Lecture 1.1: Overview of QC

Schedule:

13^{15} - 13^{30} : Intro to course and naming round
13^{30} - 13^{45} : Follow up on background material quiz
13^{45} - 14^{00} : Crawl before we walk: Classical computers & classical bits (1.2)
(break)
14^{15} - 14^{30} : Operations on cbits (1.3)
14^{30} - 15^{00} : Qbits and operations on them (1.5 + 1.6)

Follow up on quiz

- 1: What is a qubit?
  (Discussion + answer on blackboard)
- 2: What is a quantum computer?
  (Discussion + answer on blackboard)
- 3: What is a quantum simulator?
  (Discussion + answer on blackboard)
- 4: Main obstacles to quant. info. processing?
  (Discussion + answer on blackboard)
- 5: Real world applications?
  (Discussion + answer on blackboard)

A quantum computer is a special type of computer which uses quantum physics as part of calculation

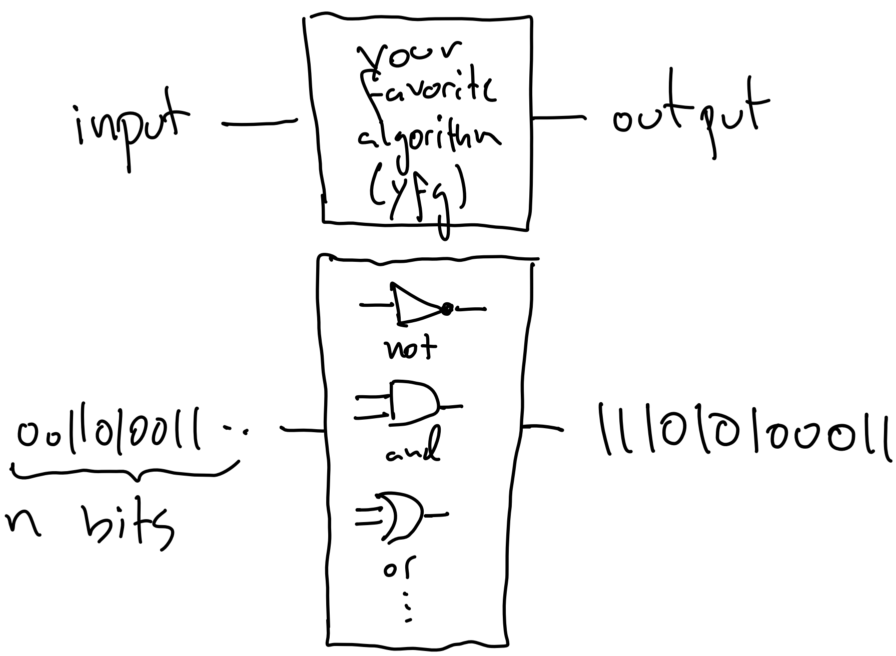
* Example of analog unprogrammable QC *

If the laws of quantum physics are exploited in just the right way, certain computations* can be solved faster on QC versus classical computer.

*) „the right way" and „certain computations" = topic of this course!

- We will first build model of classical computer, then extend to quantum computing.

CLASSICAL COMPUTING & CBITS



- Bits: A discrete number: $b \in \{0, 1\}$
- Mathematically : $b \in \mathbb{Z}_2$

- Introduce notation:

$b_1 = 1 \mapsto |b\rangle_1 = |1\rangle$. „ket" Read as: „State of b_1 is 1"

- Independent of implementation

$|b\rangle_1 = |1\rangle_1$:

charged capacitor
current flowing
Light bulb on.

- $|\ \rangle$ is called a „ket" (thanks, Dirac)

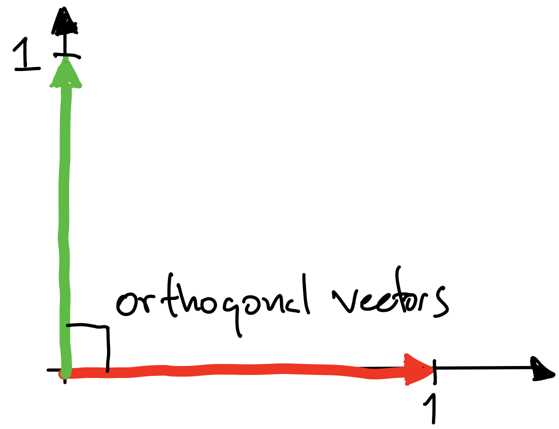
Q: How many states possible for a 2bit system?
How many states possible for a n-bit system?

A: 4 and 2^n

- May seem like overkill, but lets introduce a vector notation:

$$|0\rangle = \begin{pmatrix}1\\0\end{pmatrix}$$

$$|1\rangle = \begin{pmatrix}0\\1\end{pmatrix}$$



- How to interpret multi-bit state in this vector language?
- Need the tensor product! (dun dun duuuunnn!)

Multi c-bit states

- State of 2 cbits is: $|2\,\text{cbit}\rangle = |b_1\rangle \otimes |b_2\rangle$

$\otimes$: $|b_1\rangle = \begin{pmatrix}b_{1,1}\\b_{1,2}\end{pmatrix}$, $|b_2\rangle = \begin{pmatrix}b_{2,1}\\b_{2,2}\end{pmatrix}$: $|b_1\rangle \otimes |b_2\rangle = \begin{pmatrix}b_{11}\\b_{12}\end{pmatrix} \otimes \begin{pmatrix}b_{21}\\b_{22}\end{pmatrix} = \begin{pmatrix}b_{11}\cdot\begin{pmatrix}b_{21}\\b_{22}\end{pmatrix}\\ b_{12}\cdot\begin{pmatrix}b_{21}\\b_{22}\end{pmatrix}\end{pmatrix} = \begin{pmatrix}b_{11}b_{21}\\b_{11}b_{22}\\b_{12}b_{21}\\b_{12}b_{22}\end{pmatrix}$

Some notation:

NOTE: DROPPED SUBSCRIPT, ⇒ ORDER NOW MATTERS

$$|\alpha\rangle \otimes |\beta\rangle = |\alpha\rangle_1 |\beta\rangle_2 = |\alpha\beta\rangle$$

$|11\rangle = |1\rangle \otimes |1\rangle = |11\rangle$ BUT! $|a,b\rangle = |a\rangle \otimes |b\rangle$

This comma is important when ambiguous if 1 state labeled by ab or (state a) ⊗ (state b)

Example

Consider the 2 c-bit state: $|01\rangle$

"First bit in 0, second in 1"

$$= |0\rangle \otimes |1\rangle = \begin{pmatrix}1\\0\end{pmatrix} \otimes \begin{pmatrix}0\\1\end{pmatrix} = \begin{pmatrix}1\cdot\begin{pmatrix}0\\1\end{pmatrix}\\ 0\cdot\begin{pmatrix}0\\1\end{pmatrix}\end{pmatrix} = \begin{pmatrix}0\\1\\0\\0\end{pmatrix}$$

Q: What is vector representation of $|\underbrace{11}_{4_{10}}\rangle$

A: $\begin{pmatrix}0\\0\\0\\1\end{pmatrix}$

Q: Why does ordering now matter?

A: Because the tensor product $|a\rangle \otimes |b\rangle \neq |b\rangle \otimes |a\rangle$ in general

- Rule of thumb: $|01101\rangle$: $01101_{\text{base }2} = 13_{\text{base }10}$
  - Vector length: $2^5 = 32$
  - In 13'th spot, has a 1 $= \begin{pmatrix}0\\ \vdots\\ 0\\ 1\\ 0\\ \vdots\\ 0\end{pmatrix}$ $\begin{matrix}\}12\\ \\ \}19\end{matrix}$
- Generalization of tensor states:

$$\begin{pmatrix}x_0\\x_1\end{pmatrix} \otimes \begin{pmatrix}y_0\\y_1\end{pmatrix} \otimes \begin{pmatrix}z_0\\z_1\end{pmatrix} = \begin{bmatrix}x_0y_0z_0\\x_0y_0z_1\\x_0y_1z_0\\x_0y_1z_1\\ \vdots\\ x_1y_1z_1\end{bmatrix}$$

Q: What is vector of $|\underbrace{110}_{7_{10}}\rangle$?

A: $\begin{pmatrix}0\\0\\0\\0\\0\\0\\1\\0\end{pmatrix}$

OPERATIONS ON CBITS

- Will only study reversible operations on cbits.

The NOT operation:

$$X : |b\rangle \longrightarrow |\tilde{b}\rangle \quad (|\neg b\rangle)$$

$$\tilde{0} = 1, \quad \tilde{1} = 0$$

$$\neg 0 = 1, \quad \neg 1 = 0$$

- Obvious:

$$XX : |b\rangle \rightarrow |b\rangle \Rightarrow XX = 1$$

- From the column vector definition, X must have the form

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

Q: Any physicist/nano recognize matrix?
A: Pauli σ_x

Q: What is the matrix representation of identity operator?
A: $1 = \mathbb{1} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \mathbb{1}_2$

- Multi-cbit operators are also defined through tensor product:

2 cbit: $|\alpha, \beta\rangle$: A acting on $|\alpha\rangle$, B acting on $|\beta\rangle$: $\Lambda = A \otimes B$

Q: Consider $\Lambda : |b_1 b_2\rangle \mapsto |\tilde{b}_1 b_2\rangle$. What is matrix form of Λ?

A: $\Lambda = \overline{\overline{X}} \otimes \mathbb{1} = \begin{bmatrix} 0 & 1 \cdot \mathbb{1}_2 \\ 1 \cdot \mathbb{1}_2 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$

- Rules about manipulating many-cbit operators:

1: $(A \otimes B)|b_1 b_2\rangle = (A \otimes B)(|b_1\rangle \otimes |b_2\rangle) = A|b_1\rangle \otimes B|b_2\rangle$

2: $(A \otimes B)(C \otimes D) = (AC) \otimes (BD)$

- What if A acts on bit 5 and B on bit 8?

3: $\mathbb{1} \otimes \mathbb{1} \otimes \mathbb{1} \otimes \mathbb{1} \otimes A \otimes \mathbb{1} \otimes \mathbb{1} \otimes B = A_5 B_8 = B_8 A_5$

• Now move to more interesting 2-qbit operation:

The controlled NOT operation

• Two qubit operation which applies X to 2nd bit, if first is in $|1\rangle$:

$$CNOT|b_1\rangle|b_2\rangle = |b_1\rangle|b_1 \oplus b_2\rangle$$

where "$\oplus$" is addition modulo 2:

$$b \oplus 0 = b \qquad b \oplus 1 = \bar{b} = 1-b$$

sometimes called "exclusive OR" (XOR)

Q: Fill in truth table for CNOT

control $|b_1\rangle$	target $|b_2\rangle$	$CNOT|b_1 b_2\rangle$
$|0\rangle$	$|0\rangle$	$|00\rangle$
$|0\rangle$	$|1\rangle$	$|01\rangle$
$|1\rangle$	$|0\rangle$	$|11\rangle$
$|1\rangle$	$|1\rangle$	$|10\rangle$

exchanges these 2.

• The matrix form of CNOT:

$$CNOT = \begin{array}{c} \\ \langle 00| \\ \langle 01| \\ \langle 10| \\ \langle 11| \end{array} \begin{array}{c} \begin{array}{cccc} |00\rangle & |01\rangle & |10\rangle & |11\rangle \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \end{array}$$

identity in $\{|00\rangle, |01\rangle\}$ subspace

Flip in $\{|10\rangle, |11\rangle\}$ subspace

• We will use this matrix x1.000.000 times!

$$\left[CNOT_{21} = \begin{array}{c} \\ \langle 00| \\ \langle 01| \\ \langle 10| \\ \langle 11| \end{array} \begin{array}{c} \begin{array}{cccc} |00\rangle & |01\rangle & |10\rangle & |11\rangle \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \end{array} \right]$$

Finally: Qubits

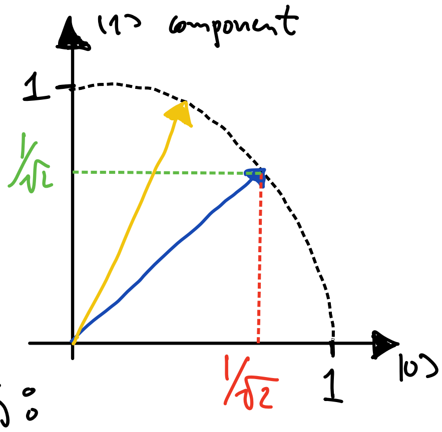
- Expressing cbits as vectors and reversible operations on them as matrices was a weird game. The vector space was rather boring
- Extend now to quantum states:

$$\boxed{|\psi\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle \quad \text{where} \quad \alpha_0, \alpha_1 \in \mathbb{C} \quad \text{and} \quad |\alpha_0|^2 + |\alpha_1|^2 = 1}$$

- We say „$|\psi\rangle$ is in superposition of $|0\rangle$ and $|1\rangle$ with amplitudes α_0 and α_1"
- In the case $\alpha_0 = 0$ (or $\alpha_1 = 0$) is in a classical state.

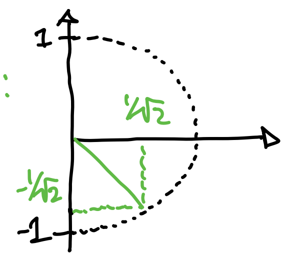
Example:

$$|\psi\rangle = \boxed{\frac{1}{\sqrt{2}}|0\rangle} + \boxed{\frac{1}{\sqrt{2}}|1\rangle}$$



Q: Draw state $|\lambda\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$ in $\{|0\rangle, |1\rangle\}$ space:

A:



- A two qubit state is:

$$\boxed{|\Psi\rangle = \alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle = \begin{pmatrix} \alpha_{00} \\ \alpha_{01} \\ \alpha_{10} \\ \alpha_{11} \end{pmatrix} \quad (+) \quad \text{and} \quad |\alpha_{00}|^2 + |\alpha_{01}|^2 + |\alpha_{10}|^2 + |\alpha_{11}|^2 = 1 \ \text{(normalization)}}$$

- We say the $|0\rangle, |1\rangle, |00\rangle, |01\rangle, \ldots$ basis is the computational basis.
- Combining 2 single qubit states to a single 2 qubit state:

$$|\alpha\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle$$
$$|\beta\rangle = \beta_0|0\rangle + \beta_1|1\rangle$$

$$\Rightarrow |\Psi\rangle = |\psi\rangle \otimes |\phi\rangle = (\alpha_0|0\rangle + \alpha_1|1\rangle) \otimes (\beta_0|0\rangle + \beta_1|1\rangle)$$
$$= \alpha_0\beta_0|00\rangle + \alpha_1\beta_0|10\rangle + \alpha_0\beta_1|01\rangle + \alpha_1\beta_1|11\rangle$$
$$= \begin{pmatrix} \alpha_0\beta_0 \\ \alpha_0\beta_1 \\ \alpha_1\beta_0 \\ \alpha_1\beta_1 \end{pmatrix} \quad (*)$$

- Important note:
  if and only if $\Psi_{00}\Psi_{11} = \Psi_{01}\Psi_{10}$ then (†) is of form (*)
- General state need not be of form (*)
- (*) is a product state
- Cbit systems are <u>always</u> product states
- Qubit systems are not in general product states.

<u>Famous example:</u> Bell states

$$|\Phi_\pm\rangle = \frac{1}{\sqrt{2}}(|00\rangle \pm |11\rangle), \quad |\Psi_\pm\rangle = \frac{1}{\sqrt{2}}(|01\rangle \pm |10\rangle)$$

<u>Q</u>: Can you write these as product states? Try!

<u>A</u>: You can't!

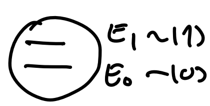
- The 2 qubits in the bell state is <u>entangled</u>
- While a general n-cbit state has 2^n available states, any given cbit-state will only require n bits to write down
- At a heuristic (but deep) sense the power of quantum computing can be traced back to non-product states.
- Entanglement (in a specific form) is what gives quantum speedups.*, **

*) only <u>strictly</u> true for pure state QC.

**) Certain restricted models of QC have „marginal" entanglement, but has other non-classical correlations (e.g. the „one clean qubit" model)

<u>Qudits</u>

- So far everything has been two-level systems



- But nature certainly has systems which are 3, 4... level systems:

$$\text{qudits}: \quad |\Psi\rangle = \sum_{i=1}^{d} \alpha_i |i\rangle :$$

d-level system:

- Qubits is the most common. Will always be clear from context if qudit

OPERATIONS ON QUBITS

- For Cbits there was just one single-bit operation: X (boring vector space)
- For qubits there is a lot more room:

> **Operations on qubits:**
>
> Any linear operator that takes unit vectors to unit vectors (maintain normalization)

- Such transformations are called unitary:

$$UU^{\dagger} = \mathbb{1} \qquad (U \text{ is } 2^n \times 2^n)$$

- Recall: $M = \begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{11} \end{bmatrix}$: $M^{\dagger} = (M^*)^T = \begin{bmatrix} m_{11}^* & m_{21}^* \\ m_{12}^* & m_{22}^* \end{bmatrix}$

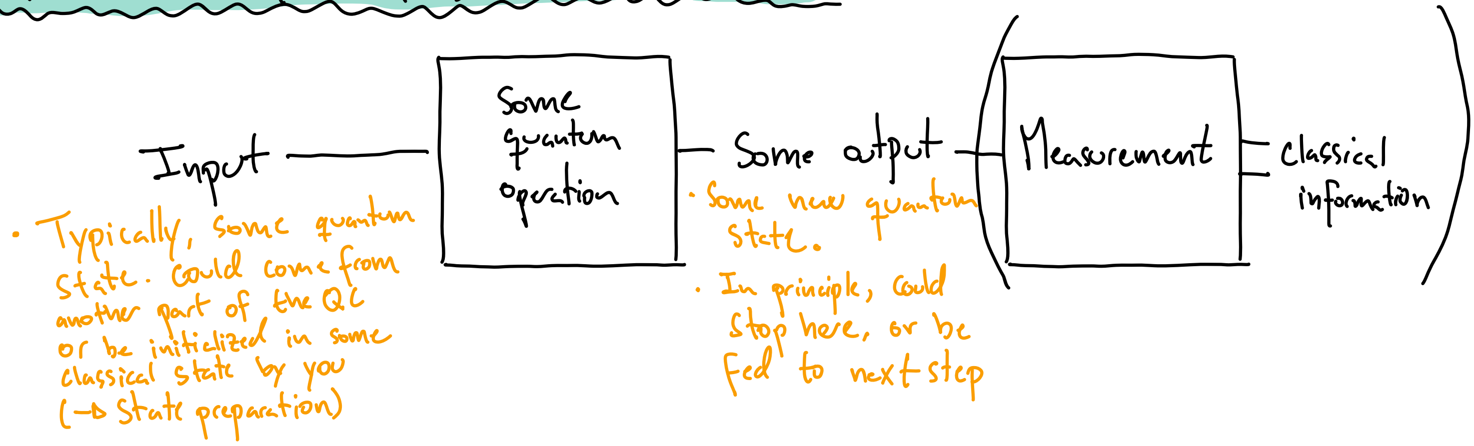
- Any unitary operator has a unitary inverse: This was why we restricted ourselves to reversible classical computing before.

Q: Is X and CNOT unitary?

A: $XX^{\dagger} = XX = \mathbb{1}$, $\text{CNOT}\,\text{CNOT}^{\dagger} = \text{CNOT}\,\text{CNOT} = \mathbb{1}$: yes!

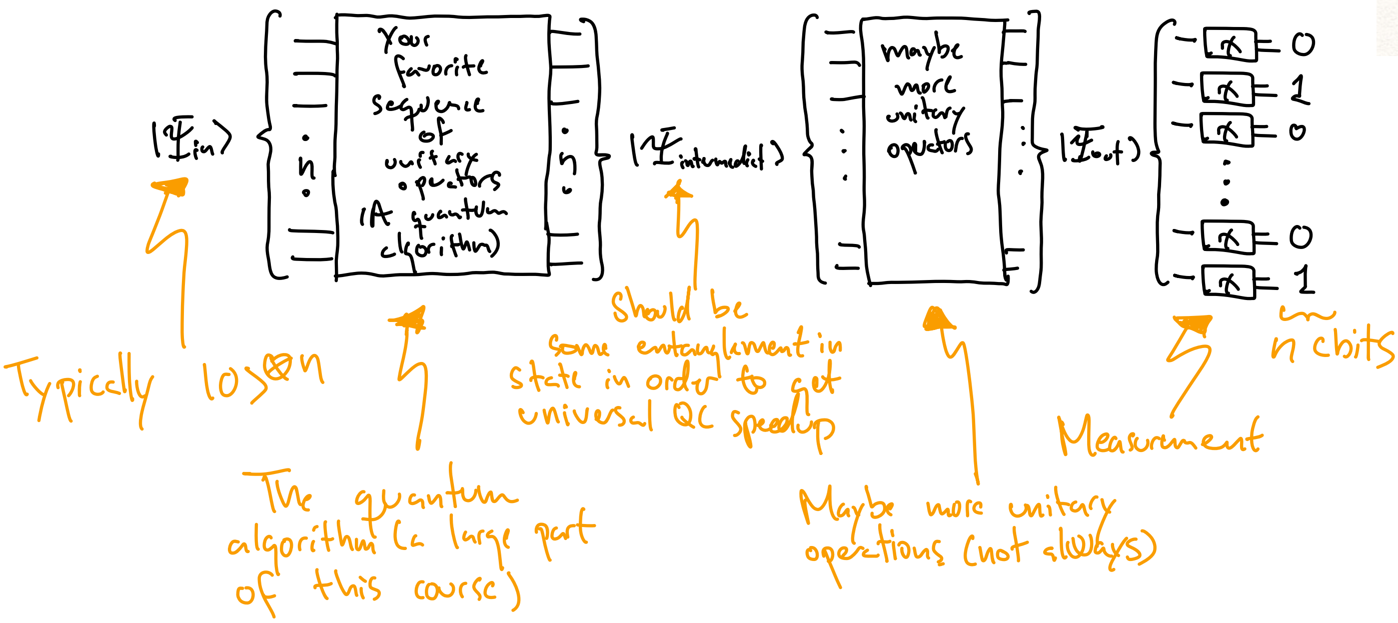
- We typically call X a 1-qubit gate and CNOT a 2 qubit gate
- MUCH of the work in experimental quantum computing is devoted to implementing 1- and 2-qubit gates with low error.

A GENERAL MODEL OF A QC:



A VERY COMMON SCHEME (specific) OF QC

L1.1-



· Practical examples

$|\Psi_{in}\rangle$ =
- · Charged ion (Ion trap QC)
- · Spin of an electron (spin qubits)
- · Charge of a superconducting island (superconducting QC) — MK research field — [Google, IBM etc]
- · Polarized light (Photonic QC)
- · Exotic quasiparticles (Topological QC) [Microsoft]
- ...