POLITECNICO DI MILANO

Dipartimento di
Elettronica, Informazione
e Bioingegneria

Quantum Computing: From Circuit To Architecture

Nicholas Mainardi
Email: nicholas.mainardi@polimi.it
home.deib.polimi.it/nmainardi

8th Semptember 2017

Credits & Outline

This talk is mainly based on portions of the course held by Carmen G. Almudever (TU Delft) at Acaces summer school (Fiuggi, 9-14th July 2017), enriched with some material by myself

Outline

1. Qu-bit definition
2. Quantum gates
3. Multi-states
4. Example: Teleportation
5. Quantum Algorithms
6. Quantum Processor
7. Compilation
8. Quantum Computers Architecture

Qu-Bit: Definition

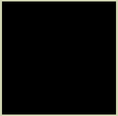
Classical Bit

Bit is either 0 or 1:

Qu-Bit: Definition

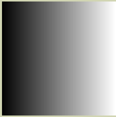
Classical Bit

Bit is either 0 or 1:



Quantum Bit

The bit is in a superposition state: it is both 0 and 1



A qu-bit ψ is defined as: $|\psi\rangle = \alpha_0 |0\rangle + \alpha_1 |1\rangle \qquad \alpha_0, \alpha_1 \in \mathbb{C}$

Since $|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ and $|1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \Rightarrow |\psi\rangle = \begin{bmatrix} \alpha_0 \\ \alpha_1 \end{bmatrix}$

Qu-Bit: Measurement

What do α_0 and α_1 actually mean?

Measurement

Consider a qu-bit $|\psi\rangle = \begin{bmatrix} \alpha_0 & \alpha_1 \end{bmatrix}$. Define the measurement as a function $M(|\psi\rangle)$ with range $\{0, 1\}$, such that:

- $Pr(M(|\psi\rangle) = 0) = |\alpha_0|^2$
- $Pr(M(|\psi\rangle) = 1) = |\alpha_1|^2$

Therefore, it must be $|\alpha_0|^2 + |\alpha_1|^2 = 1$

Qu-Bit: Measurement

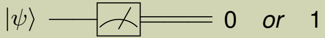
What do α_0 and α_1 actually mean?

Measurement

Consider a qu-bit $|\psi\rangle = \begin{bmatrix}\alpha_0 & \alpha_1\end{bmatrix}$. Define the measurement as a function $M(|\psi\rangle)$ with range $\{0, 1\}$, such that:

- $Pr(M(|\psi\rangle) = 0) = |\alpha_0|^2$
- $Pr(M(|\psi\rangle) = 1) = |\alpha_1|^2$

Therefore, it must be $|\alpha_0|^2 + |\alpha_1|^2 = 1$



$$\Downarrow$$

The superposition state is destroyed after measurement!

Qu-Bit: Measurement

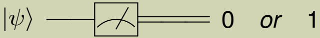
What do α_0 and α_1 actually mean?

Measurement

Consider a qu-bit $|\psi\rangle = \begin{bmatrix}\alpha_0 & \alpha_1\end{bmatrix}$. Define the measurement as a function $M(|\psi\rangle)$ with range $\{0, 1\}$, such that:

- $Pr(M(|\psi\rangle) = 0) = |\alpha_0|^2$
- $Pr(M(|\psi\rangle) = 1) = |\alpha_1|^2$

Therefore, it must be $|\alpha_0|^2 + |\alpha_1|^2 = 1$



$\Downarrow$

The superposition state is destroyed after measurement!

$\Downarrow$

We cannot directly measure the superposition, only probabilistic estimation

Qu-Bit: A Real World Example

Qu-Bit: A Real World Example

Crystal of Tourmaline: Classical World

- Interaction with plane-polarized light:
  1. Light polarized perpendicularly w.r.t. the crystal axis $\Rightarrow$ The light goes through the crystal
  2. Light polarized parallel w.r.t. the crystal axis $\Rightarrow$ The light is filtered by the crystal
  3. Light polarized with angle α w.r.t. the crystal axis $\Rightarrow$ A fraction $\sin^2 \alpha$ goes through

Qu-Bit: A Real World Example

Crystal of Tourmaline: Classical World

- Interaction with plane-polarized light:
  1. Light polarized perpendicularly w.r.t. the crystal axis $\Rightarrow$ The light goes through the crystal
  2. Light polarized parallel w.r.t. the crystal axis $\Rightarrow$ The light is filtered by the crystal
  3. Light polarized with angle α w.r.t. the crystal axis $\Rightarrow$ A fraction $\sin^2 \alpha$ goes through

Crystal of Tourmaline: Quantum World

- Interaction with a single plane-polarized photon:
  1. Photon polarized perpendicularly w.r.t. the crystal axis $\Rightarrow$ The photon is detected after the crystal
  2. Photon polarized parallel w.r.t. the crystal axis $\Rightarrow$ The photon is not detected after the crystal
  3. Photon polarized with angle α w.r.t. the crystal axis $\Rightarrow$ A photon perpendicularly polarized is detected $\sin^2 \alpha$ times, no photon detected otherwise

Qu-Bit: A Real World Example

From Physic World to Qu-Bit

Qu-bit $\Leftrightarrow$ the polarization direction of a single photon

$|0\rangle$ photon polarized perpendicular w.r.t. the crystal axis

$|1\rangle$ photon polarized parallel w.r.t. the crystal axis

Superposition state? a photon polarized with angle α w.r.t. the crystal axis: $|\psi\rangle = \sin\alpha\,|0\rangle + \cos\alpha\,|1\rangle$

Measurement

- The qu-bit is 0 with probability $\sin^2\alpha$
- The qu-bit is 1 with probability $\cos^2\alpha$
- The qu-bit is destroyed: no longer polarized with angle α

Quantum Gates

- Qu-bits are vectors in $\mathbb{C}$ $\Rightarrow$ Gates are matrices in $\mathbb{C}$
- Properties? Unitary Operations!
- Generic gate $—\boxed{U}—$: $UU^* = U^*U = I \Rightarrow |det(U)| = 1$

$$\Downarrow$$

Quantum gates are reversible!

Quantum Gates

- Qu-bits are vectors in $\mathbb{C} \Rightarrow$ Gates are matrices in $\mathbb{C}$
- Properties? Unitary Operations!
- Generic gate $-\boxed{U}-$: $UU^* = U^*U = I \Rightarrow |det(U)| = 1$

$$\Downarrow$$

Quantum gates are reversible!

Main Single Qu-Bit Gates

- Bit Flip Gate: $-\boxed{X}- \Rightarrow \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$
- Identity Gate: $-\boxed{I}- \Rightarrow \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$
- Phase Flip Gate: $-\boxed{Z}- \Rightarrow \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$
- Hadamar Gate: $-\boxed{H}- \Rightarrow \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$

Quantum Gates: Hadamar

Hadamar Gate Effect

$$|\psi_{out}\rangle = H\,|\psi_{in}\rangle\,, \qquad H = \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}$$

$$|\psi_{\mathbf{in}}\rangle = |\mathbf{0}\rangle$$

$$|\psi_{out}\rangle = H\,|0\rangle = \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}\begin{bmatrix}1\\ 0\end{bmatrix}$$
$$= \frac{1}{\sqrt{2}}\begin{bmatrix}1\\ 1\end{bmatrix} = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$

$$|\psi_{\mathbf{in}}\rangle = |\mathbf{1}\rangle$$

$$|\psi_{out}\rangle = H\,|1\rangle = \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}\begin{bmatrix}0\\ 1\end{bmatrix}$$
$$= \frac{1}{\sqrt{2}}\begin{bmatrix}1\\ -1\end{bmatrix} = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

$\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) = |+\rangle$ and $\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) = |-\rangle$ are 2 relevant states.
Why?

Quantum Gates: Hadamar

Hadamar Gate Effect

$$|\psi_{out}\rangle = H\,|\psi_{in}\rangle\,, \qquad H = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

$$|\psi_{\mathbf{in}}\rangle = |\mathbf{0}\rangle$$

$$|\psi_{out}\rangle = H\,|0\rangle = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}\begin{bmatrix} 1 \\ 0 \end{bmatrix} = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 \\ 1 \end{bmatrix} = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$

$$|\psi_{\mathbf{in}}\rangle = |\mathbf{1}\rangle$$

$$|\psi_{out}\rangle = H\,|1\rangle = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}\begin{bmatrix} 0 \\ 1 \end{bmatrix} = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 \\ -1 \end{bmatrix} = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

$\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) = |+\rangle$ and $\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) = |-\rangle$ are 2 relevant states. Why?

$$\Downarrow$$

For both of them, $|\alpha_0|^2 = |\alpha_1|^2 = \frac{1}{2}$

Quantum Circuits

From Gates to Circuits

$$|\psi_{in}\rangle \text{ --- } \boxed{U_1} \text{ --- } \boxed{U_2} \text{ --- } \boxed{U_3} \text{ --- } |\psi_{out}\rangle$$

$$|\psi_{out}\rangle = U_3 U_2 U_1 \, |\psi_{in}\rangle$$

Reversibility: The way back!

$$|\psi_{in}\rangle = U_1^* U_2^* U_3^* \, |\psi_{out}\rangle$$

Quantum Circuits

Examples

$$|0\rangle \;-\boxed{H}-\boxed{Z}-\boxed{H}- = \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}\begin{bmatrix}1 & 0\\ 0 & -1\end{bmatrix}\frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}\begin{bmatrix}1\\ 0\end{bmatrix}$$

$$= \frac{1}{\sqrt{2}}\begin{bmatrix}1 & -1\\ 1 & 1\end{bmatrix}\frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}\begin{bmatrix}1\\ 0\end{bmatrix}$$

$$= \frac{1}{2}\begin{bmatrix}0 & 2\\ 2 & 0\end{bmatrix}\begin{bmatrix}1\\ 0\end{bmatrix} = \begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix}\begin{bmatrix}1\\ 0\end{bmatrix} = X\,|0\rangle$$

$$= |1\rangle$$

$$|1\rangle \;-\boxed{H}-\boxed{X}-\boxed{H}- = \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}\begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix}\frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}\begin{bmatrix}0\\ 1\end{bmatrix}$$

$$= \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ -1 & 1\end{bmatrix}\frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}\begin{bmatrix}0\\ 1\end{bmatrix}$$

$$= \frac{1}{2}\begin{bmatrix}2 & 0\\ 0 & -2\end{bmatrix}\begin{bmatrix}0\\ 1\end{bmatrix} = \begin{bmatrix}1 & 0\\ 0 & -1\end{bmatrix}\begin{bmatrix}0\\ 1\end{bmatrix} = Z\,|1\rangle$$

$$= -\,|1\rangle$$

Multi Qu-Bit State

- Concise representation of superposition of multiple bits
- Real computational power of quantum computers!
- We need to introduce a new algebraic operation: the tensor product $\otimes$

Multi Qu-Bit State

- Concise representation of superposition of multiple bits
- Real computational power of quantum computers!
- We need to introduce a new algebraic operation: the tensor product $\otimes$

Tensor Product

Given $A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$ and $B \in \mathbb{C}^{n \times m}$, the tensor product is defined as:

$$T = A \otimes B = \begin{bmatrix} a_{11}B & a_{12}B \\ a_{21}B & a_{22}B \end{bmatrix}$$

Example: $A = \begin{bmatrix} 2 \\ 3 \end{bmatrix}$ $\qquad B = \begin{bmatrix} 1 & 4 \\ 5 & 2 \end{bmatrix}$

$$T = \begin{bmatrix} 2\begin{bmatrix} 1 & 4 \\ 5 & 2 \end{bmatrix} \\ 3\begin{bmatrix} 1 & 4 \\ 5 & 2 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} 2 & 8 \\ 10 & 4 \\ 3 & 12 \\ 15 & 6 \end{bmatrix}$$

Multi Qu-Bit State

- The superposition state may apply to multiple qu bits
- Instead of the 2 kets $|0\rangle$ and $|1\rangle$, there is a ket for each possible combination of bits
- The coefficients are related to the measurement probability of the corresponding combination

2 Qu-bits State

$$|\psi\rangle = \alpha_0 |00\rangle + \alpha_1 |01\rangle + \alpha_2 |10\rangle + \alpha_3 |11\rangle$$
$$|\alpha_0|^2 + |\alpha_1|^2 + |\alpha_2|^2 + |\alpha_3|^2 = 1$$

Vector representation? $\Rightarrow$ tensor product between single qu-bit kets

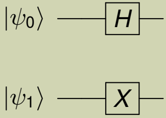
$$|00\rangle = |0\rangle \otimes |0\rangle = \begin{bmatrix}1 & 0\end{bmatrix} \otimes \begin{bmatrix}1 & 0\end{bmatrix} = \begin{bmatrix}1\begin{bmatrix}1 & 0\end{bmatrix} & 0\begin{bmatrix}1 & 0\end{bmatrix}\end{bmatrix} = \begin{bmatrix}1 & 0 & 0 & 0\end{bmatrix}$$

$$|10\rangle = |1\rangle \otimes |0\rangle = \begin{bmatrix}0 & 1\end{bmatrix} \otimes \begin{bmatrix}1 & 0\end{bmatrix} = \begin{bmatrix}0\begin{bmatrix}1 & 0\end{bmatrix} & 1\begin{bmatrix}1 & 0\end{bmatrix}\end{bmatrix} = \begin{bmatrix}0 & 0 & 1 & 0\end{bmatrix}$$

$$\begin{aligned}|\psi_2\rangle = |\psi_0\rangle \otimes |\psi_1\rangle &= \begin{bmatrix}\alpha_0 & \alpha_1\end{bmatrix} \otimes \begin{bmatrix}\beta_0 & \beta_1\end{bmatrix} = \begin{bmatrix}\alpha_0\begin{bmatrix}\beta_0 & \beta_1\end{bmatrix} & \alpha_1\begin{bmatrix}\beta_0 & \beta_1\end{bmatrix}\end{bmatrix} \\ &= \begin{bmatrix}\alpha_0\beta_0 & \alpha_0\beta_1 & \alpha_1\beta_0 & \alpha_1\beta_1\end{bmatrix}\end{aligned}$$

Multi Qu-Bit: Quantum Circuit

2 Qu-bits Circuit



$|\psi_{out}\rangle = ??$

- The gates operate on the multi qu-bit $|\psi_{in}\rangle = |\psi_0\rangle \otimes |\psi_1\rangle$
- To apply the gates, we need to combine them in a 2 qu-bit gate. How?
- Tensor Product!

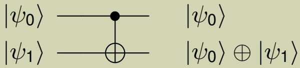
$$|\psi_{out}\rangle = (H \otimes X)\,|\psi_{in}\rangle = \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}\begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix} = \frac{1}{\sqrt{2}}\begin{bmatrix}\begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix} & \begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix}\\ \begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix} & \begin{bmatrix}0 & -1\\ -1 & 0\end{bmatrix}\end{bmatrix}$$

$$|\psi_{in}\rangle = |00\rangle \rightarrow |\psi_{out}\rangle = \frac{1}{\sqrt{2}}\begin{bmatrix}0 & 1 & 0 & 1\\ 1 & 0 & 1 & 0\\ 0 & 1 & 0 & -1\\ 1 & 0 & -1 & 0\end{bmatrix}\begin{bmatrix}1\\ 0\\ 0\\ 0\end{bmatrix} = \begin{bmatrix}0\\ \frac{1}{\sqrt{2}}\\ 0\\ \frac{1}{\sqrt{2}}\end{bmatrix} = \frac{1}{\sqrt{2}}(|01\rangle + |11\rangle)$$

Multi Qu-Bit Gates

There are also quantum gates which apply only on multiple qu-bits

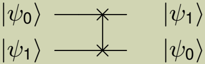
CNOT Gate



- If the control bit ($|\psi_0\rangle$) is 1, then the target bit ($|\psi_1\rangle$) is inverted
- The gate matrix is already 4×4: $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$
- Superpostion state: act independently on the fundamental states
  - $\hookrightarrow$ Example: $|\psi_{in}\rangle = \frac{\sqrt{3}}{2}|00\rangle + \frac{1}{2}|10\rangle \rightarrow |\psi_{out}\rangle = \frac{\sqrt{3}}{2}|00\rangle + \frac{1}{2}|11\rangle$

Multi Qu-Bit Gates: More Examples

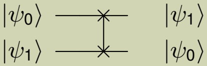
Swap Gate



- The Qu-Bits are swapped
- Gate Matrix: $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

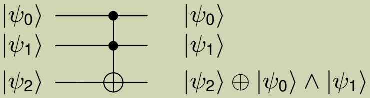
Multi Qu-Bit Gates: More Examples

Swap Gate



- The Qu-Bits are swapped
- Gate Matrix: $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

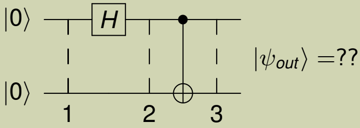
Toffoli Gate



- CNOT gate with 2 control bits instead of 1
- That is, the target bit ($|\psi_2\rangle$) is inverted when both control bits are 1

Circuits with Multi Qu-Bits Gates

Example

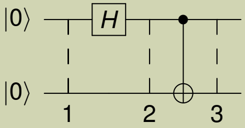


Step-by-Step evaluation:

1. $|\psi_{in}\rangle = |00\rangle$
2. $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\,|0\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |10\rangle)$
3. $|\psi_{out}\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$

Circuits with Multi Qu-Bits Gates

Example



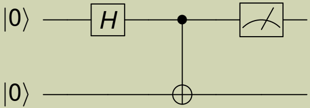
$|\psi_{out}\rangle = ??$

Step-by-Step evaluation:

1. $|\psi_{in}\rangle = |00\rangle$
2. $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\,|0\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |10\rangle)$
3. $|\psi_{out}\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$

Measuring 1 Qu-Bit

Introduce a slight variation of the above circuit:

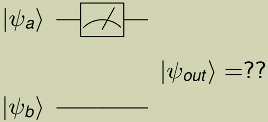


$|\psi_{out}\rangle = ??$

Multi Qu-Bit Circuits: Measurement

What happens to the multi qu-bits state when not all the bits are measured?

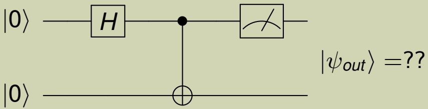
Partial Measurement: 2 Qu-Bits State



- Before measurement, the multi qu-bit $|\psi_{in}\rangle = |\psi_a\rangle \otimes |\psi_b\rangle$
- As with single bit gates, bit-wise reasoning
  1. Splitting qu-bits: Given a 2 multi qu-bit state $|\psi\rangle$, it is always possible to split is as $|\psi\rangle = \alpha_0 |0\rangle \otimes (|\psi_0\rangle) + \alpha_1 |1\rangle \otimes (|\psi_1\rangle)$, such that $|\alpha_0|^2 + |\alpha_1|^2 = 1$ and $|\psi_0\rangle, |\psi_1\rangle$ are valid qu-bits
  2. Then, depending on the measurement outcome on the first qu-bit:
     - Measurement of the first qu-bit is 0 $\rightarrow$ the second qu-bit is $|\psi_0\rangle \rightarrow |\psi_{out}\rangle = |\psi_0\rangle$
     - Measurement of the first qu-bit is 1 $\rightarrow$ the second qu-bit is $|\psi_1\rangle \rightarrow |\psi_{out}\rangle = |\psi_1\rangle$

Multi Qu-Bit Circuits: Measurement

Back to the Example Circuit

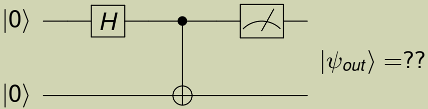


Before measurement, the multi qu-bit $|\psi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$

1. $|\psi\rangle = \frac{1}{\sqrt{2}}\,|0\rangle \otimes |0\rangle + \frac{1}{\sqrt{2}}\,|1\rangle \otimes |1\rangle$
2. $\psi_{out} = |0\rangle$ if the measured qu-bit is 0
3. $\psi_{out} = |1\rangle$ if the measured qu-bit is 1

Multi Qu-Bit Circuits: Measurement

Back to the Example Circuit



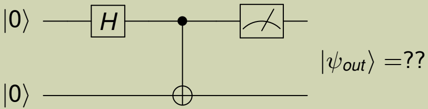
Before measurement, the multi qu-bit $|\psi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$

1. $|\psi\rangle = \frac{1}{\sqrt{2}}\,|0\rangle \otimes |0\rangle + \frac{1}{\sqrt{2}}\,|1\rangle \otimes |1\rangle$
2. $\psi_{out} = |0\rangle$ if the measured qu-bit is 0
3. $\psi_{out} = |1\rangle$ if the measured qu-bit is 1

- The measurement of a bit determines the second qu-bit
- This is weird, since the measurement affects only 1 bit

Multi Qu-Bit Circuits: Measurement

Back to the Example Circuit



Before measurement, the multi qu-bit $|\psi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$

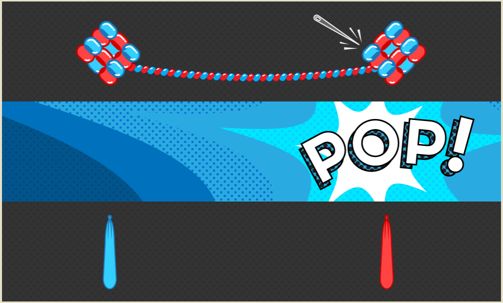
1. $|\psi\rangle = \frac{1}{\sqrt{2}}\,|0\rangle \otimes |0\rangle + \frac{1}{\sqrt{2}}\,|1\rangle \otimes |1\rangle$
2. $\psi_{out} = |0\rangle$ if the measured qu-bit is 0
3. $\psi_{out} = |1\rangle$ if the measured qu-bit is 1

- The measurement of a bit determines the second qu-bit
- This is weird, since the measurement affects only 1 bit

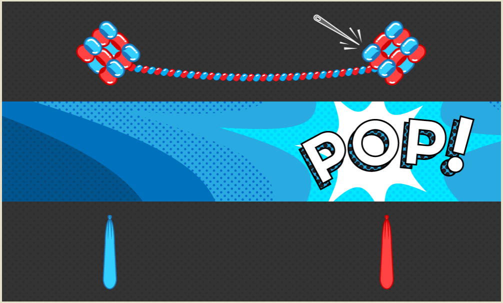
$$\Downarrow$$

A quantum phenomenon is happening: entanglement!

Multi Qu-Bit Circuits: Entanglement



Multi Qu-Bit Circuits: Entanglement



Entanglement Definition

- Recall the splitting of a 2 multi qu-bit state $|\psi\rangle = |\psi_a\rangle \otimes |\psi_b\rangle$: $|\psi\rangle = \alpha_0 |0\rangle \otimes (|\psi_0\rangle) + \alpha_1 |1\rangle \otimes (|\psi_1\rangle)$, such that $|\alpha_0|^2 + |\alpha_1|^2 = 1$ and $|\psi_0\rangle, |\psi_1\rangle$ are valid qu-bits
- $|\psi_a\rangle$ and $|\psi_b\rangle$ are entangled $\Leftrightarrow |\psi_0\rangle \neq |\psi_1\rangle$
- Meaning: the quantum state of the unmeasured bit depends on the measurement outcome of the entangled bit.

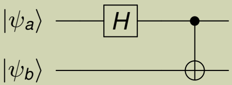
Entanglement: Examples

- $|\psi\rangle = \frac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle) = \frac{1}{\sqrt{2}}|0\rangle \otimes \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) + \frac{1}{\sqrt{2}}|1\rangle \otimes \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \rightarrow$ Not entangled!
- $|\psi\rangle = \frac{1}{2}(|00\rangle + |01\rangle + |10\rangle - |11\rangle) = \frac{1}{\sqrt{2}}|0\rangle \otimes \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) + \frac{1}{\sqrt{2}}|1\rangle \otimes \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \rightarrow$ Entangled!

Entanglement: Examples

- $|\psi\rangle = \frac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle) = \frac{1}{\sqrt{2}}|0\rangle \otimes \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) + \frac{1}{\sqrt{2}}|1\rangle \otimes \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \rightarrow$ Not entangled!
- $|\psi\rangle = \frac{1}{2}(|00\rangle + |01\rangle + |10\rangle - |11\rangle) = \frac{1}{\sqrt{2}}|0\rangle \otimes \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) + \frac{1}{\sqrt{2}}|1\rangle \otimes \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \rightarrow$ Entangled!

Bell States

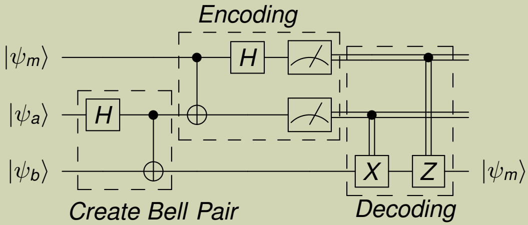


$|\psi_a\rangle \otimes |\psi_b\rangle = |\psi_{in}\rangle \in \{|00\rangle, |01\rangle, |10\rangle, |11\rangle\} \Rightarrow |\psi_a\rangle$ and $|\psi_b\rangle$ are entangled at the end of the circuit:

$\|\psi_a\rangle$	$\|\psi_b\rangle$: $\|0\rangle$	$\|\psi_b\rangle$: $\|1\rangle$
$\|0\rangle$	$\frac{1}{\sqrt{2}}(\|00\rangle + \|11\rangle)$	$\frac{1}{\sqrt{2}}(\|01\rangle + \|10\rangle)$
$\|1\rangle$	$\frac{1}{\sqrt{2}}(\|00\rangle - \|11\rangle)$	$\frac{1}{\sqrt{2}}(\|01\rangle - \|10\rangle)$

Quantum Teleportation

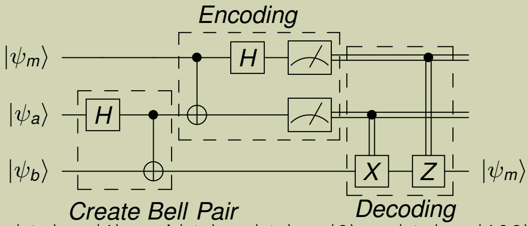
Teleportation Circuit



- The sender and the receiver generates a bell pair
- The sender keeps $|\psi_a\rangle$, while the receiver keeps $|\psi_b\rangle$
- When the sender wants to send a qu-bit $|\psi_m\rangle$, it performs encoding using $|\psi_a\rangle$ too
- 2 classical bits are sent to the receiver for the decoding procedure
- After decoding, the entangled qu-bit $|\psi_b\rangle$ has become equal to $|\psi_m\rangle$

Quantum Teleportation

Teleportation Circuit



Example with $|\psi_m\rangle = |1\rangle$ and $|\psi_a\rangle = |\psi_b\rangle = |0\rangle \rightarrow |\psi_{in}\rangle = |100\rangle$:

1. Bell pair creation: $\frac{1}{\sqrt{2}}(|100\rangle + |111\rangle)$
2. Encoding before measurement: $\frac{1}{2}(|010\rangle - |110\rangle + |001\rangle - |101\rangle) = \frac{1}{2}(|00\rangle \otimes |1\rangle + |01\rangle \otimes |0\rangle + |10\rangle \otimes (-|1\rangle) + |11\rangle \otimes (-|0\rangle))$
3. Decoding:

Measure	$\lvert\psi_b\rangle$	Corrections	Output
00	$\lvert 1\rangle$	No	$\lvert 1\rangle$
01	$\lvert 0\rangle$	X	$\lvert 1\rangle$
10	$-\lvert 1\rangle$	Z	$\lvert 1\rangle$
11	$-\lvert 0\rangle$	X,Z	$\lvert 1\rangle$

Quantum Computing Power

- With quantum teleportation, we can send N qu-bits with $2N$ classical bits
- Is it worthy?

Quantum Computing Power

- With quantum teleportation, we can send N qu-bits with $2N$ classical bits
- Is it worthy?

Qu-Bit Power

Classical Bits:

- N classical bits hold 1 single value between 0 and $2^N - 1$
- For instance, 010 is the value 2
- Classical computation performs only on the single value of the bits

Quantum Computing Power

- With quantum teleportation, we can send N qu-bits with $2N$ classical bits
- Is it worthy?

Qu-Bit Power

Classical Bits:

- N classical bits hold 1 single value between 0 and $2^N - 1$
- For instance, 010 is the value 2
- Classical computation performs only on the single value of the bits

Quantum Bits:

- N qu-bits contains all possible 2^N values representable by N bits
- For instance, $\alpha_0 |000\rangle + \alpha_1 |001\rangle + \alpha_2 |010\rangle + \alpha_3 |011\rangle + \alpha_4 |100\rangle + \alpha_5 |101\rangle + \alpha_6 |110\rangle + \alpha_7 |111\rangle$ represents all integers from 0 to 7
- Performing quantum computation is equivalent to compute at the same time with all these 2^N values. How? $\Rightarrow$ Quantum Algorithms!

Quantum Algorithms

Description

Quantum algorithms structure:

- Work on multi qu-bits in superposition states
- The operations performed on the qu-bits are chosen to get to a final superposition state
- Measuring this state generally yields the solution of the problem with probability close to 1
- Quantum algorithms have usually a classical part too, where standard bits are employed

Quantum Algorithms

Description

Quantum algorithms structure:

- Work on multi qu-bits in superposition states
- The operations performed on the qu-bits are chosen to get to a final superposition state
- Measuring this state generally yields the solution of the problem with probability close to 1
- Quantum algorithms have usually a classical part too, where standard bits are employed

Example: Integer Factorization

Classical computing for a 2048 bits number?

- 100 years
- 10^5 Trillion €
- 398549 km^2 server farm

Shor Algorithm

Quantum computation? $\Rightarrow$ 26.7 hours using Shor algorithm!

Description

- Classical reduction to the order-finding subproblem
- This sub-problem is solved with the quantum algorithm
- Some properties of the solution are tested, otherwise the procedure is repeated to yield a new solution of the subproblem

Shor Algorithm

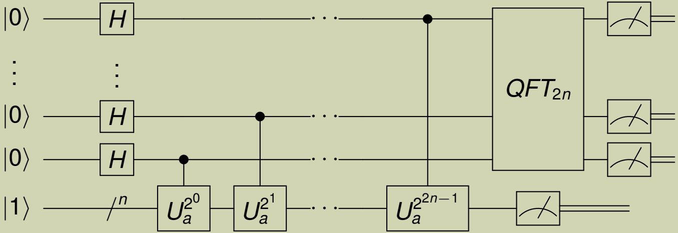
Quantum computation? $\Rightarrow$ 26.7 hours using Shor algorithm!

Description

- Classical reduction to the order-finding subproblem
- This sub-problem is solved with the quantum algorithm
- Some properties of the solution are tested, otherwise the procedure is repeated to yield a new solution of the subproblem

Order-Finding Solver

Given $f(x) = a^x \mod N$, find the period of f, i.e. the order of a



Quantum Technologies

Quantum computing is extremely powerful, but . . .

$\Downarrow$

Quantum Technologies are extremely fragile!

Quantum Technologies

Quantum computing is extremely powerful, but . . .

$\Downarrow$

Quantum Technologies are extremely fragile!

Quantum Computation Errors

- A qu-bit is affected by external noise
- For instance, the Brownian motions of the molecules may interfere with the quantum estate
- Each qu-bit has a decoherence time: The maximum time a qu-bit can keep its superposition state
- Typically in the order of tens of μs

Quantum Technologies

Quantum computing is extremely powerful, but . . .

$\Downarrow$

Quantum Technologies are extremely fragile!

Quantum Computation Errors

- A qu-bit is affected by external noise
- For instance, the Brownian motions of the molecules may interfere with the quantum estate
- Each qu-bit has a decoherence time: The maximum time a qu-bit can keep its superposition state
- Typically in the order of tens of μs

Quantum technology fragility

$\Downarrow$

Error correction codes are necessary to preserve the computation

Quantum Error Correction Codes (QECC)

Quantum Error Correction Codes

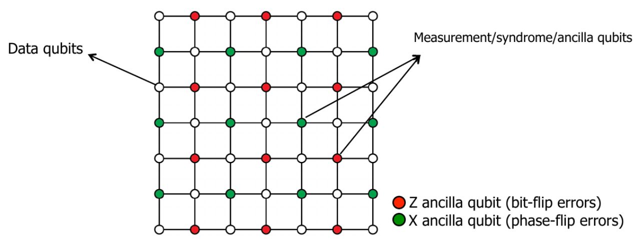
- Correction process is carried on after each operation
- As every correction code, redundancy is employed to correct errors
- A lot of reduncancy is necessary, since:
  1. Qu-bits are continuous, not discrete
  2. Error rate is high
- Each qu-bit becomes a logical qu-bit, which is encoded in *n* physical qu-bits: data and ancilla ones

Quantum Error Correction Codes (QECC)

Quantum Error Correction Codes

- Correction process is carried on after each operation
- As every correction code, redundancy is employed to correct errors
- A lot of reduncancy is necessary, since:
    1. Qu-bits are continuous, not discrete
    2. Error rate is high
- Each qu-bit becomes a logical qu-bit, which is encoded in *n* physical qu-bits: data and ancilla ones

Surface Code Logical Qu-Bit:



QECC: Logical Gates

Quantum Operations on Logical Qu-Bits

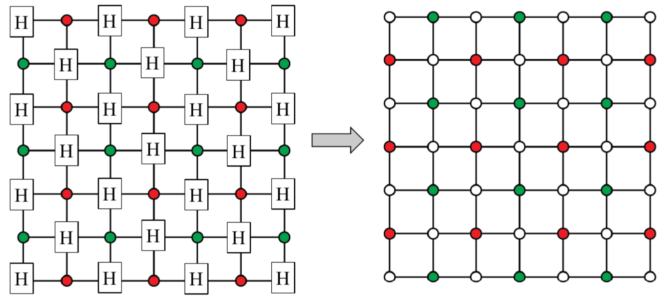
- Each of the fundamental operations (X,Z,H, . . .) needs to be defined on the logical qu-bit
- The way the logical operation is performed is code dependent
- Each logical operation is represented by a logical gate

QECC: Logical Gates

Quantum Operations on Logical Qu-Bits

- Each of the fundamental operations (X,Z,H, . . .) needs to be defined on the logical qu-bit
- The way the logical operation is performed is code dependent
- Each logical operation is represented by a logical gate

Surface Code Hadamar Gate:



QECC Overhead

Error correction is the main responsible for the blowup of qu-bits required for a quantum algorithm

Shor Algorithm Overhead

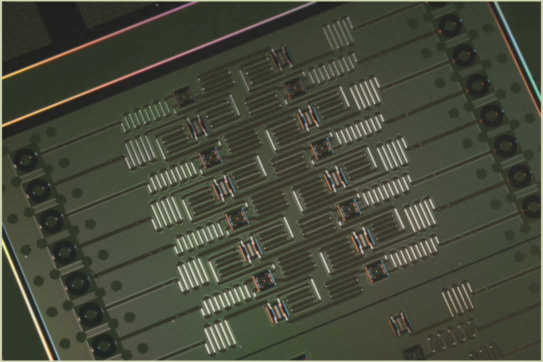
For instance, Shor algorithm on $L = 2048$ bits number requires:

Rationale	#Physical Qu-bits (cumulative)
$6L$ logical qu-bits	$12,288$
$8\times$ ancilla qu-bits	$98,304$
$1.33\times$ to provide 'wiring' room to move qu-bits	$133,000$
$10k\times$ surface code	$1.3bn$
$4x$ micro-architecture details	$5.2bn$

Quantum Technologies

How Many Qu-Bits?

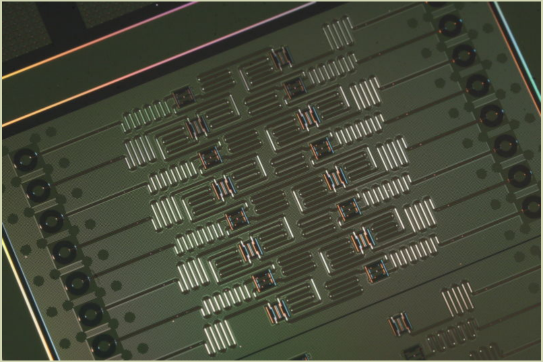
16 qu-bit IBM quantum processor, publicly available online:



Quantum Technologies

How Many Qu-Bits?

16 qu-bit IBM quantum processor, publicly available online:



IBM Trend:

TECH —

IBM will sell 50-qubit universal quantum computer "in the next few years"

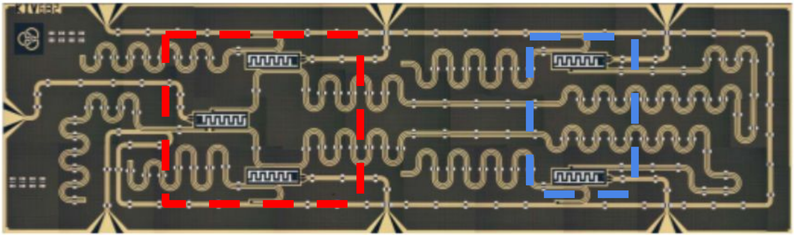
IBM has solved most of the science behind quantum computing. Time to make some money.

SEBASTIAN ANTHONY - 6/3/2017, 12:59

Quantum Technologies

Quantum Chips

5 qu-bits chip scheme for a logical qu-bit:



- data qu-bits, ancilla qu-bits
- The gates are implemented via microwave pulses ($10^{-8}s$) sent to the qu-bits
- We want to perform different gates within the decoherence time

Quantum Computation Concepts

Quantum Processor: A chip where there are *n* available qu-bits

Quantum Computation Concepts

Quantum Processor:	A chip where there are *n* available qu-bits
Quantum Programming Language:	A language to describe a circuit using gate-level instructions or known functions

Quantum Computation Concepts

Quantum Processor:	A chip where there are *n* available qu-bits
Quantum Programming Language:	A language to describe a circuit using gate-level instructions or known functions
Quantum Algorithm:	A quantum circuit to be executed

Quantum Computation Concepts

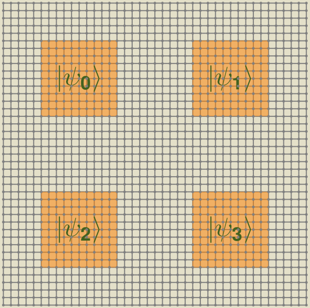
Quantum Processor:	A chip where there are *n* available qu-bits
Quantum Programming Language:	A language to describe a circuit using gate-level instructions or known functions
Quantum Algorithm:	A quantum circuit to be executed
Quantum Compilation:	***Optimization*** Use heuristics to merge gates or re-arrange operations ***Placement*** Initial Mapping of the circuit qu-bits to the on-chip qu-bits ***Scheduling*** Schedule gates execution ***Routing*** Move qu-bits to execute multibit operations

Quantum Computation Concepts

Quantum Processor:	A chip where there are *n* available qu-bits
Quantum Programming Language:	A language to describe a circuit using gate-level instructions or known functions
Quantum Algorithm:	A quantum circuit to be executed
Quantum Compilation:	***Optimization*** Use heuristics to merge gates or re-arrange operations ***Placement*** Initial Mapping of the circuit qu-bits to the on-chip qu-bits ***Scheduling*** Schedule gates execution ***Routing*** Move qu-bits to execute multibit operations
Quantum Execution:	Translation of gate-level instructions (QISA) to signals sent to the processor

Quantum Compilation

Logical view of the chip:



Main issue to be addressed (imposed by the technology) : 2 input qu-bits of a non single gate (e.g. CNOT) needs to be adjiacent to compute the gate

$\hookrightarrow$ 2 qu-bits are adjacent if they are either on the same row or on the same column

$\hookrightarrow$ If they are not adjacent, they need to be moved to satisfy this constraint (routing process)

Quantum Compilation: Placement

Placement

Maps the qu-bits on chip, deriving the initial configuration of the processor

- We want the qu-bits which are combined in a 2 qu-bits gate to be placed as close as possible

Quantum Compilation: Placement

Placement

Maps the qu-bits on chip, deriving the initial configuration of the processor

- We want the qu-bits which are combined in a 2 qu-bits gate to be placed as close as possible
- How close? Minimizing Manhattan distance $\equiv$ Minimizing routing cost

$\lvert\psi_0\rangle$	$\lvert\psi_1\rangle$	$\lvert\psi_2\rangle$	$\lvert\psi_3\rangle$
$\lvert\psi_4\rangle$	$\lvert\psi_5\rangle$	$\lvert\psi_6\rangle$	

E.g $CNOT(|\psi_3\rangle, |\psi_4\rangle) \rightarrow d = 4$

Quantum Compilation: Placement

Placement

Maps the qu-bits on chip, deriving the initial configuration of the processor

- We want the qu-bits which are combined in a 2 qu-bits gate to be placed as close as possible
- How close? Minimizing Manhattan distance $\equiv$ Minimizing routing cost

$\lvert\psi_0\rangle$	$\lvert\psi_1\rangle$	$\lvert\psi_2\rangle$	$\lvert\psi_3\rangle$
$\lvert\psi_4\rangle$	$\lvert\psi_5\rangle$	$\lvert\psi_6\rangle$	

E.g $CNOT(\lvert\psi_3\rangle, \lvert\psi_4\rangle) \rightarrow d = 4$

- Target: finding the placement minimizing the sum of Manhattan distances over all pairs involved in multiple bits gates

Quantum Compilation: Placement

Placement

Maps the qu-bits on chip, deriving the initial configuration of the processor

- We want the qu-bits which are combined in a 2 qu-bits gate to be placed as close as possible
- How close? Minimizing Manhattan distance $\equiv$ Minimizing routing cost

$\lvert\psi_0\rangle$	$\lvert\psi_1\rangle$	$\lvert\psi_2\rangle$	$\lvert\psi_3\rangle$
$\lvert\psi_4\rangle$	$\lvert\psi_5\rangle$	$\lvert\psi_6\rangle$	

E.g $CNOT(\lvert\psi_3\rangle, \lvert\psi_4\rangle) \rightarrow d = 4$

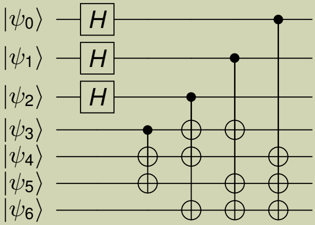
- Target: finding the placement minimizing the sum of Manhattan distances over all pairs involved in multiple bits gates
- One possible approach: Quantum Interaction Graph (QIG)

Quantum Compilation: QIG

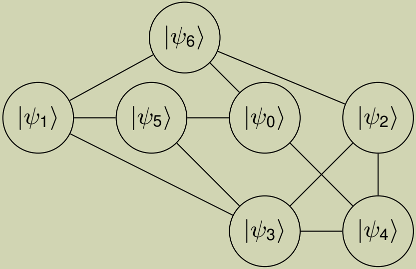
Quantum Interaction Graph

- QIG purpose: represent the relationships between qu-bits involved in multiple bits gates
- The corresponding symmetric matrix can be used to define a linear programming problem
- The solution of this problem provide a good placement.

Example Circuit:



Quantum Interaction Graph:



Quantum Compilation: Scheduling

Gates can theoretically be all executed simultaneously, but:

Scheduling Issues

- Data dependencies
- Privileged Writings on each qu-bit
- Out of order execution must preserve the correctness of the computation

Quantum Compilation: Scheduling

Gates can theoretically be all executed simultaneously, but:

Scheduling Issues

- Data dependencies
- Privileged Writings on each qu-bit
- Out of order execution must preserve the correctness of the computation

Scheduling Policy

- An As Soon As Possible (ASAP) policy is usually employed
  - An operation is performed as soon as the input data are available
- Mainly due to decoherence time constraints, As Late As Possible (ALAP) policy is generally preferable in quantum scenarios
- Try to minimize the time between an operation writing a qu-bit and the next operation reading it $\rightarrow$ reducing the time interval the quantum state needs to be preserved

Quantum Computation: ASAP vs ALAP

ASAP Policy

C0: .init $|\psi_0\rangle$, $|\psi_1\rangle$, $|\psi_2\rangle$, $|\psi_3\rangle$, $|\psi_4\rangle$, $|\psi_5\rangle$, $|\psi_6\rangle$

C1: $H(|\psi_0\rangle)$, $H(|\psi_1\rangle)$, $H(|\psi_2\rangle)$, $CNOT(|\psi_3\rangle , |\psi_4\rangle)$, $CNOT(|\psi_3, \psi_5\rangle)$

C2: $CNOT(|\psi_2\rangle , |\psi_3\rangle)$, $CNOT(|\psi_2\rangle , |\psi_4\rangle)$, $CNOT(|\psi_2\rangle , |\psi_6\rangle)$, $CNOT(|\psi_1\rangle , |\psi_5\rangle)$

C3: $CNOT(|\psi_1\rangle , |\psi_3\rangle)$, $CNOT(|\psi_1\rangle , |\psi_6\rangle)$, $CNOT(|\psi_0\rangle , |\psi_4\rangle)$, $CNOT(|\psi_0\rangle , |\psi_5\rangle)$

C4: $CNOT(|\psi_0\rangle , |\psi_6\rangle)$

Quantum Computation: ASAP vs ALAP

ASAP Policy

C0: .init $|\psi_0\rangle$, $|\psi_1\rangle$, $|\psi_2\rangle$, $|\psi_3\rangle$, $|\psi_4\rangle$, $|\psi_5\rangle$, $|\psi_6\rangle$

C1: $H(|\psi_0\rangle)$, $H(|\psi_1\rangle)$, $H(|\psi_2\rangle)$, $CNOT(|\psi_3\rangle , |\psi_4\rangle)$, $CNOT(|\psi_3, \psi_5\rangle)$

C2: $CNOT(|\psi_2\rangle , |\psi_3\rangle)$, $CNOT(|\psi_2\rangle , |\psi_4\rangle)$, $CNOT(|\psi_2\rangle , |\psi_6\rangle)$, $CNOT(|\psi_1\rangle , |\psi_5\rangle)$

C3: $CNOT(|\psi_1\rangle , |\psi_3\rangle)$, $CNOT(|\psi_1\rangle , |\psi_6\rangle)$, $CNOT(|\psi_0\rangle , |\psi_4\rangle)$, $CNOT(|\psi_0\rangle , |\psi_5\rangle)$

C4: $CNOT(|\psi_0\rangle , |\psi_6\rangle)$

ALAP Policy

C0: .init $|\psi_2\rangle$

C1: .init $|\psi_1\rangle$, $|\psi_3\rangle$, $|\psi_4\rangle$, $|\psi_5\rangle$, $|\psi_6\rangle$, $H(|\psi_2\rangle)$

C2: .init $|\psi_0\rangle$, $H(|\psi_1\rangle)$, $CNOT(|\psi_3\rangle , |\psi_4\rangle)$, $CNOT(|\psi_3, \psi_5\rangle)$, $CNOT(|\psi_2\rangle , |\psi_6\rangle)$

C3: $H(|\psi_0\rangle)$, $CNOT(|\psi_2\rangle , |\psi_3\rangle)$, $CNOT(|\psi_2\rangle , |\psi_4\rangle)$, $CNOT(|\psi_1\rangle , |\psi_6\rangle)$, $CNOT(|\psi_1\rangle , |\psi_5\rangle)$

C4: $CNOT(|\psi_1\rangle , |\psi_3\rangle)$, $CNOT(|\psi_0\rangle , |\psi_4\rangle)$, $CNOT(|\psi_0\rangle , |\psi_5\rangle)$, $CNOT(|\psi_0\rangle , |\psi_6\rangle)$

Postpone initialization as late as possible!

Quantum Compilation: Routing

Recall: to perform multi qu-bit gates the qu-bits need to be adjacent

$\hookrightarrow$ If they are not, we need to move them. How?

$\hookrightarrow$ Using swap gate!

Quantum Compilation: Routing

Recall: to perform multi qu-bit gates the qu-bits need to be adjacent

$\hookrightarrow$ If they are not, we need to move them. How?

$\hookrightarrow$ Using swap gate!

Routing Example

Consider this initial placement obtained from our linear programing algorithm:

$\lvert\psi_6\rangle$	$\lvert\psi_0\rangle$	$\lvert\psi_1\rangle$	$\lvert\psi_5\rangle$
$\lvert\psi_2\rangle$	$\lvert\psi_4\rangle$	$\lvert\psi_3\rangle$	

Now, consider the third cycle using ALAP scheduling policy:

.init $|\psi_0\rangle$, $H(|\psi_1\rangle)$, $CNOT(|\psi_3, |\psi_4\rangle)\rangle$, $CNOT(|\psi_3, \psi_5\rangle)$, $CNOT(|\psi_2\rangle , |\psi_6\rangle)$

$\hookrightarrow$ We need to add a swap between $|\psi_5\rangle$ and $|\psi_1\rangle$

$\hookrightarrow$ .init $|\psi_0\rangle$, $H(|\psi_1\rangle)$, $CNOT(|\psi_3, |\psi_4\rangle)\rangle$, $SWAP(|\psi_1, \psi_5\rangle)$, $CNOT(|\psi_2\rangle , |\psi_6\rangle)$

Quantum Compilation: Routing

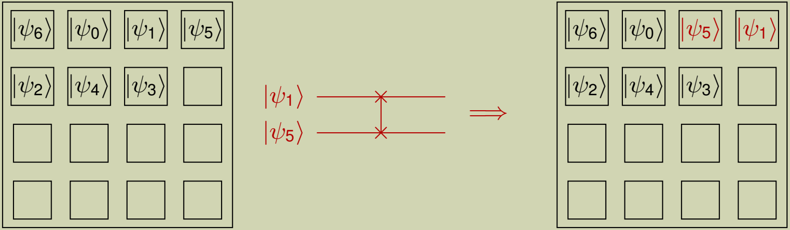
Recall: to perform multi qu-bit gates the qu-bits need to be adjacent

↪ If they are not, we need to move them. How?

↪ Using swap gate!

Routing Example

Consider this initial placement obtained from our linear programing algorithm:



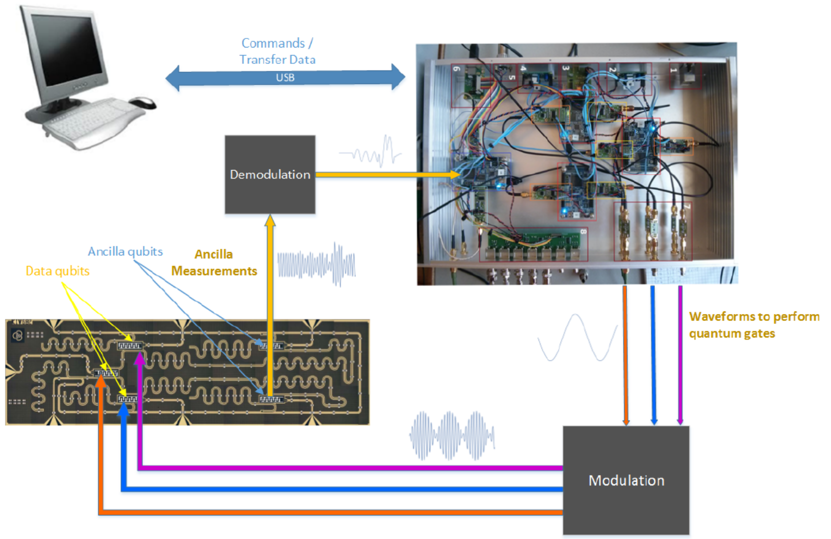
Now, consider the third cycle using ALAP scheduling policy:

.init $|\psi_0\rangle$, $H(|\psi_1\rangle)$, $CNOT(|\psi_3, |\psi_4\rangle)\rangle$, $CNOT(|\psi_3, \psi_5\rangle)$, $CNOT(|\psi_2\rangle , |\psi_6\rangle)$

↪ We need to add a swap between $|\psi_5\rangle$ and $|\psi_1\rangle$

↪ .init $|\psi_0\rangle$, $H(|\psi_1\rangle)$, $CNOT(|\psi_3, |\psi_4\rangle)\rangle$, $SWAP(|\psi_1, \psi_5\rangle)$, $CNOT(|\psi_2\rangle , |\psi_6\rangle)$

Quantum Micro-architecture



Cooling Power

Technical Challenges

- Qu-bits can preserve their states only at really low temperatures (mK order)
- They need to interact with electronic components $\rightarrow$ the heat generated by these components should not affect the chip
- Cooling methods:
  1. Heat bath: liquid helium is employed
  2. In 2017, a quantum refrigerator chip based on tunnel effect has been proposed

Shielding & Wiring

Shielding

- The quantum computer needs to be isolated from external electro-magnetic waves
- The quantum processor is shielded with magnetic elements which zero the magnetic field coming from electronic circuitry

Shielding & Wiring

Shielding

- The quantum computer needs to be isolated from external electro-magnetic waves
- The quantum processor is shielded with magnetic elements which zero the magnetic field coming from electronic circuitry

Wiring

- Wiring: with a lot of qu-bits, placement of wires to perform operations may become complex
- In particular, interferences among different wires is a relevant issue
- Due to:
  1. Wires need to work at extremely low temperatures
  2. Material used to build wires cannot be magnetic

$$\Downarrow$$

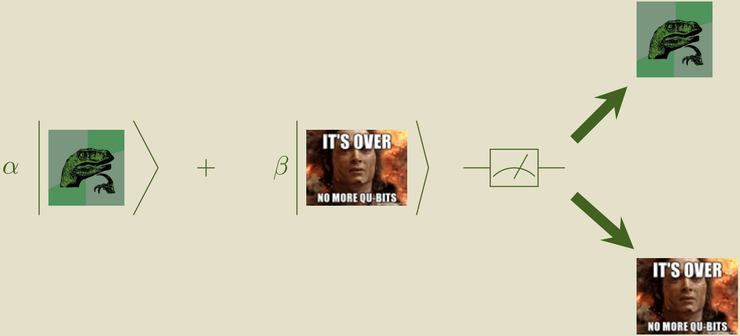
non conventional material is necessary

Conclusion

Take Home Messages

- Qu-bit superposition state allows to represent simultaneously both 0 and 1
- Qu-bit measurement is probabilistic
- Quantum gates are reversible
- Entanglement: 2 entangled qu-bits are strictly linked, and operations performed on one qu-bit may affect the other one too
- Entanglement can be used for quantum teleportation
- Exponential improvement: N qu-bits allow to represent 2^N values
- Quantum Algorithm idea: perform computation which yields to high probability of measuring the correct result
- Quantum technology is fragile $\rightarrow$ error correction is necessary
- Quantum processor are nowadays too limited for practical application
- Quantum computer architecture & compilation
- There are relevant technical challenges to build a quantum computer

Questions



Quantum Compilation: Phases Overview

- From the previous example, we can see that routing affects the scheduling
- Swaps are additional gates, which introduce new constraints
- But we cannot properly insert swaps if we do not know the scheduling of the operations

$\Downarrow$

Routing & Scheduling should be performed together given the initial placement

Quantum Compilation: Phases Overview

- From the previous example, we can see that routing affects the scheduling
- Swaps are additional gates, which introduce new constraints
- But we cannot properly insert swaps if we do not know the scheduling of the operations

$\Downarrow$

Routing & Scheduling should be performed together given the initial placement

Optimal Solution?

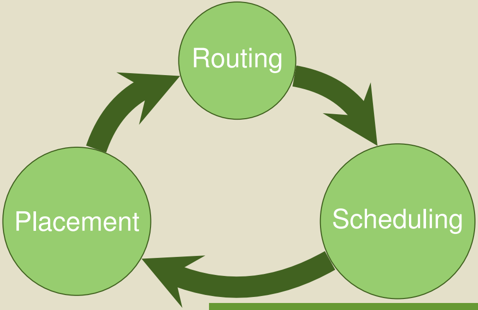
- Routing cost is estimated on the initial placement for all gates
- But the placement of the qu-bits changes during the execution
- The solution may not be optimal!
- However, if we consider the temporal dependencies during placement the problem becomes more complex: scheduling of the operations is relevant too!

Quantum Compilation: Phases Overview

- From the previous example, we can see that routing affects the scheduling
- Swaps are additional gates, which introduce new constraints
- But we cannot properly insert swaps if we do not know the scheduling of the operations

$\Downarrow$

Routing & Scheduling should be performed together given the initial placement



Useful Resources

- 5 qu-bits IBM processor under the curtains: https://arstechnica.com/science/2016/05/how-ibms-new-five-qubit-universal-quantum-computer-works/
- IBM Quantum Experience: https://quantumexperience.ng.bluemix.net/qx
- Quantum Computer Simulator: http://quantum-studio.net/

Quantum Teleportation

Generic Qu-Bit Computation

$|\psi_m\rangle = \alpha_0 |0\rangle + \alpha_1 |1\rangle$, $|\psi_a\rangle = |\psi_b\rangle = |0\rangle \Rightarrow |\psi_{in}\rangle = \alpha_0 |000\rangle + \alpha_1 |100\rangle$

1. Bell pair creation: $\alpha_0 \frac{1}{\sqrt{2}}(|000\rangle + |011\rangle) + \alpha_1 \frac{1}{\sqrt{2}}(|100\rangle + |111\rangle)$
2. CNOT gate: $\alpha_0 \frac{1}{\sqrt{2}}(|000\rangle + |011\rangle) + \alpha_1 \frac{1}{\sqrt{2}}(|110\rangle + |101\rangle)$
3. Hadamar gate:
   $\alpha_0 \frac{1}{2}(|000\rangle + |100\rangle + |011\rangle + |111\rangle) + \alpha_1 \frac{1}{2}(|010\rangle - |110\rangle + |001\rangle - |101\rangle)$
4. Split before measurement: $\frac{1}{2}|00\rangle \otimes (\alpha_0 |0\rangle + \alpha_1 |1\rangle) + \frac{1}{2}|01\rangle \otimes (\alpha_1 |0\rangle + \alpha_0 |1\rangle) + \frac{1}{2}|10\rangle \otimes (\alpha_0 |0\rangle - \alpha_1 |1\rangle) + \frac{1}{2}|11\rangle \otimes (-\alpha_1 |0\rangle + \alpha_0 |1\rangle)$
5. Decoding:

Measure	$\lvert\psi_b\rangle$	Corrections	Output
00	$\alpha_0 \lvert 0\rangle + \alpha_1 \lvert 1\rangle$	No	$\alpha_0 \lvert 0\rangle + \alpha_1 \lvert 1\rangle$
01	$\alpha_1 \lvert 0\rangle + \alpha_0 \lvert 1\rangle$	X	$\alpha_0 \lvert 0\rangle + \alpha_1 \lvert 1\rangle$
10	$\alpha_0 \lvert 0\rangle - \alpha_1 \lvert 1\rangle$	Z	$\alpha_0 \lvert 0\rangle + \alpha_1 \lvert 1\rangle$
11	$-\alpha_1 \lvert 0\rangle + \alpha_0 \lvert 1\rangle$	X,Z	$\alpha_0 \lvert 0\rangle + \alpha_1 \lvert 1\rangle$