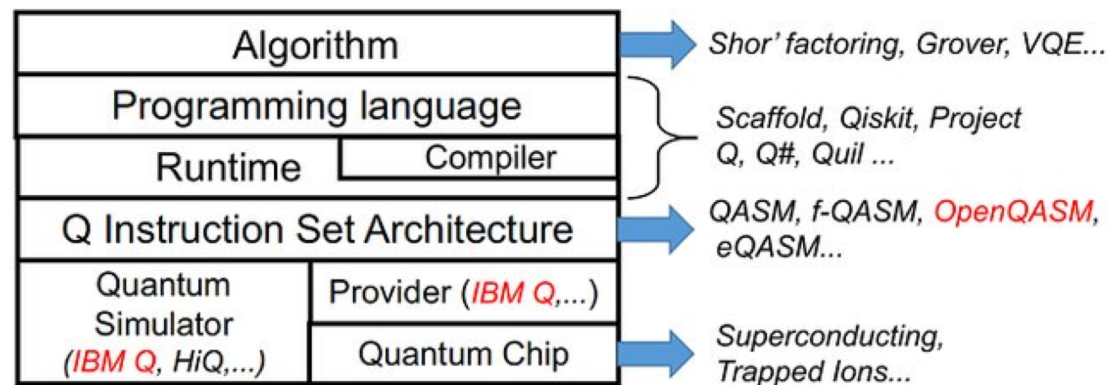


# Hands on with IBM-Q

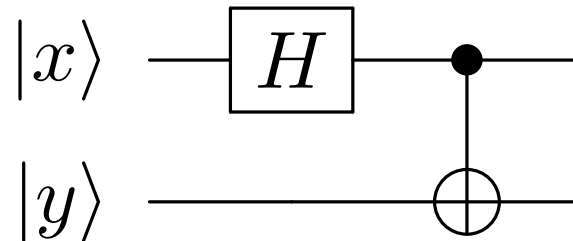
Survey of IBM-Q, circuit composer, backends.  
Install Qiskit locally and get started with  
tutorials.

# IBM-Q

- Go to IBM-Q: <https://quantum-computing.ibm.com>
- Sign-up, explore Circuit Composer (Home>Tools>Circuit Composer).
- How to make  $|\beta_{00}\rangle$ , compare simulator to real backend.
- OpenQASM: Quantum Assembler Language, the "Intermediate Representation" for quantum operations (the stack between high level Python programming and hardware).

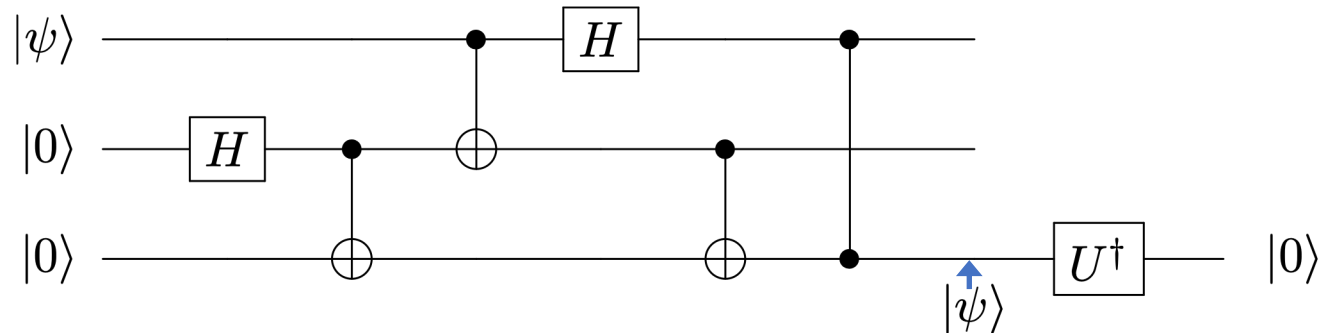


Now get your hands dirty and make  $|\beta_{xy}\rangle$



- Check that you got the right Bell state by doing appropriate measurements.
- How does the simulator compare to real devices? Is there a trend? Can you get better results by *judiciously* choosing the best qubits in the best devices?

# Teleportation algorithm with OpenQASM



- Teleport your favourite  $|\psi\rangle = U|0\rangle$ . After it's teleported, to check how well it worked.  
 Note: Conditionals (if then else) and qubit resets do not work on real backends yet. But you can replace one layer of conditionals by a suitable control-U operation.

- Note  in the circuit above.

# Install Qiskit locally in your machine

- Go to the installation guide: <https://github.com/Qiskit/qiskit-ix-tutorials/blob/master/INSTALL.md>  
Follow the instructions to install Anaconda. This will install Jupyter, Python 3.x and other packages.
- Go back to the installation guide and install Qiskit using pip.
- Follow the instructions to configure IBM-Q (you need to get your API\_token from IBM-Q Dashboard > My Account > copy token).
- Go to <https://github.com/Qiskit>. Click on qiskit-ix-tutorials > basics.