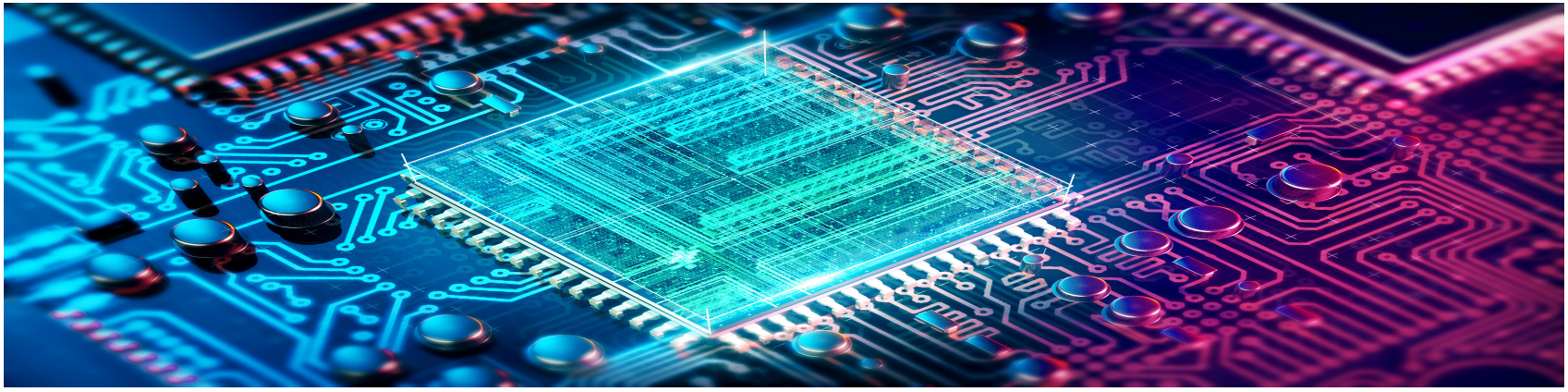
# Cómo programar un ordenador cuántico sin morir en el intento

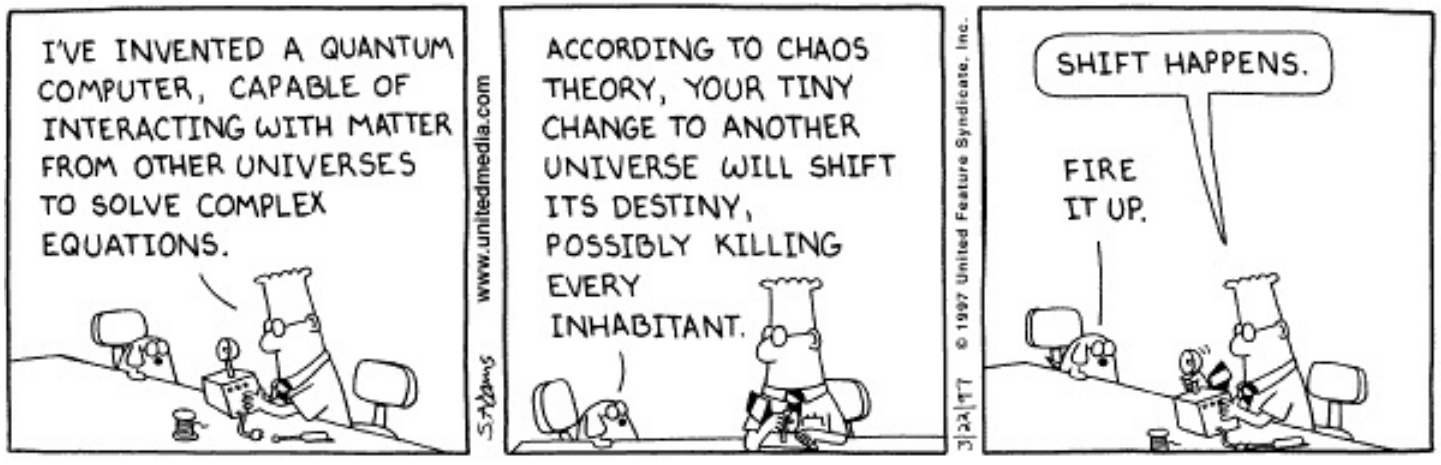
**Elías F. Combarro (Universidad de Oviedo)**
efernandezca@uniovi.es
https://github.com/EliasCombarro/EIITechFest2020

EII Oviedo - 30 de enero de 2020



# Objetivos de este taller

- Comprender los dos modelos principales de programación cuántica:
  - Circuitos cuánticos
  - Computación adiabática
- Conocer algunas de las librerías principales de programación cuántica
- Ejecutar programas en:
  - Simuladores
  - **Dos** ordenadores cuánticos reales distintos



# Nuestras herramientas



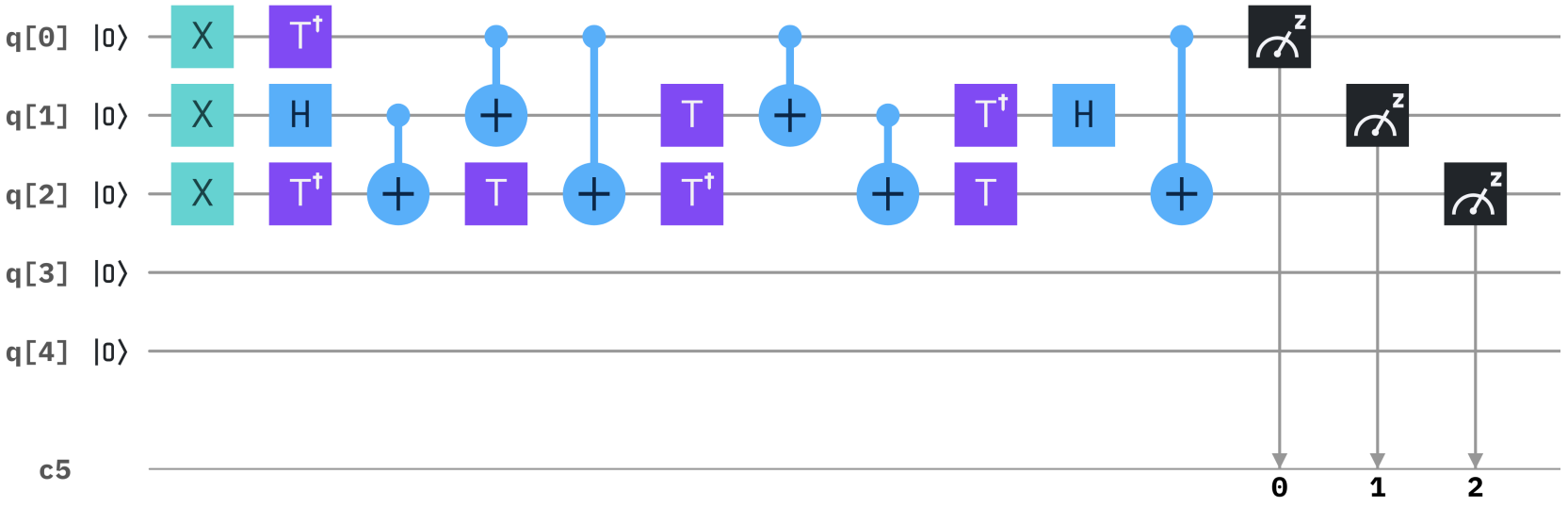




# Así es un ordenador cuántico basado en circuitos

# Elementos de un circuito cuántico

- Datos = **qubits**
- Operaciones = **puertas cuánticas**
- Resultados = **mediciones**



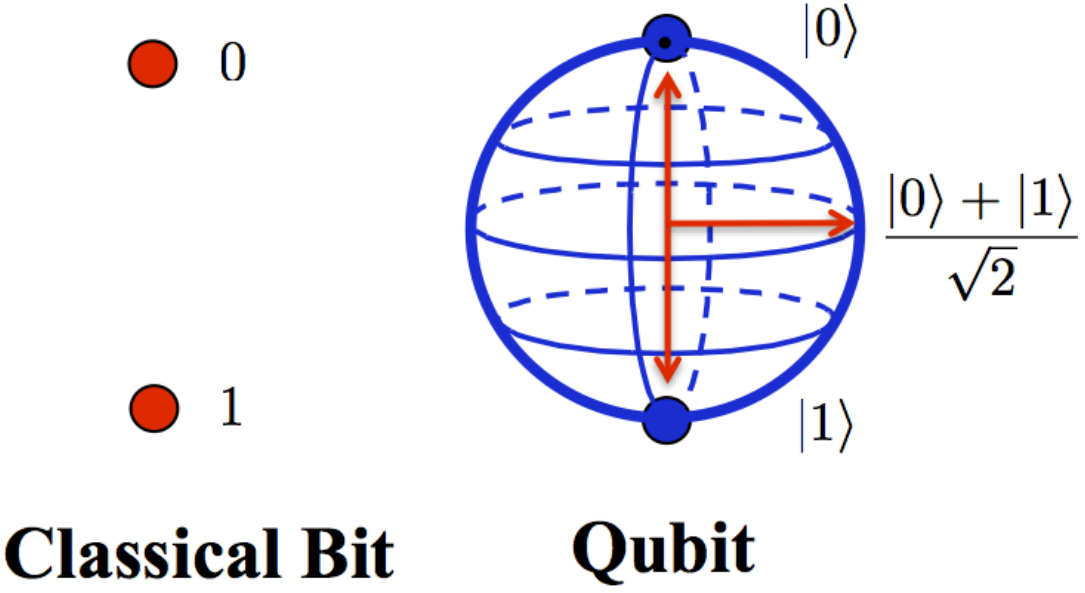
# ¿Qué es un qubit?

- Un qubit genérico tiene la forma de una **superposición**

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$$

donde $\alpha$ y $\beta$ son **números complejos** que cumplen

$$|\alpha|^2 + |\beta|^2 = 1$$



# Midiendo un qubit

- Al medir el qubit

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$$

obtenemos

0 con probabilidad $|\alpha|^2$

o

1 con probabilidad $|\beta|^2$

# Sistemas de $n$ qubits

- Tenemos $2^n$ posibilidades:

$$|00\ldots0\rangle, |00\ldots1\rangle, \ldots, |11\ldots1\rangle$$

- Un estado genérico del sistema será

$$|\psi\rangle = \alpha_0 |0\rangle + \alpha_1 |1\rangle + \ldots + \alpha_{2^n-1} \left|2^n - 1\right\rangle$$

  donde los $\alpha_i$ son números complejos que cumplen

$$\sum_{i=0}^{2^n-1} |\alpha_i|^2 = 1$$

- Si lo medimos obtenemos
  - 0 con probabilidad $|\alpha_0|^2$
  - 1 con probabilidad $|\alpha_1|^2$
  - $\ldots$
  - $2^n - 1$ con probabilidad $|\alpha_{2^n-1}|^2$

# Puertas cuánticas

- Las operaciones que se pueden hacer con un obedecen a la ecuación de Schrödinger

$$H(t)|\psi(t)\rangle = i\hbar\frac{\partial}{\partial t}|\psi(t)\rangle$$

- Eesto implica que las puertas cuánticas deben ser matrices matrices unitarias:

$$UU^{\dagger} = U^{\dagger}U = I$$

donde $U^{\dagger}$ es la transpuesta conjugada de U.

# La puerta *X* o *NOT*

- La puerta *X* viene definida por la matriz (unitaria)

$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

- Su acción es (notación del modelo de circuitos)

$$|0\rangle \; \boxed{X} \; |1\rangle$$

$$|1\rangle \; \boxed{X} \; |0\rangle$$

  es decir, actúa como un *NOT*
- Su acción sobre un qubit general sería

$$\alpha\,|0\rangle + \beta\,|1\rangle \; \boxed{X} \; \beta\,|0\rangle + \alpha\,|1\rangle$$

# La puerta $H$

- La puerta $H$ o puerta de Hadamard viene definida por la matriz (unitaria)

$$\frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

- Su acción es

$$|0\rangle \;\text{—}\boxed{H}\text{—}\; \frac{|0\rangle+|1\rangle}{\sqrt{2}}$$

$$|1\rangle \;\text{—}\boxed{H}\text{—}\; \frac{|0\rangle-|1\rangle}{\sqrt{2}}$$

- Se suele denotar

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

y

$$|-\rangle := \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

# La puerta $Z$

- La puerta $Z$ viene definida por la matriz (unitaria)

$$\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

- Su acción es

$$|0\rangle \text{ —}\boxed{Z}\text{— } |0\rangle$$

$$|1\rangle \text{ —}\boxed{Z}\text{— } -|1\rangle$$

# La puerta *CNOT*

- La puerta *CNOT* (controlled-NOT) viene definida por la matriz (unitaria)

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

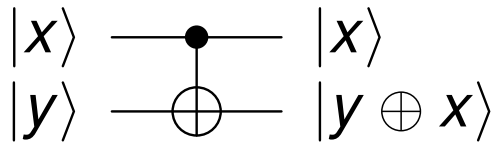
- Si el primer qubit es $|0\rangle$, no se hace nada. Si es $|1\rangle$, se invierte el segundo qubit (y el primero queda igual)
- Es decir:

$$|00\rangle \to |00\rangle \qquad |01\rangle \to |01\rangle$$

$$|10\rangle \to |11\rangle \qquad |11\rangle \to |10\rangle$$

# La puerta *CNOT*

- Su acción con elementos $x, y \in \{0, 1\}$ es, por tanto:




- Es una puerta muy importante, puesto que nos permite:
  - Realizar entrelazamientos (más sobre ello enseguida)
  - Copiar información clásica, ya que:

$$|00\rangle \to |00\rangle$$

$$|10\rangle \to |11\rangle$$

# Otras puertas

- Puerta $Y$

$$\begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$$

- Puerta $T$

$$\begin{pmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{4}} \end{pmatrix}$$

- Puerta $S$

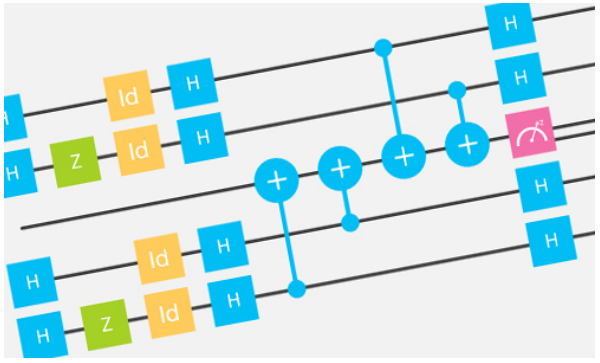
$$\begin{pmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{2}} \end{pmatrix}$$

- La puerta $R(\alpha)$ o puerta de fase, que depende de un parámetro (el ángulo $\alpha$)

$$\begin{pmatrix} 1 & 0 \\ 0 & e^{i\alpha} \end{pmatrix}$$

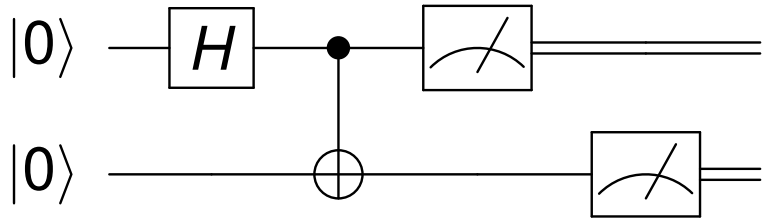
# ¿Qué diferencias hay entre los circuitos cuánticos y los algoritmos clásicos?

- Las operaciones siempre son reversibles
- La computación es probabilista
- El vector tiene tamaño exponencial en el número de qubits (**paralelismo cuántico**)
- Se pueden producir efectos de interferencia
- Podemos crear entrelazamiento
- No podemos copiar información en general



# Hello, entangled world!

- Podemos construir un estado entrelazado con un circuito sencillo




- Obtenemos el estado

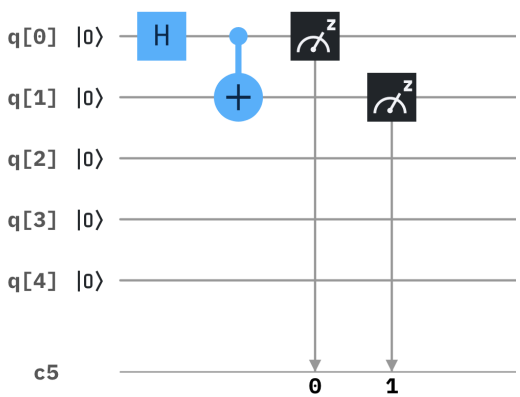
$$\frac{|00\rangle + |11\rangle}{\sqrt{2}}$$

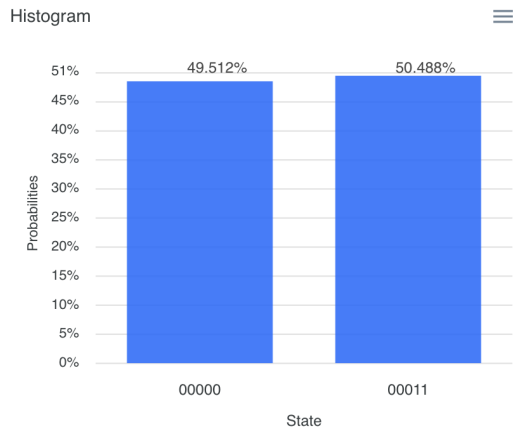
- Si medimos cada qubit por separado, la probabilidad de obtener 0 (o 1) es del 50 %
- Pero cuando medimos un qubit, el valor del otro queda completamente determinado

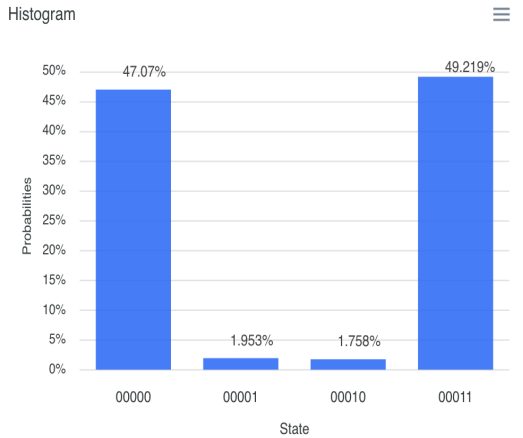
# Ejecución del circuito

Ejecutaremos nuestro primer programa cuántico en la IBM Quantum Experience

```
1   OPENQASM 2.0;
2   include "qelib1.inc";
3
4   qreg q[5];
5   creg c[5];
6
7   h q[0];
8   cx q[0],q[1];
9   measure q[0] -> c[0];
10  measure q[1] -> c[1];
```

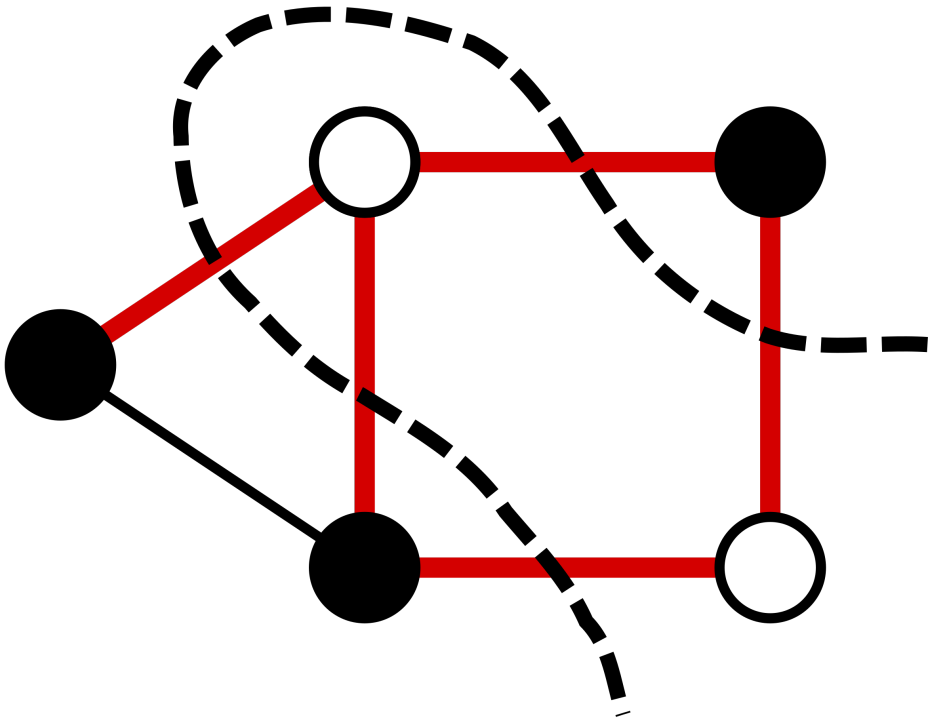


# El problema del corte máximo

- Consideremos el problema de dividir los vértices de un grafo en dos grupos maximizando los ejes cortados



- Es un problema NP-hard (si podemos resolverlo, podemos resolver cualquier problema que esté en NP)

# Planteando el problema del corte máximo con *spins*

- Identificamos cada vértice $i$ del grafo con una variable $Z_i$
- Asignamos valor 1 a los vértices de un grupo y -1 a los del otro
- Entonces, si $E$ es el conjunto de ejes, el problema se puede plantear como

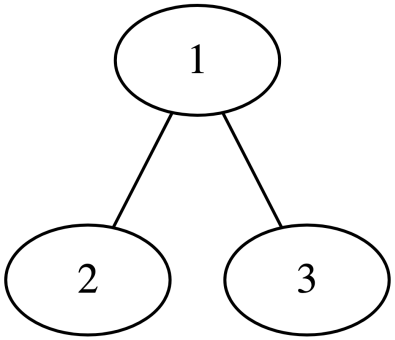
$$\text{Minimizar} \sum_{(i,j)\in E} Z_i Z_j$$

ya que vértices en distintos grupos aportan -1 a la suma y vértices del mismo grupo aportan 1

# Ejemplo de corte máximo

- Para el grafo de la figura se trata de minimizar

$$H = Z_1 Z_2 + Z_1 Z_3$$




- Por inspección (o enumerando todas las posibilidades) se ve que las soluciones óptimas son 011 y 100

# ¿Y dónde metemos la computación cuántica en todo esto?

- Recordemos que la puerta $Z$ tiene como matriz

$$\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

y que el vector $|0\rangle$ tiene como coordenadas

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

- Entonces

$$\begin{pmatrix} 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = 1$$

- Solemos denotar el anterior producto de matrices y vectores como

$$\langle 0| Z |0\rangle = 1$$

# ¿Y dónde metemos la computación cuántica en todo esto?

- Análogamente

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

- Así que

$$\langle 1| Z |1\rangle = \begin{pmatrix} 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = -1$$

- Si tenemos más qubits, evaluamos cada uno por separado y multiplicamos. Por ejemplo:

$$\langle 01| Z_1 Z_2 |01\rangle = (\langle 0| Z_1 |0\rangle) \cdot (\langle 1| Z_2 |1\rangle) = 1 \cdot (-1) = -1$$

y

$$\langle 101| Z_1 Z_3 |101\rangle = (\langle 1| Z_1 |1\rangle) \cdot (\langle 1| Z_3 |1\rangle) = (-1) \cdot (-1) = 1$$

# Volviendo al ejemplo de corte máximo

- Teníamos el problema de corte representado por

$$H = Z_1 Z_2 + Z_1 Z_3$$

- Podemos identificar un posible corte con $|011\rangle$ (tomar los vértices 2 y 3 y dejar fuera el 1) y evaluar su coste mediante

$$\begin{aligned} &\langle 011| H |011\rangle = \langle 011| (Z_1 Z_2 + Z_1 Z_3) |011\rangle \\ &= \langle 011| Z_1 Z_2 |011\rangle + \langle 011| (Z_1 Z_3) |011\rangle = -1 + (-1) = -2 \end{aligned}$$

- Del mismo modo

$$\begin{aligned} &\langle 010| H |010\rangle = \langle 010| (Z_1 Z_2 + Z_1 Z_3) |010\rangle \\ &= \langle 010| Z_1 Z_2 |010\rangle + \langle 010| (Z_1 Z_3) |010\rangle = -1 + 1 = 0 \end{aligned}$$

# El maravilloso mundo de los hamiltonianos

- Entonces, lo que nos interesa es hallar un estado cuántico $|x\rangle$ de forma que

$$\langle x| H |x\rangle$$

  sea mínimo, con $H = \sum_{(i,j)\in E} Z_i Z_j$ la función de coste del problema del corte máximo
- Se trata de un caso particular de un problema muy importante en física: hallar el estado de energía mínima (**ground state**) de un hamiltoniano
- Un hamiltoniano es una matriz $H$ hermitiana ($H = H^\dagger$)
- Físicamente, puede representar fuerzas, potenciales... en la ecuación de Schrödinger
- La energía de un estado $|\psi\rangle$ es

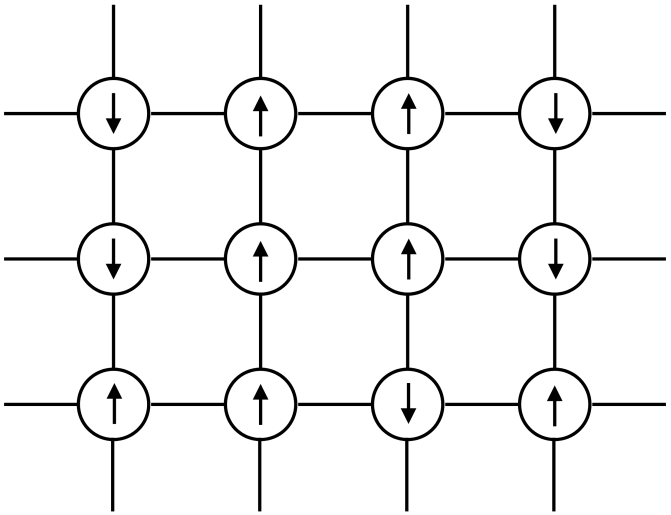
$$\langle\psi|H|\psi\rangle$$

# Ejemplo: el modelo de Ising

- Se tienen $n$ partículas con spin, que interactúan entre sí con ciertas fuerzas de acoplamiento
- Su hamiltoniano es

$$H = \sum_{1 \leq i < j \leq n} J_{ij} Z_i Z_j + \sum_{i=1}^{n} h_i Z_i$$

  con $J_{ij}$ y $h_i$ coeficientes reales
- Queremos encontrar una asignación de valores de spins (1 o -1) que minimice la suma
- El problema general es NP-hard



# QUBO: Quadratic Unconstrained Binary Optimization

- Una formulación alternativa del modelo de Ising son los problemas QUBO (Quadratic Unconstrained Binary Optimization)
- Se plantean como

$$\text{Minimizar} \sum_{1 \leq i \leq j \leq n}^{n} w_{ij} x_i x_j$$

donde cada $x_i$ es una variable binaria y los $w_{ij}$ son coeficientes reales

- Se puede reescribir como un modelo de Ising con la transformación

$$x_i = \frac{z_i + 1}{2}$$

y volver a QUBO con

$$z_i = 2x_i - 1$$

# Computación cuántica adiabática

- ¿Cómo obtener el *ground state* de $H$?
- Una solución natural es aplicar el propio hamiltoniano $H$ para llegar a la solución
- El **teorema adiabático** nos asegura que si comenzamos en el estado de mínima energía de un hamiltoniano y lo vamos variando lentamente, nos mantendremos siempre en el estado de mínima energía
- La idea de la computación cuántica adiabática es:
  - Comenzar en el estado de mínima energía de un hamiltoniano sencillo $H_i$
  - Evolucionar el sistema hacia el estado de mínima energía del hamiltoniano del problema $H_f$
  - Para ello se aplica el hamiltoniano dependiente del tiempo

$$H(t) = (1 - \frac{t}{T})H_i + \frac{t}{T}H_f$$

durante tiempo $T$

# Computación cuántica adiabática (2)

- Para garantizar la adiabaticidad, $T$ debe crecer como el inverso del cuadrado del *spectral gap* de H(t) (diferencia entre el primer y segundo nivel de energía)
- El spectral gap es **difícil** de calcular
- En la práctica, se usa el *quantum annealing*:
  - Se toma $H_i = -\sum_{i=1}^{n} X_i$ (con ground state $\sum_{x=0}^{2^n-1} |x\rangle$)
  - Como $H_f$ se toma un hamiltoniano de Ising
  - Se deja evolucionar durante un tiempo $T$ (no necesariamente adiabático)
  - Se mide para obtener una solución
  - Se repite un cierto número de veces y se devuelve la mejor solución obtenida
- Es la base de los ordenadores cuánticos de D-Wave

# Los ordenadores cuánticos de D-Wave

- Son ordenadores de propósito específico: resolver el modelo de Ising
- Accesibles gratuitamente (1 minuto/mes) a través de `https://www.dwavesys.com/take-leap`

# Quantum Approximate Optimization Algorithm (QAOA)

- El QAOA está inspirado en el modelo adiabático, pero para el paradigma de circuitos cuánticos
- El hamiltoniano adiabático es $H(t) = (1 - \frac{t}{T})H_i + \frac{t}{T}H_f$
- En la resolución de la ecuación de Schrödinger aparecen expresiones de la forma

$$e^{-i\alpha H(t)}$$

- En este caso, aproximamos la solución por

$$|\beta, \gamma\rangle = e^{-i\beta_p H_i} e^{-i\gamma_p H_f} \dots e^{-i\beta_2 H_i} e^{-i\gamma_2 H_f} e^{-i\beta_1 H_i} e^{-i\gamma_1 H_f} |s\rangle$$

donde

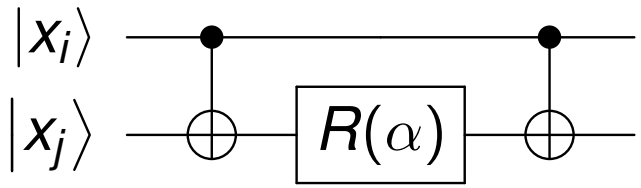
$$|s\rangle = \sum_{i=0}^{2^n-1} |x\rangle$$

# Optimización con QAOA

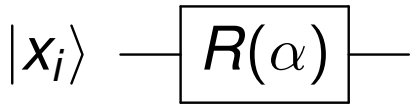
- Se trata de un método híbrido en el que intervienen un ordenador clásico y uno cuántico
- Sus pasos son:
  1. Elegir un valor $p$ y unos ángulos iniciales $\beta, \gamma$
  2. Preparar el estado $|\beta, \gamma\rangle$
  3. Estimar la energía $E(\beta, \gamma)$ de $|\beta, \gamma\rangle$ con respecto al hamiltoniano $H_f$
  4. Variar los parámetros $\beta$ y $\gamma$ para minimizar $E(\beta, \gamma)$
- El paso 2 se hace en el ordenador cuántico y los pasos 1, 3 y 4, en uno clásico

# Cómo preparar el estado $|\beta, \gamma\rangle$

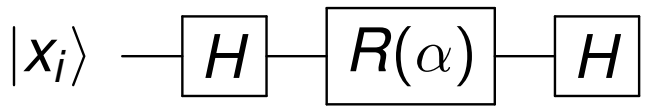
- El estado $|s\rangle = \sum_{i=0}^{2^n-1} |x\rangle$ se prepara fácilmente con puertas de Hadamard
- Cada $e^{-i\beta_k H_i}$ y $e^{-i\gamma_k H_f}$ se consigue con rotaciones y puertas CNOT y de Hadamard
- Para $e^{-i\alpha Z_i Z_j}$




- Para $e^{-i\alpha Z_i}$




- Para $e^{-i\alpha X_i}$



# Cómo estimar la energía

- Nos reduciremos al caso en el que tenemos un hamiltoniano tipo Ising

$$H_f = \sum_{i,j=1}^{n} J_{ij} Z_i Z_j + \sum_{i=1}^{n} h_i Z_i$$

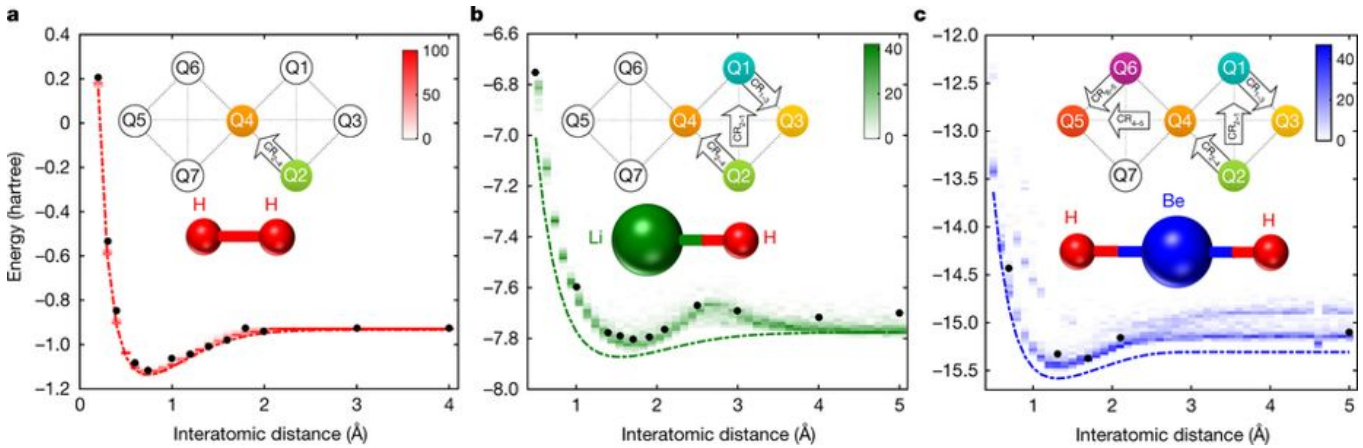
- Los pasos son:
  - Medimos el estado preparado $|\beta, \gamma\rangle$
  - Calculamos la energía de la secuencia de bits obtenida
    - Cada uno de los términos $Z_i Z_j$ y $Z_i$ sólo puede ser 1 o -1
    - $Z_i Z_j$ sólo depende de los bits de las posiciones $i$ y $j$. Será 1 si son iguales y -1 si son distintos.
    - $Z_i$ sólo depende del bit de la posición $i$. Será 1 si el bit es 0 y -1 si el bit es 1.
  - Repetimos un cierto número de veces y promediamos
- Es interesante guardar el valor mínimo de energía de entre los valores medidos

# Propiedades del QAOA

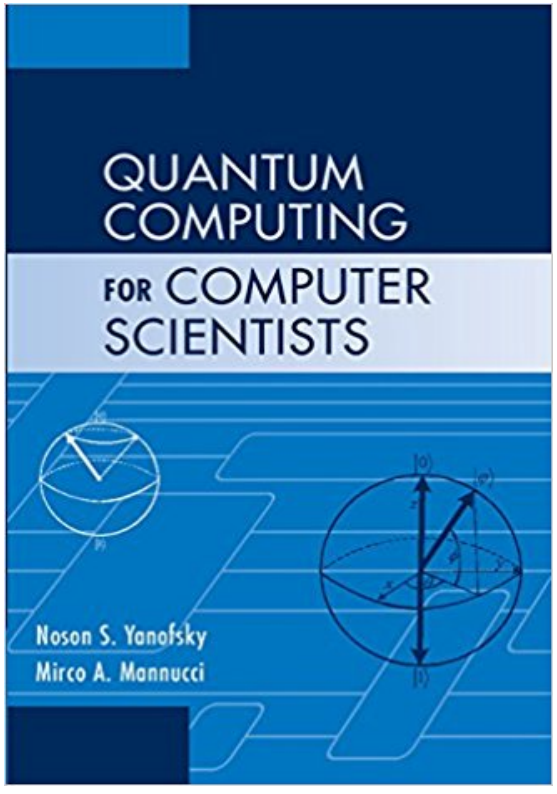
- Los circuitos del QAOA tienen un número de puertas polinomial en $p$ si $H_f$ tiene un número polinomial de sumandos
- Existen resultados teóricos sobre el ratio de aproximación del QAOA en algunos problemas (corte máximo)
- Hay dependencia de $p$ (y otros factores) en la bondad de la aproximación
- El método se puede extender a otros problemas (factorización, Grover...)

# VQE: Variational Quantum Eigensolver

- El QAOA es un caso particular de un algoritmo más general: el VQE (Variational Quantum Eigensolver)
- En lugar de $|\beta, \gamma\rangle$ se usa un estado (ansatz) que también depende de parámetros y en el que utilizamos conocimiento del problema
- Estos métodos se han usado, por ejemplo, para obtener energías de enlace de algunas moléculas



# Para saber más




- Quantum Computing for Computer Scientists, Noson Yanofsky y Mirco Mannucci
- Quantum Computing Lecture Notes, John Watrous (PDF)
- Quantum Computation and Information at CMU, Ryan O'Donnell (Vídeos)