Lecture on Quantum Abstract Machines & Quil

- Defn A Quantum Abstract Machine (QAM) is a 6-tuple $(|\psi\rangle, C, G, G', P, \kappa)$:
  - $|\psi\rangle$ is the state of the QAM's quantum memory of N_q qubits init to $|0\ldots00\rangle$
  - C is the state of the QAM's classical memory of N_c bits init to $0\ldots00$
  - G is a set of quantum gates
  - G' is a set of parametrized quantum gates
  - P is a list of Quil instructions in sequence
  - an integer $0 \le \kappa \le |P|$ indicating what instruction to execute or if $\kappa = |P|$ to halt.

In diagram form | Qubits Mem | | Classical Mem |

- Quantum Gates operate by unitary matrix multiplication on the indexed qubits. <gate> <idx....>...< >

Defn A quantum gate or quantum operation is a unitary matrix of size $2^n \times 2^n$.

e.g $H := \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$ so | H 0 is a Hadamard on zeroth qubit.

$|00\rangle \xmapsto{H\ 0} \frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|01\rangle$

$|00\rangle \xmapsto{H\ 1} \frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|10\rangle$

e.g. $CNOT := \begin{pmatrix} 1&0&0&0 \\ 0&1&0&0 \\ 0&0&0&1 \\ 0&0&1&0 \end{pmatrix}$ | CNOT 1 2

CNOT 4 7

Each QAM has a set G of gates that can be operated on it. These are sometimes called the "natural gates" or "natural gate set" (when combined w/ the parametrized gates (G'))

Common gates From paulis $\{X, Y, Z\}$ Others T gate $:= \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}$

I identity

- Parametrized gates Defn A parametric gate is a parametrized unitary matrix $U(\vec{\theta})$ where $\vec{\theta}$ is a vector of complex numbers.

Examples Single-qubit rotations

$$RX(\theta) = e^{-i\theta X/2} = \begin{bmatrix} \cos\theta/2 & -i\sin\theta/2 \\ -i\sin\theta/2 & \cos\theta/2 \end{bmatrix} = \cos\tfrac{\theta}{2} I - i\sin\tfrac{\theta}{2} X$$

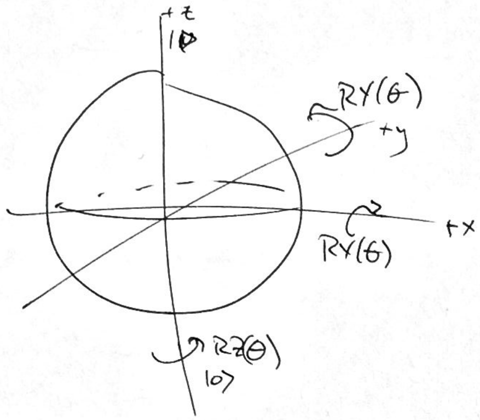
similarly for Y Z

$$RY(\theta) = e^{-i\theta Y/2} = \begin{bmatrix} \cos\theta/2 & -\sin\theta/2 \\ \sin\theta/2 & \cos\theta/2 \end{bmatrix}$$

Ex. 4.2 verify this

$$RZ(\theta) = \begin{bmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{bmatrix}$$

These are rotations on a Bloch sphere around the X, Y, and Z axes respectively



Thm Any single qubit operation can be constructed from RX, RY, RZ rotations.

Examples Two-qubit parametrized

$$PSWAP(\theta) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & e^{i\theta} & 0 \\ 0 & e^{i\theta} & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$ when $\theta = \pi$ this is a SWAP

$$CZ(\theta) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & e^{i\theta} \end{pmatrix}$$

PSWAP(1.7) 0 1

CZ(3.4) 1 7

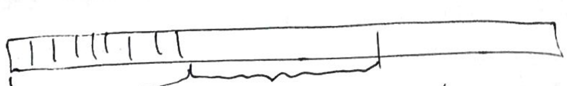
Universality CNOT OR PSWAP OR CZ + single qubit operations is universal. See Mike & Ike Section 4.5 for CNOT proof.

- A QAM's Natural Gate set is $G \cup G'$

- Measurements and Classical Memory
  - MEASURE $\langle q \in \mathbb{N}_q \rangle$ [$\langle c \in \mathbb{N}_c \rangle$] e.g. MEASURE 0 [1]
    measures the q^{th} qubit and stores the result in classical memory index[1]
  - DECLARE is used to specify these regions of classical memory. For example
    DECLARE ro BIT[16]

    Memory types are REAL, BIT, OCTET, INTEGER
    (BIT: size = 1, OCTET: size = 8)

classical memory



$\underbrace{\text{ro BIT[8]}}$ $\underbrace{\text{theta REAL[1]}}$ etc.

This means a MEASURE 0 ro[1] deposits into that region's second bit.

e.g. H 0
MEASURE 0 ro[0]

50% → |0⟩ & |0|0|0|0| . . .
50% → |1⟩ & |1|0|0| . . .

e.g. H 0
CNOT 0 2
MEASURE 0 ro[0]
MEASURE 2 ro[2]

$\frac{|00\rangle + |11\rangle}{\sqrt{2}}$

50% → |000⟩ & |0|0|0|0 . . .
50% → |1001⟩ & |1|0|0|1 - . . .

*careful indexing ← indexing →
in Quil both are ←

- Classically parameterized control (not fully implemented)

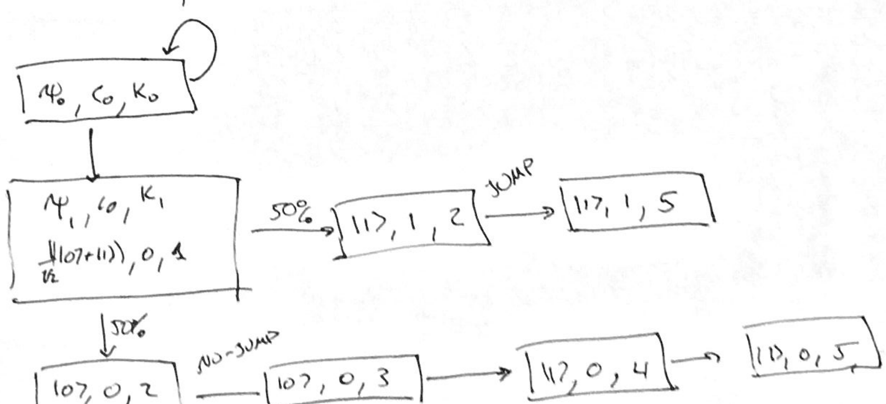
DECLARE theta REAL[1]
RX(theta) 0

This is critical for hybrid variational quantum programming as we will see later in the course.

- Classical control & Program counters

K describes program control. For example, we illustrate a QAM as a state machine

K	
0	DECLARE ro BIT[2]
1	H 0
2	MEASURE 0 ro[0]
3	JUMP-WHEN @END ro[0]
4	X 0
5	LABEL @END



- Program control commands:

JUMP-WHEN <str> c_i
JUMP-UNLESS <str> c_i
LABEL <str>
JUMP <str>

▷ Other useful Quil commands

RESET resets the quantum state to |00...⟩

NOP (distinguished from I)

```
DEFGATE HADAMARD:
    1/sqrt(2), 1/sqrt(2)
    1/sqrt(2), -1/sqrt(2)
```

```
DEFGATE RX(%theta):
    cos(%theta/2), -i*sin(%theta/2)
    -i*sin(%theta/2), cos(%theta/2)
```

```
DEFCIRCUIT BELL Qm Qn:
    H Qm
    CNOT Qm Qn
BELL 0 1
```

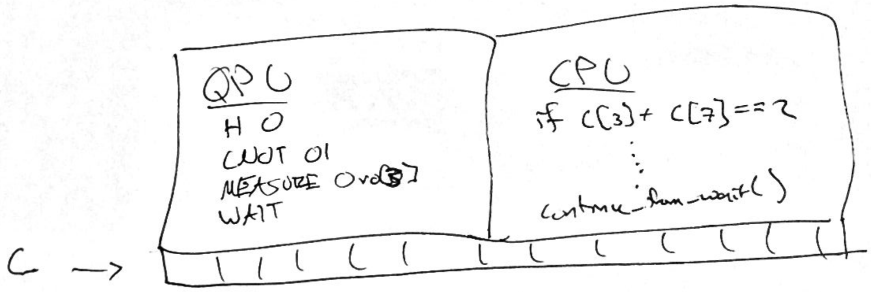
```
DEFCIRCUIT EULER(%alpha, %beta, %gamma) q:
    RX(%alpha) q
    RY(%beta) q
    RZ(%gamma) q
```

INCLUDE "stdgates.quil"

PRAGMA <id> <string> ⟵ used in noise modelling as we will see later

▷ synchronization (not yet implemented)

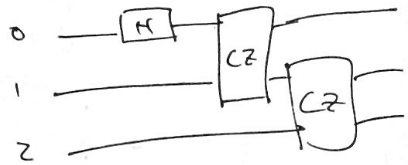
WAIT



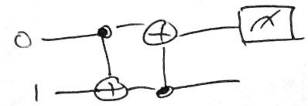
→ How to run Notebook example

Def A quantum circuit is a diagram representation of instructions (gates) on quantum memory e.g.

H 0
CZ 0 1 $\implies$
CZ 1 2



or CNOT 0 1
CNOT 1 0 $\implies$
MEASURE 1 [1]



These should look suspiciously like neural nets! They are ~~[illegible]~~ a family of them.

Historically circuits came before instruction sets.

They deal poorly w/ classical control and classical memory.

- also only good for small or highly structured circuits.

[Neural nets — tensor networks — monoidal categories] a whole different class!