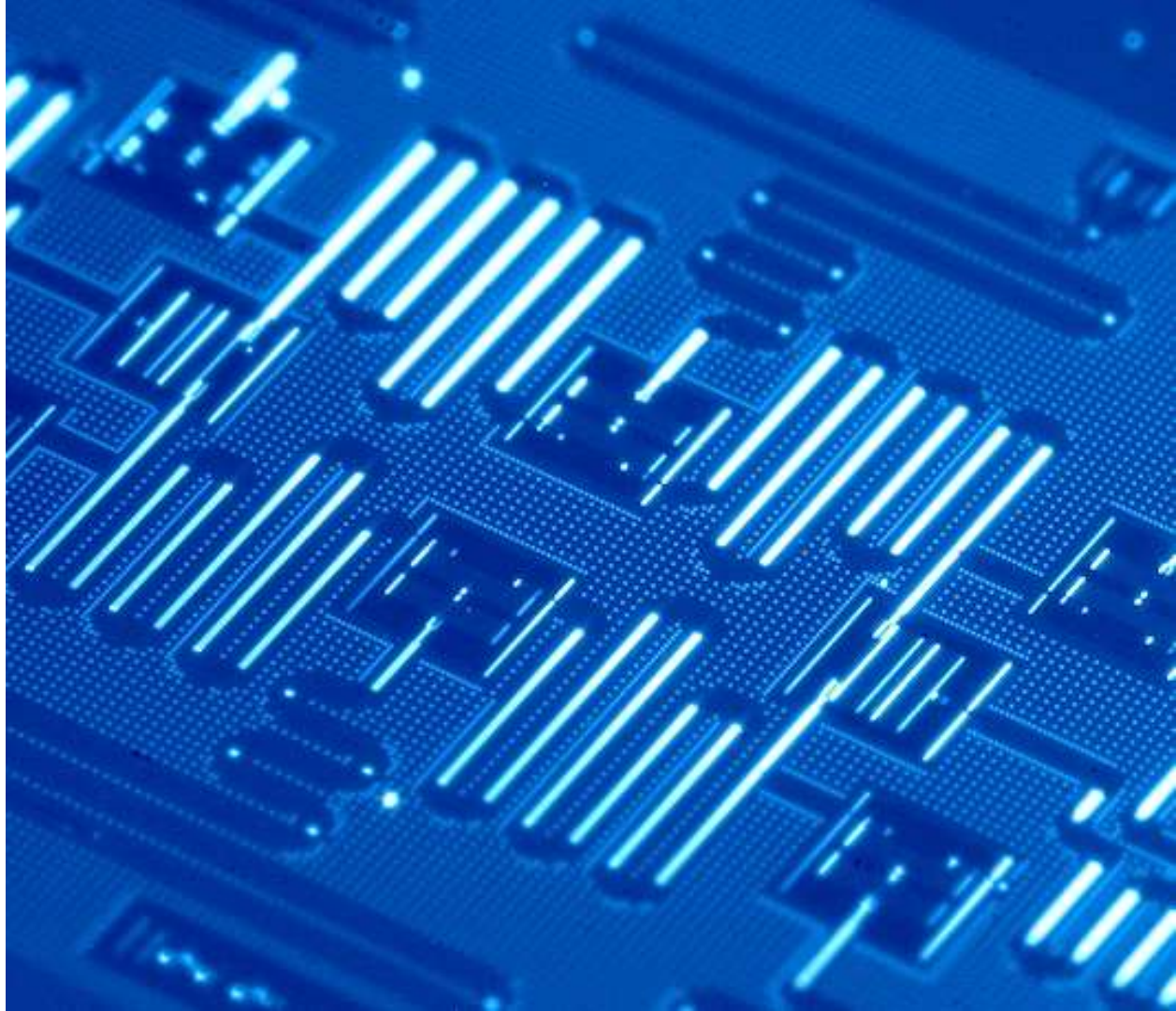
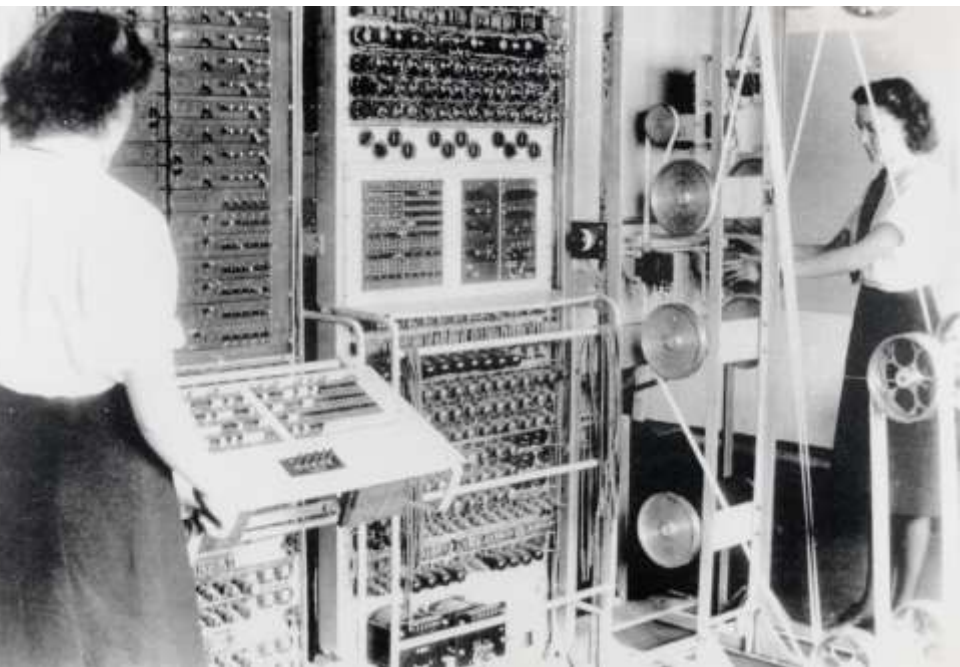




Quantum Computing





N bit input 100110...

**Quantum
Computer**

**N qubits
 2^N paths**

N bit output 010101...

Types of universal quantum computers

Universal fault-tolerant quantum computer

The holy grail of quantum information science. Allows one to run **useful** quantum algorithms which achieve **exponential speed ups** over their classical counterparts. However the over head of quantum error correction estimates **1M-5M qubits**

Approximate quantum computer

A quantum device which does not need fault tolerance, with the goal of demonstrating a useful application by interacting with a classical computing system, e.g. quantum chemistry, optimization. Estimate **1K-5K qubits**

Quantum Advantage

Quantum advantage is an idea that **before** any useful quantum computer is built it may be possible to demonstrate a **special purpose** quantum device or application whose output cannot be simulated as **fast** using existing classical computers. Estimate **50-100 qubits**

IBM Quantum Experience



Since launch

- >45,000 users
- >300,000 unique quantum circuits
- 15+ scientific publications
- 10+ professors committing to using IBM Quantum Experience for quantum course education

www.research.ibm.com/quantum

Defining the quantum assembly language (QASM)

Statement	Description	Example
<code>IBMQASM 2.0;</code> <code>qreg name[size];</code> <code>reg name[size];</code> <code>include "filename";</code> <code>gate name(params) qargs { body }</code> <code># comment text</code>	Denotes a file in IBM QASM 2.0 format ^a Declare a named register of qubits Declare a named register of bits Open and parse another source file Declare a unitary gate subroutine Comment a line of text	<code>IBMQASM 2.0;</code> <code>qreg q[5];</code> <code>reg c[5];</code> <code>include "mygates.incl";</code> (see text) <code># oops!</code>
<code>U(theta,phi,lambda) qubit qreg;</code> <code>CX qubit qreg,qubit qreg;</code> <code>measure qubit qreg -> bit reg;</code> <code>reset qubit qreg;</code> <code>gatename(params) qargs;</code> <code>if(reg==int) qop;</code>	Apply built-in single qubit gate(s) ^b Apply built-in CNOT gate(s) Make measurement(s) in Z basis Prepare qubit(s) in $ 0\rangle$ Apply a user-defined unitary gate Conditionally apply quantum operation	<code>U(pi/2,2*pi/3,0) q[0];</code> <code>CX q[0],q[1];</code> <code>measure q -> c;</code> <code>reset q[0];</code> <code>crz(pi/2) q[1],q[0];</code> <code>if(c==5) CX q[0],q[1];</code>
<code>barrier qargs;</code>	Prevent optimization across this source line	<code>barrier q[0],q[1];</code>

^a This must appear as the first line of the file.

^b The parameters `theta`, `phi`, and `lambda` are given by *parameter expressions*; see text.