{10} & O_{11} \end{pmatrix}$$

$$|\Psi'\rangle = \hat{O}|\Psi\rangle = \begin{pmatrix} O_{00}\alpha + O_{01}\beta \\ O_{10}\alpha + O_{11}\beta \end{pmatrix}$$

The Scaling Problem

- Two classical bits can be represented by two numbers: 00, 01, 10, 11.
  - The composite state is a separable product of the individual bits.
  - Representation scales like N.
- Recall what we said about the superposition principle, and entanglement…
  - Any linear combination of any possible state of the composite system is valid.
  - In fact, there are valid composite states that **cannot** be represented as a separable product of individual qubit states.
- We need a complex number for every possible composite state.
  - Representation scales like 2^N.

$$|\Psi\rangle = \alpha|00\rangle + \beta|01\rangle + \gamma|10\rangle + \delta|11\rangle$$
$$= \begin{pmatrix} \alpha \\ \beta \\ \gamma \\ \delta \end{pmatrix}$$

Compression

- Why not just use compression?
  - Good news for operators (gates) – they're typically sparse, and generally get *more* sparse as they get larger. They might even have a simple enough structure that we don't need to write them out at all!
  - Bad news for statevectors… They are usually *not* sparse for states of interest.
- We can use a more advanced technique for state compression – Matrix Product States.
  - Uses SVD compression, but the caveat is that you lose access to highly entangled states.
  - Number of strategies to reduce the impact of that, but the bottom line is, if you can do exact simulation, do exact simulation.

Full image, $\chi = 807$. Compressed image, $\chi = 10$.

QuEST

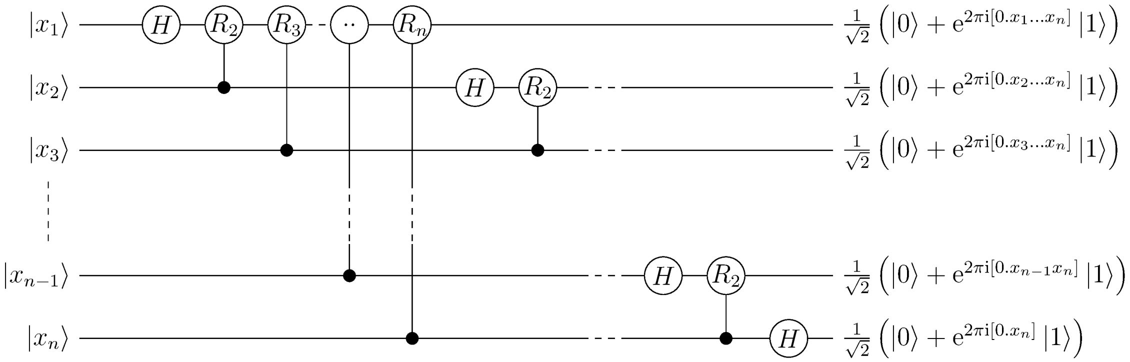
- QuEST (Quantum Exact Simulation Toolkit) is the software we recommend for emulating a quantum computer on ARCHER2.
  - https://github.com/QuEST-Kit/QuEST
  - Developed at University of Oxford by the QTechTheory group.
  - Parallelised using MPI+OpenMP. (Can be GPU accelerated too).
  - Written in C/C++.
  - Minimises the number of messages sent, at the cost of 1 additional qubit's worth of memory.
    - You can read more about their communication strategy in Jones, T., Brown, A., Bush, I. *et al.* QuEST and High Performance Simulation of Quantum Computers. *Sci Rep* **9,** 10736 (2019). https://doi.org/10.1038/s41598-019-47174-9.

QuEST

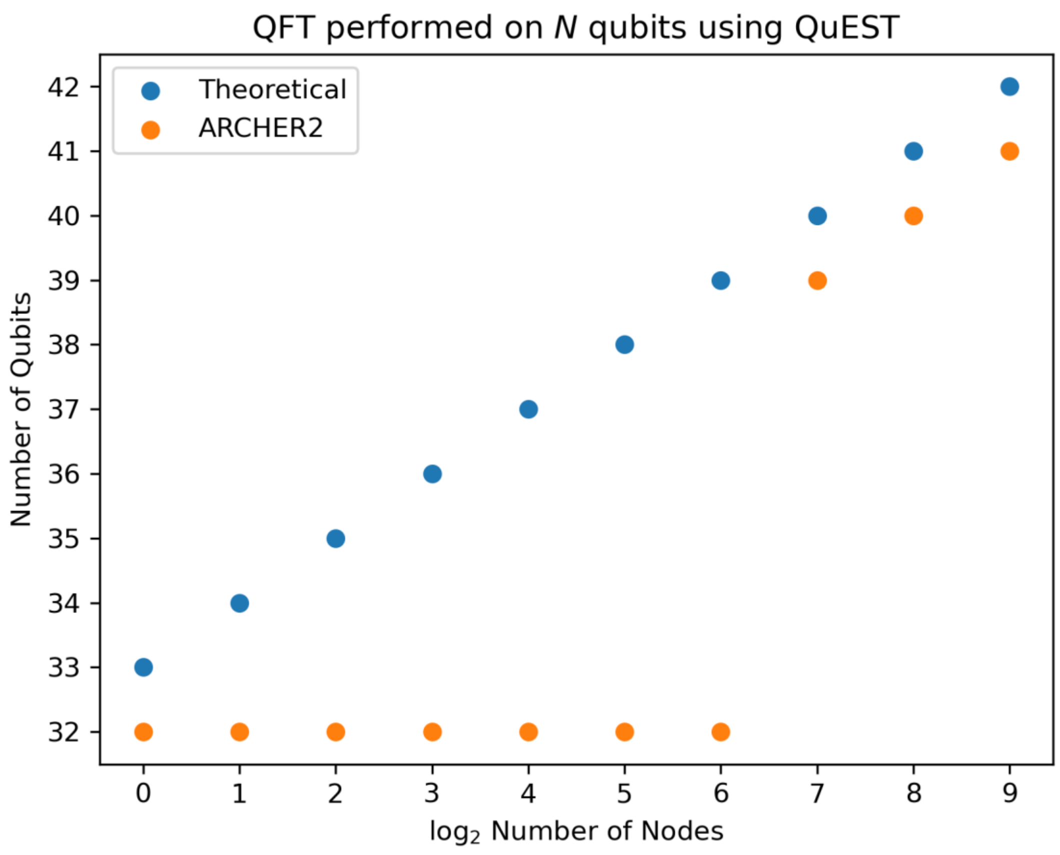
- Includes almost all canonical gates, and you can specify your own.
  - Beware very wide arbitrary gates when multithreading.
- Use one MPI process per node, 128 OpenMP threads.
  - `#SBATCH --ntasks-per-node=1`
  - `#SBATCH --cpus-per-task=128`
- There is a power-of-2 restriction on the number of MPI processes (and thus, nodes).
- Not centrally available on ARCHER2, but easy to build.
  - We could centrally install, if there's demand.

QFT

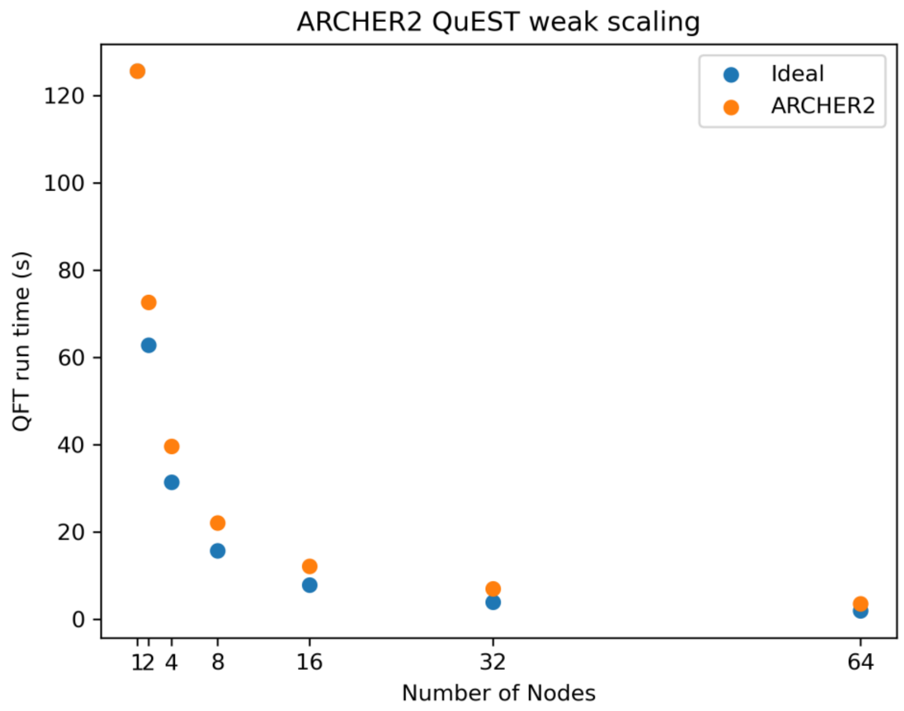
- Test code implemented a quantum Fourier transform on an $N-\mathrm{qubit}$, all $|0\rangle$ state.
  - Result is to place every qubit in the state $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$. Boring, but convenient!
- Total number of gates is $N(N-1) = \mathcal{O}(N^2)$.
- Assumes N perfect logical qubits – no noise.



ARCHER2 Results



- 41 qubit QFT successfully emulated on 512 nodes!
  - 35.18TB statevector.



ARCHER2 Results

- Number of nodes for N qubits:
  - $\lceil 2^{N-32} \rceil$
  - 32 is the number of qubits you can emulate on a single 256GB node (actually it's 33, because you don't need MPI!).
- Number of qubits on N nodes:
  - $\lfloor \log_2(N) + 32 \rfloor$
  - In theory once ARCHER2 is fully installed, should be able to emulate **44** qubits on **4096** nodes.
- On average a single qubit gate takes 3s when the node memory is fully loaded.

Qiskit-MPI

- Qiskit fans rejoice! An MPI parallelised version has been developed (but not yet released).
  - Once it's in release, we'll put it through its paces on ARCHER2.
- Details in the following paper:
  - J. Doi and H. Horii, "Cache Blocking Technique to Large Scale Quantum Computing Simulation on Supercomputers," *2020 IEEE International Conference on Quantum Computing and Engineering (QCE)*, Denver, CO, USA, 2020, pp. 212-222, doi: 10.1109/QCE49297.2020.00035.

What will we do with it?

- As a centre EPCC is interested in applying novel compute to real-world problems.
  - Working to understand where quantum computing can be applied, and what the benefits might be.
  - How can it be integrated with classical HPC?
- Working in collaboration with the Quantum Informatics group at Edinburgh.
- Part of a wider collaboration between University of Strathclyde, University of Edinburgh, and University of Glasgow, focussed on applied quantum computing. Name TBD!

Summary

- Quantum computing is hard.
- Emulating quantum computers is hard.
- Using QuEST, you can emulate a 41-qubit quantum computer on the ARCHER2 4-cabinet system!
- A scalable version of Qiskit is coming soon.
- EPCC is rapidly increasing our involvement in quantum computing through collaboration with partners across the UK (and beyond).
  - If you have an application that might benefit from quantum computing – get in touch!