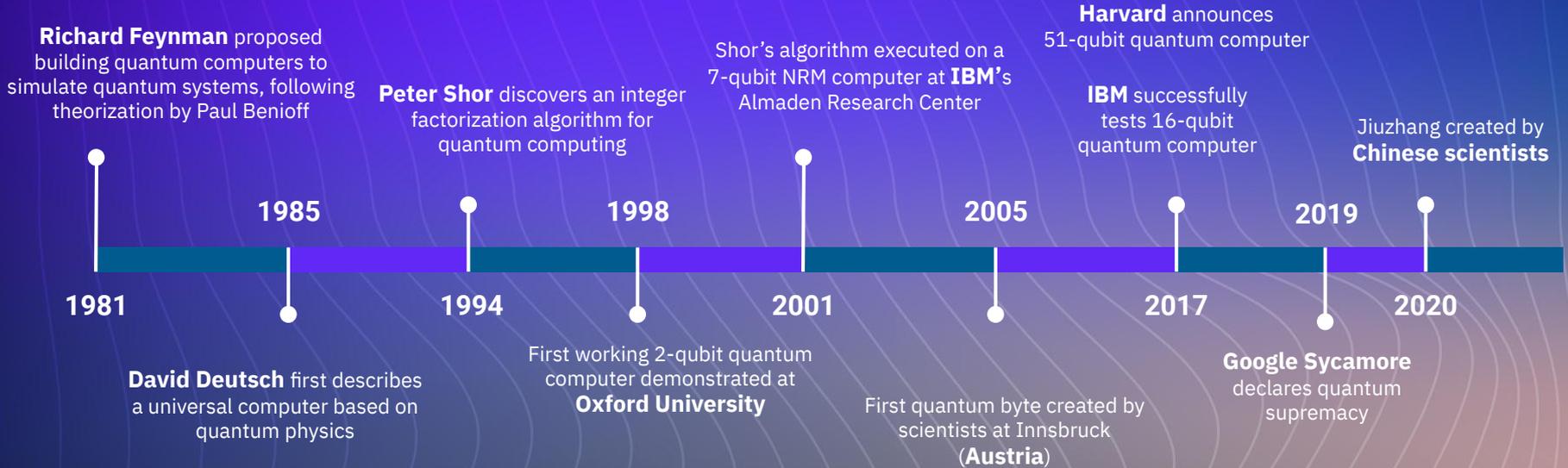


# Quantum Computing

Il contributo del QC nello sviluppo delle applicazioni di Artificial Intelligence

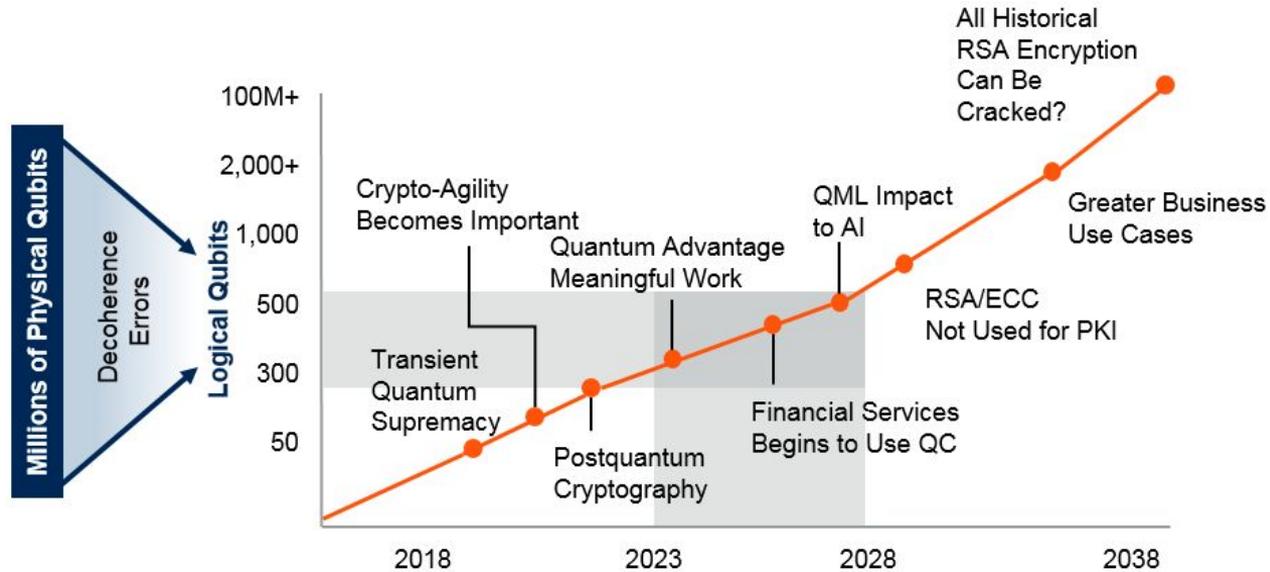


# La storia ...



# ... e le prospettive

## Qubit Timeline Estimates

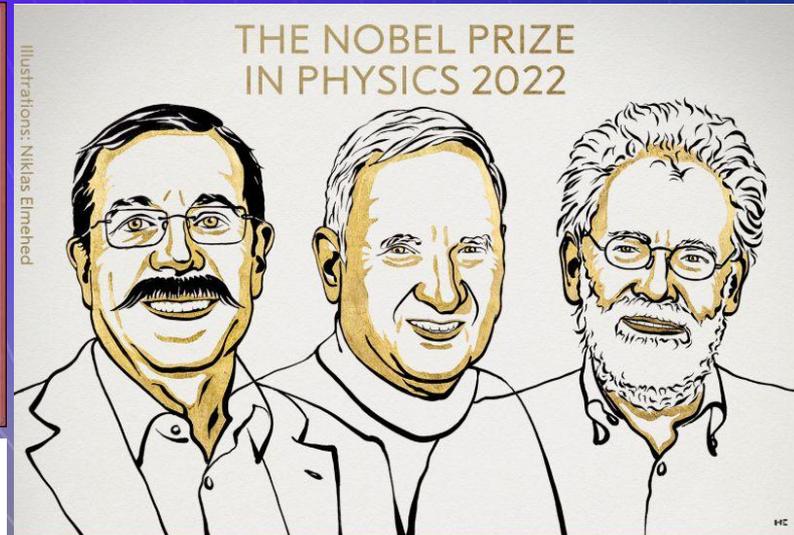
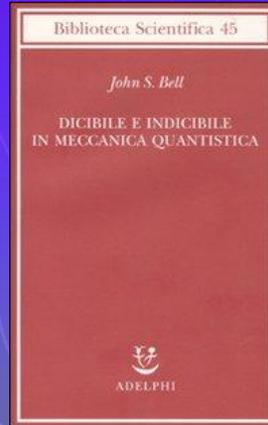


Note: Dates are speculative  
ID: 374252

© 2018 Gartner, Inc.

# Una terrificante azione a distanza...

Le misure sperimentali sulle diseguaglianze derivate da **J. Bell** nel 1964 (A. Aspect, 1982) dimostrano la non località della MQ



234

*Speakable and unspeakable in quantum mechanics*

The situation is further complicated by the fact that there *are* things which *do* go faster than light. British sovereignty is the classical example. When the Queen dies in London (may it long be delayed) the Prince of Wales, lecturing on modern architecture in Australia, becomes *instantaneously* King. (Greenwich Mean Time rules here.) And there are

Alain  
Aspect

John F.  
Clauser

Anton  
Zeilinger

"for experiments with entangled photons, establishing the violation of Bell inequalities and pioneering quantum information science"

THE ROYAL SWEDISH ACADEMY OF SCIENCES

# Sommario

L'unità di base di informazione è il **qubit**

Un qubit usa le proprietà della MQ per fornire un'unità che può essere un qualsiasi **valore intermedio tra 0 e 1**

Le regole della MQ includono meccanismi non presenti nel mondo classico, come **sovrapposizione, interferenza ed entanglement**

Progettando algoritmi che sfruttano questi effetti, si possono affrontare problemi che altrimenti richiederebbero risorse esorbitanti

BIT



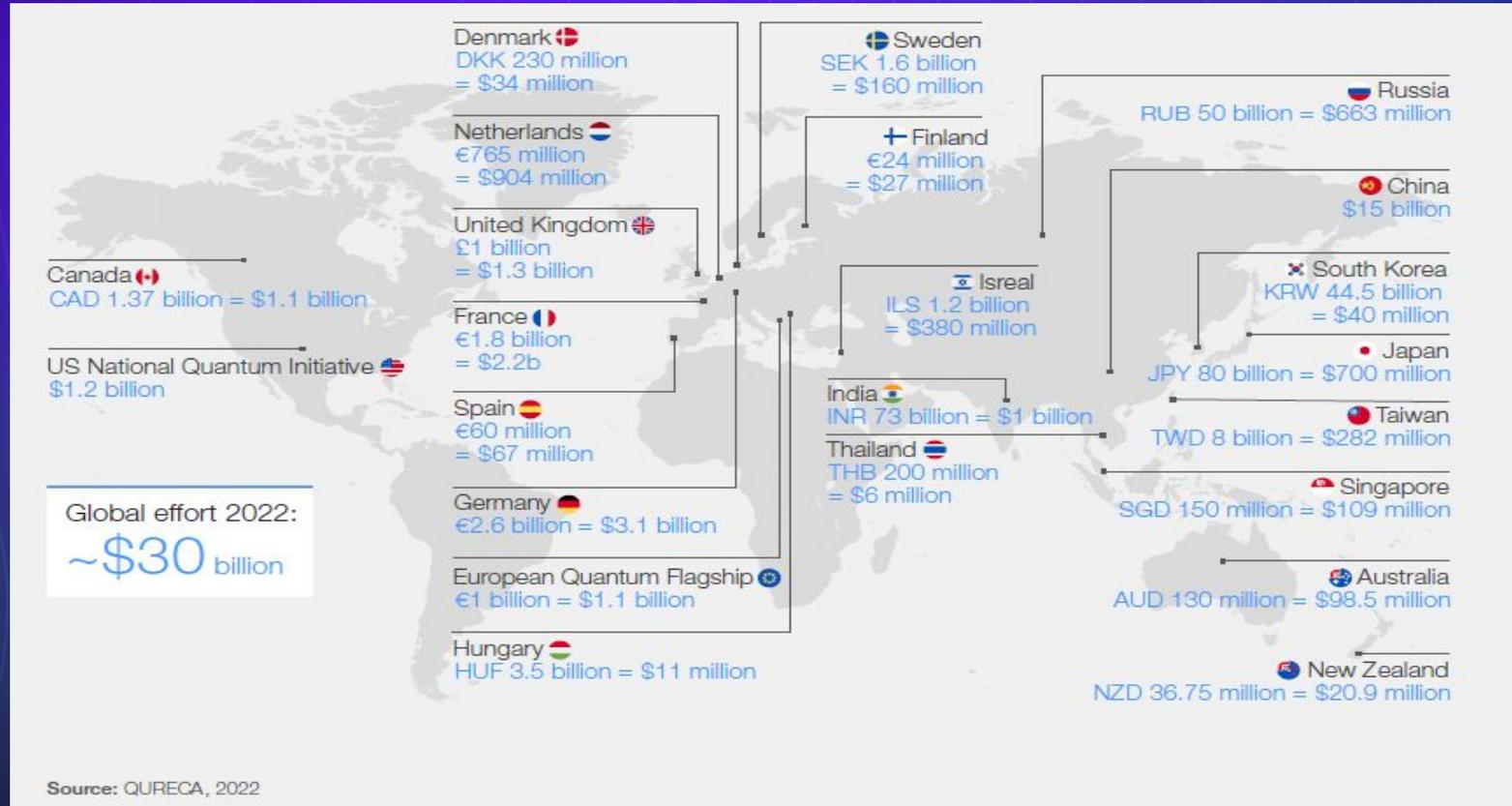
QUBIT



1

1

# Investimenti sulle quantum tech

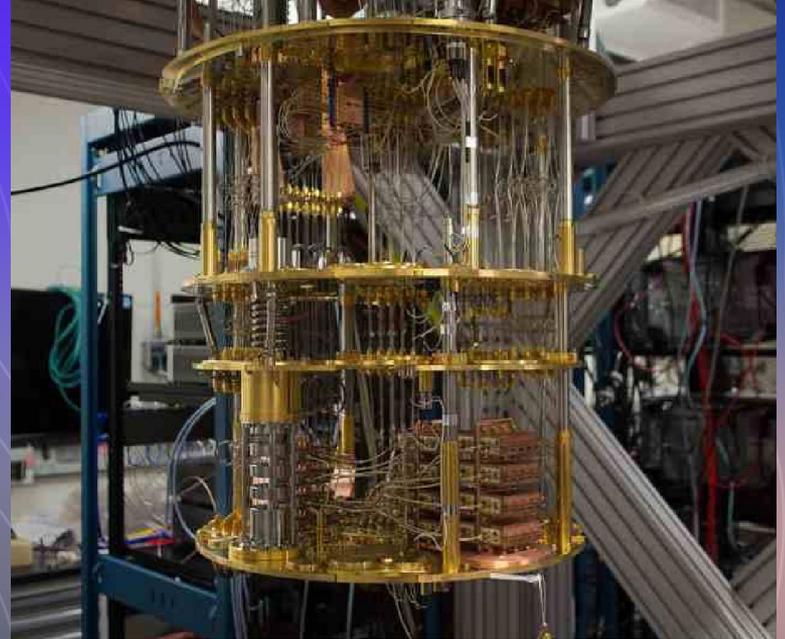


# Il potenziale delle quantum tech

Domini applicativi	Industria	Sistemi complessi	HiTech
Settore	Energia, agricoltura, food, medicina, chimica	Finanza, trasporti e logistica	Digicomm, scienze, difesa e sicurezza
Casi d'uso	Progettazione di nuove molecole, materiali, farmaci	Ottimizzazione di sistemi con un grande numero di variabili	Impatto sulle tecnologie esistenti come AI e blockchain
Impatto su ambiente e società	Materiali più efficienti, colture più rispettose della natura, medicina di precisione	Riduzione del consumo di energia, modelli commerciali circolari	Crittografia più forte, privacy, accelerazione della ricerca di base
Esempi	Molecole per cattura CO <sub>2</sub> , grani più resistenti	Ottimizzare lo spazio vuoto nei container. Routing di veicoli	Accelerazione del training nel ML, codifica criptovalute

# Hardware disponibile

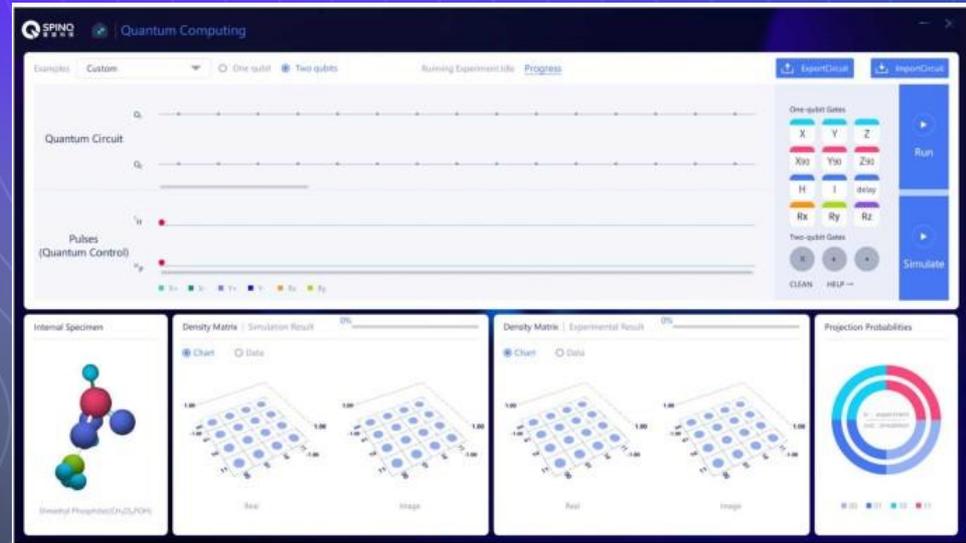
Company	Modello	Numero qbit
IBM	System One	20
	Eagle	128
Google	Sycamore	53
	Bristecome	72
Rigetti	Forest-1	36-80
IonQ	Aria	20
USTC	Jiuzhang	76-113
D-Wave	Two	512*
	2000Q	2048*
	Advantage	5640*



Alcuni QC sono accessibili in cloud con connessioni a consumo

# Novità dalla Cina ...

SpinQ commercializza Triangulum, un QC desktop con 3 qubit ad uso didattico (!)



# Requisiti per un QC

Secondo i **Divincenzo's criteria**, un QC dovrebbe disporre di:

- Un numero scalabile di qubit logicamente distinti
- Capacità di inizializzarli in uno stato puro
- Tempi di decoerenza più lunghi dei tempi di calcolo
- Un set universale di gate di manipolazione
- Una funzione di osservazione dello stato
- Capacità di correggere errori

Oggi nessun QC esistente li soddisfa tutti

**NISQ**: Noise Intermediate Scale Quantum



# Il Qubit

Lo stato di un qubit viene rappresentato come sovrapposizione di due stati di base, rappresentati come vettori colonna o come **ket** di Dirac

$$\begin{aligned} |0\rangle &\Leftrightarrow \begin{pmatrix} 1 \\ 0 \end{pmatrix} & |1\rangle &\Leftrightarrow \begin{pmatrix} 0 \\ 1 \end{pmatrix} \\ \alpha|0\rangle + \beta|1\rangle &\Leftrightarrow \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \end{aligned}$$

Le quantità  $\alpha$  e  $\beta$  sono numeri complessi

Lo stato è normalizzato, cioè  $|\alpha|^2 + |\beta|^2 = 1$

Lo stato non è **osservabile!** Una misura produce  $|0\rangle$  con probabilità  $|\alpha|^2$

# La sfera di Bloch

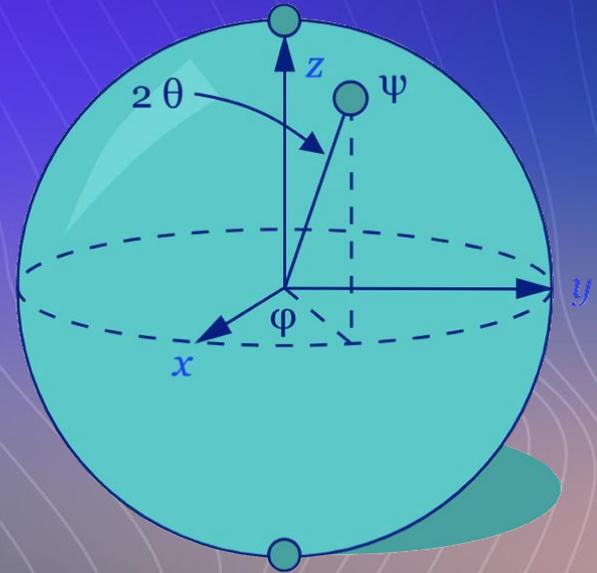
Lo stato di un qubit si può interpretare come un punto sulla superficie di una sfera 3D, definito da due angoli (lat/lon)

Gli stati base sono **antipodali** (poli)

Lo stato è definito a meno di un fattore di fase, cioè  $|\psi\rangle$  e  $\exp(i\varphi) |\psi\rangle$  sono lo stesso stato fisico

Gli assi XYZ permettono di definire le **rotazioni** dello stato (operatori unitari)

Struttura algebrica: spazio di **Hilbert**



# Operazione su un qubit

Le operazioni (**gate**) su un qubit sono in generale rotazioni sulla sfera di Bloch.  
Le tre operazioni base sono codificate dalle **matrici di Pauli**

$$\begin{array}{l} |0\rangle \Rightarrow |1\rangle \\ |1\rangle \Rightarrow |0\rangle \end{array} \quad \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

$$\begin{array}{l} |0\rangle \Rightarrow +i|1\rangle \\ |1\rangle \Rightarrow -i|0\rangle \end{array} \quad \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$$

$$\begin{array}{l} |0\rangle \Rightarrow +|0\rangle \\ |1\rangle \Rightarrow -|1\rangle \end{array} \quad \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$



# Hadamard gate

La rotazione più importante converte gli stati di base in sovrapposizioni  
(**preparazione dello stato**)

$$H = \frac{1}{\sqrt{2}}(X + Z) = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

$$|0\rangle \Rightarrow \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$

$$|1\rangle \Rightarrow \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$



# Sistemi a N qubit

Lo stato di un sistema a N qubit viene generalizzato come spazio di Hilbert a  $2^N$  dimensioni (tramite prodotto tensoriale)

Per 2 qubit

$$|\psi\rangle = \alpha|00\rangle + \beta|01\rangle + \gamma|10\rangle + \delta|11\rangle$$

$$|0\rangle \otimes |0\rangle = |00\rangle$$

$$|0\rangle \otimes |1\rangle = |01\rangle$$

$$|1\rangle \otimes |0\rangle = |10\rangle$$

$$|1\rangle \otimes |1\rangle = |11\rangle$$

oppure (notazione compatta)

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle + \gamma|2\rangle + \delta|3\rangle$$

$$|00\rangle \Leftrightarrow \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad |01\rangle \Leftrightarrow \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

$$|10\rangle \Leftrightarrow \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} \quad |11\rangle \Leftrightarrow \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

In generale

$$|\psi\rangle = \sum_{k=0}^{2^N-1} \alpha_k |k\rangle$$

# Gate controllati

Operando con 2 qubit si può impiegare uno per controllare l'azione di un generico operatore unitario  $U$  sull'altro

$$CU = \begin{Bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & U & 0 \\ 0 & 0 & 0 & 0 \end{Bmatrix}$$



Esempio: C-NOT  $\Rightarrow$

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



# Stati prodotto e stati di Bell

Se gli stati dei singoli qubit sono fra loro indipendenti, lo stato globale può essere fattorizzato come **prodotto tensoriale** degli stati dei singoli qubit

Ma esistono stati per cui questo non è possibile [**stati di Bell**]

$$|\Phi_{\pm}\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \neq (\alpha|0\rangle + \beta|1\rangle) \otimes (\gamma|0\rangle + \delta|1\rangle)$$

Significa che i qubit sono tra loro connessi (**entanglement**) ed il loro stato esiste solo in modo congiunto. L'osservazione di un singolo qubit ha effetto **immediato** sull'altro

# Operatori generici

Un generico operatore su N qubit si può sempre scrivere utilizzando solo operatori unari ed **almeno** un operatore controllato

Esempio: produrre lo stato di Bell di 2 qubit

$$CNOT((H_2 \otimes |1\rangle_1)(|0\rangle_2 \otimes |0\rangle_1))$$

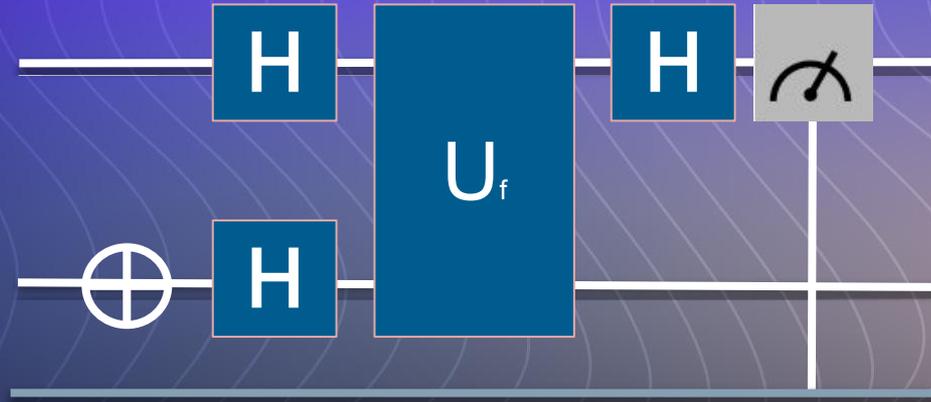
$$CNOT\left(\frac{1}{\sqrt{2}}|0\rangle_2 \otimes |0\rangle_1 + \frac{1}{\sqrt{2}}|1\rangle_2 \otimes |0\rangle_1\right)$$

$$\frac{1}{\sqrt{2}}(|0\rangle_2 \otimes |0\rangle_1 + |1\rangle_2 \otimes |1\rangle_1)$$



# Quantum circuit

Il generico programma di QC prepara N qubit in un certo stato e li fa «evolvere» tramite un opportuno insieme di operatori [**Gate**] verso uno stato finale su cui viene fatta una **misura**



# Computazione quantistica

La computazione quantistica deve prima di tutto inglobare tutta la computazione classica

La computazione quantistica si basa su trasformazioni unitarie (**reversibili**) mentre quella classica è di norma **irreversibile**

Occorre quindi disporre di un metodo standard per trasformare funzioni irreversibili in funzioni reversibili (**biunivoche**)

Esempio: data la funzione  $f : \{0, 1\}^k \mapsto \{0, 1\}^m$

si costruisce una nuova funzione  $\tilde{f} : \{0, 1\}^{k+m} \mapsto \{0, 1\}^{k+m}$

che agendo su  $(x, 0^m)$  produce  $(x, f(x))$

# Parallelismo quantistico

Disponendo di  $N$  qubit si possono generare stati in sovrapposizione a partire da  $2^N$  stati di base

È quindi possibile calcolare il valore di una funzione  $f$  (cioè di un operatore unitario  $U$ ) su tutti questi stati contemporaneamente

Ma ...

... lo stato in sovrapposizione non è **osservabile**,

... la misura è probabilistica e soprattutto **distruttiva**

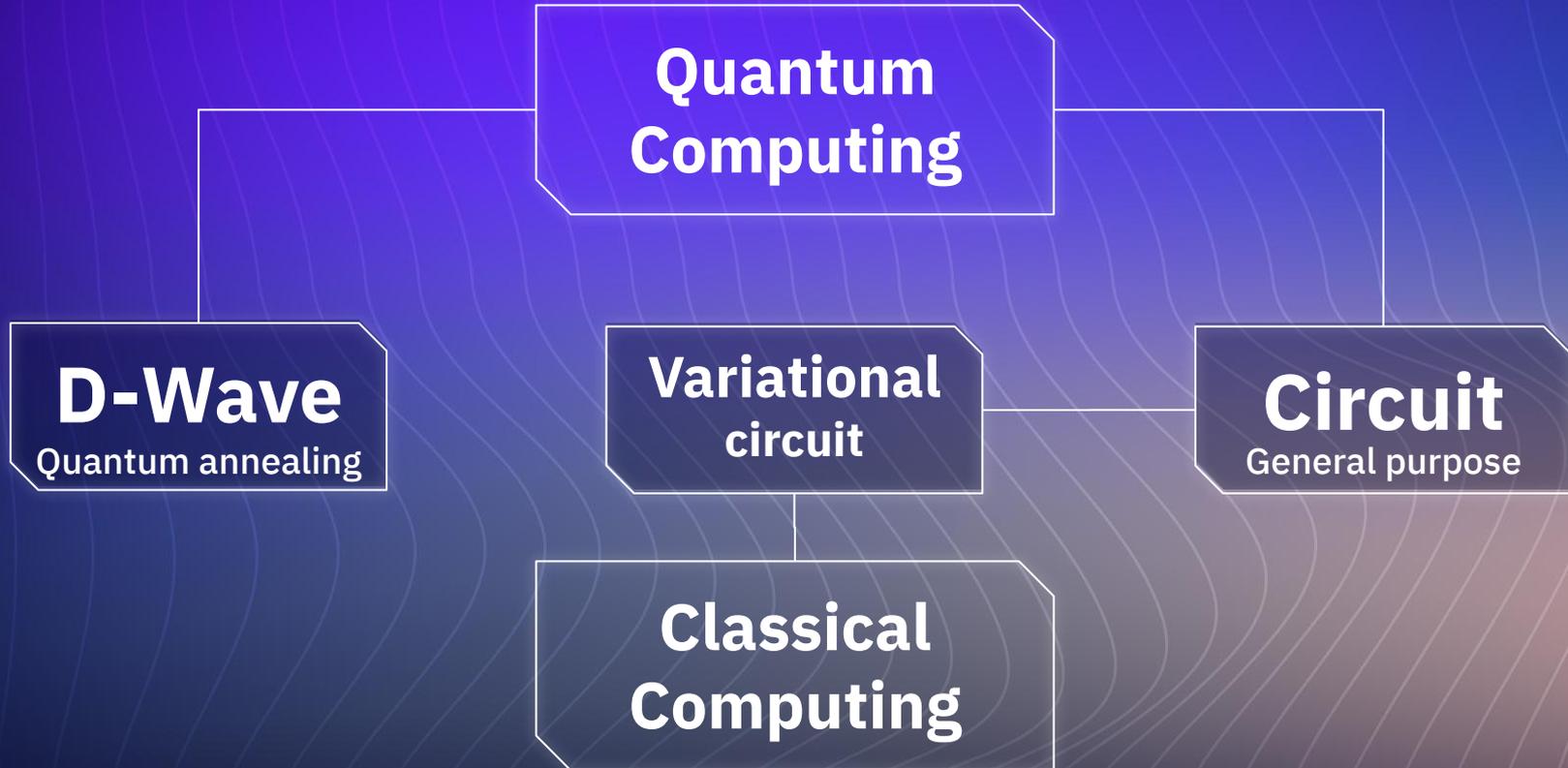
Occorrono diverse esecuzioni (shot) per ridurre l'errore sotto una soglia fissa

# Materiale utile online

Materiale utile online per lo studio e sperimentazione degli argomenti

- <https://quantumalgorithmzoo.org>
- <https://qiskit.org/documentation>
- <https://quantum-computing.ibm.com/composer>
- <https://docs.ocean.dwavesys.com/en/stable/overview/install.html>
- <https://pyqubo.readthedocs.io>

# Framework Quantistici [era NISQ]



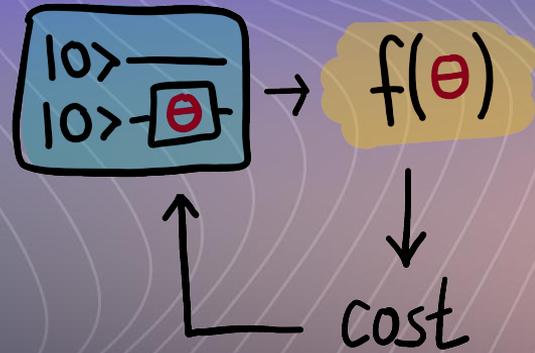
# Framework Quantistici

## NISQ: Noisy Intermediate-Scale Quantum era

Attualmente ci troviamo in un'epoca, dove i computer quantistici hanno scarse risorse e problemi di correzione degli errori.

Diventano preferibili soluzioni di dominio specifiche o ibride con i computer classici

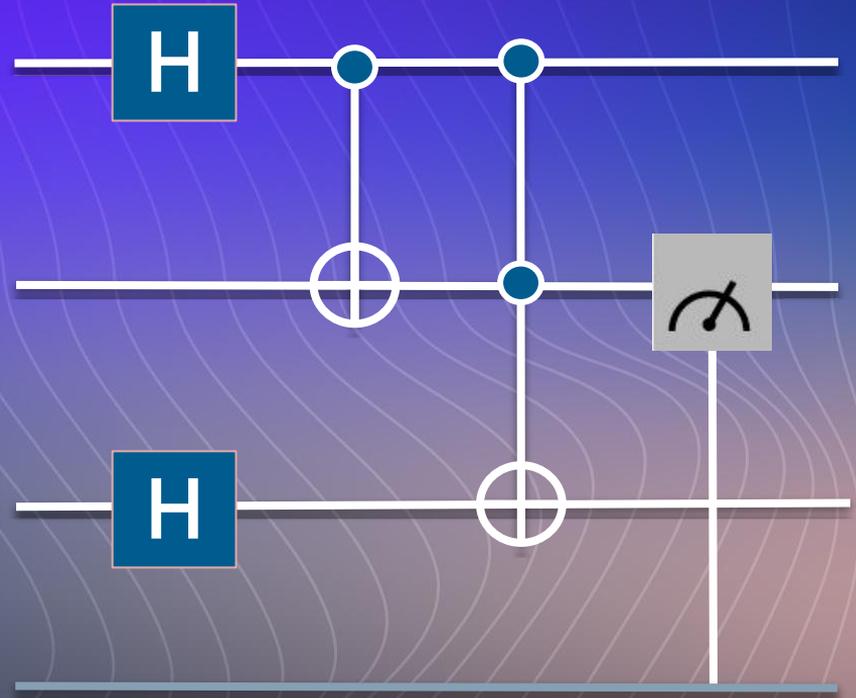
**D:wave**  
The Quantum Computing Company™



# Circuiti Quantistici

Ogni linea rappresenta la vita di un qubit durante l'esecuzione del programma

Su queste linee vengono posizionati i gate con cui i qubit vengono manipolati



# Circuiti Quantistici (gate unitari)

## GATE UNARI



NOT

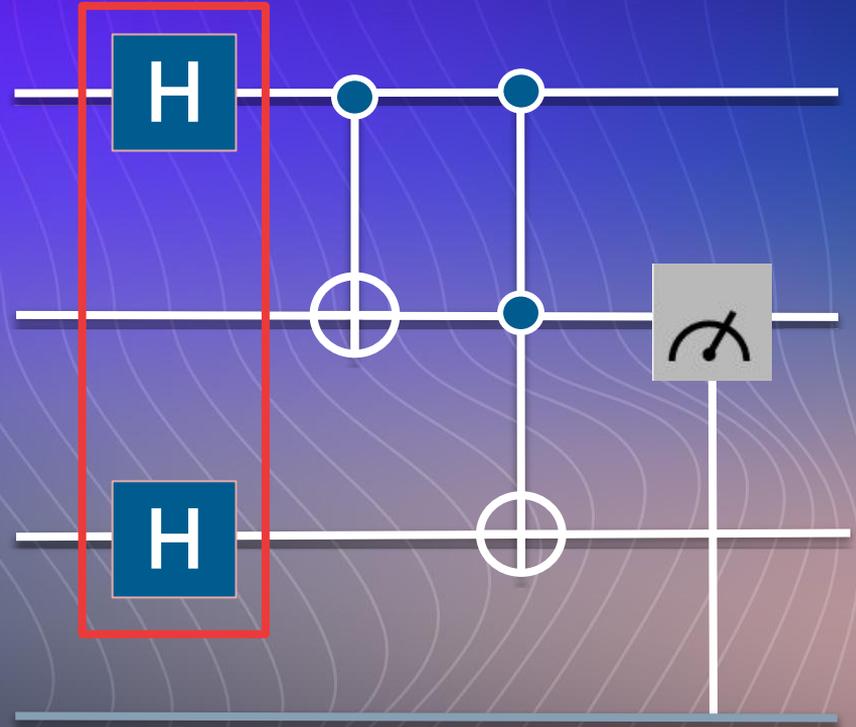
$$\begin{Bmatrix} 0 & 1 \\ 1 & 0 \end{Bmatrix}$$



HADAMARD



ROTATION

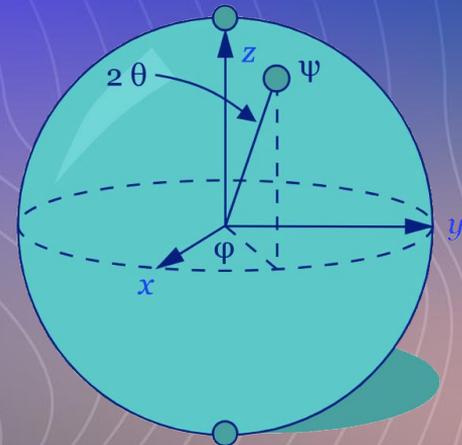
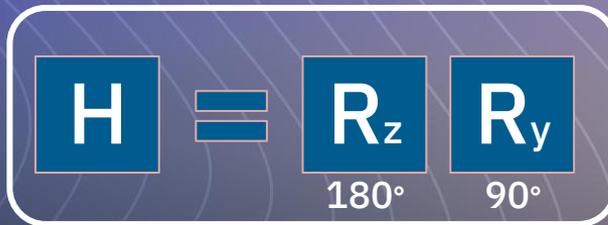
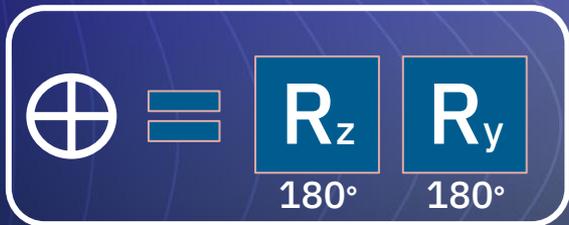


# Circuiti Quantistici (gate unitari)

Supportando un limitato set di gate è possibile ottenere qualunque altro gate unitario es.  $\{R_y, R_z\}$

A ogni qubit è associato un pacchetto informativo codificabile da un punto su una sfera

A ogni stato è associata una probabilità e una fase



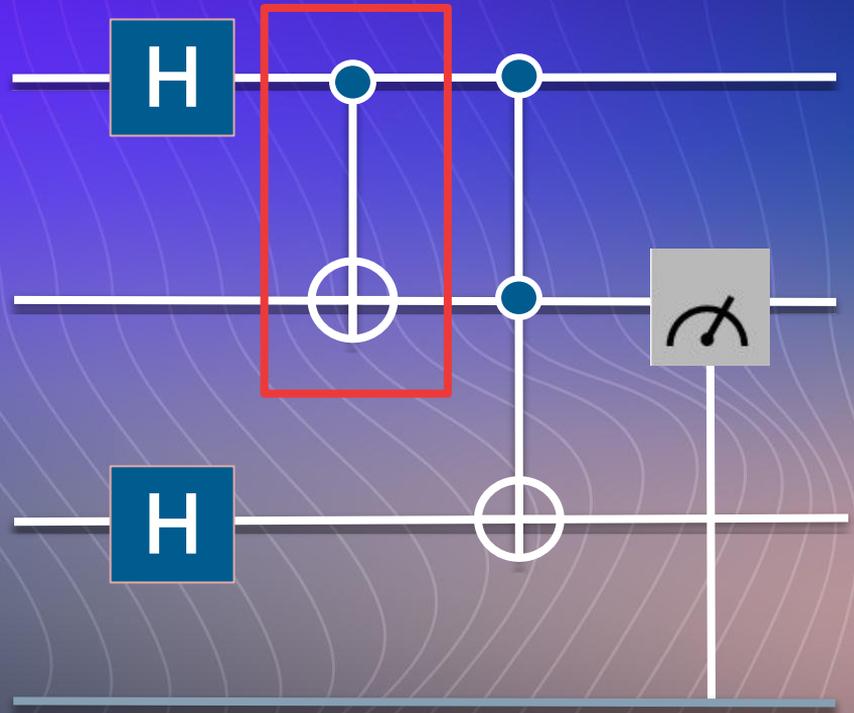
Sfera di Bloch

# Circuiti Quantistici (gate controlled)

La versione “controlled” di un gate permette di applicare un gate unario su un sottoinsieme di configurazioni dei due qubit

Questo permette di mettere due qubit in entanglement

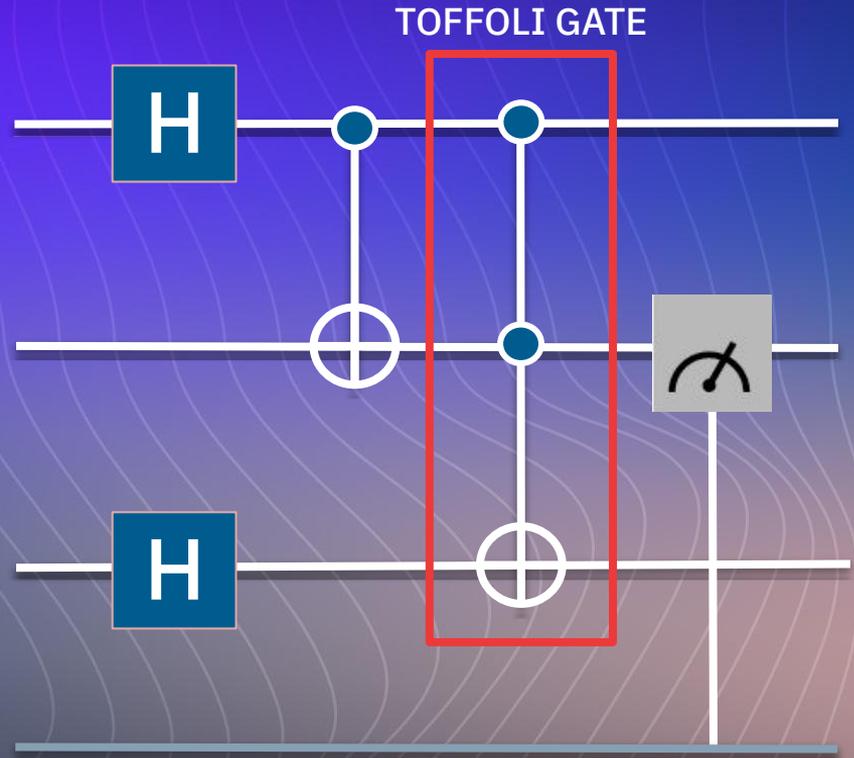
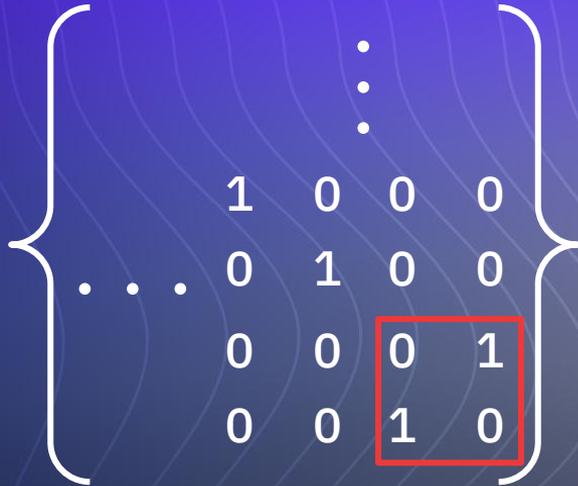
$$\left\{ \begin{array}{cccc} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{array} \right\}$$



# Circuiti Quantistici (gate controlled)

È utile utilizzare anche gate multi qubit

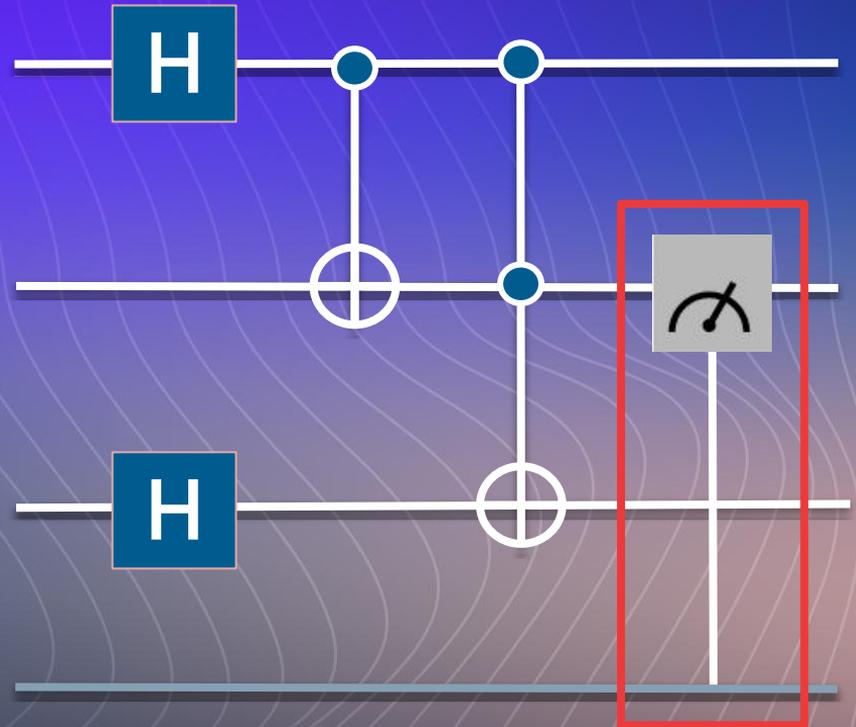
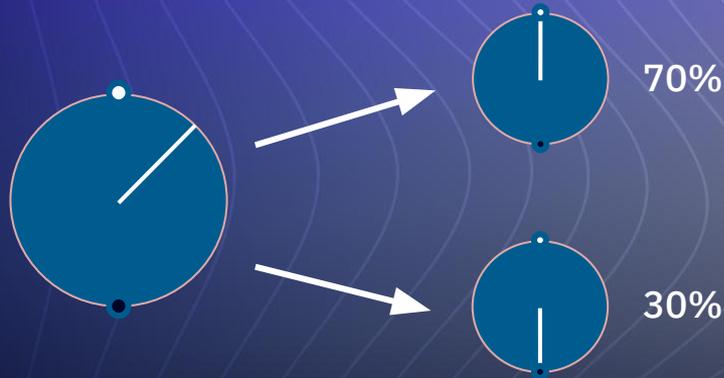
Questi gate sono sempre implementabili utilizzando solo gate unari o binari



# Circuiti Quantistici (gate di misura)

Una volta misurato un qubit questo si polarizza su uno stato certo (0 o 1)

È l'unica operazione che permette una perdita di informazione durante la computazione



# Circuiti Quantistici (riassunto elementi)

- Parallelismo intrinseco
  - Reversibile
  - Necessario considerare il sistema globalmente
  - Possibilità di computazione universale
- 
- Dati N qubit lo stato della macchina si può rappresentare con un vettore di  $2^N$  complessi
  - Per applicare un gate a uno stato base calcolare il prodotto matrice-vettore
  - Di conseguenza ogni programma quantistico è comprimibile in una matrice  $2^N \times 2^N$
  - Se  $N > 30$  diventa molto difficile simulare un programma quantistico su un computer classico

# Circuiti Quantistici

## Deutsch algorithm

### 1° Esempio: l'algoritmo di Deutsch

Data una funzione binaria  $F: \{0,1\} \rightarrow \{0,1\}$  voglio stabile se questa è una funzione **bilanciata** o **costante**

Questo problema giocattolo è utile per comprendere il parallelismo quantistico

Nella versione classica di questo algoritmo è necessario eseguire la funzione su ogni possibile input per determinare la tipologia della funzione

Nella versione quantistica vedremo che è possibile eseguire una sola inferenza a prescindere dalla dimensione dell'input

Generalizzabile su funzioni

$$F: 2^N \rightarrow \{0,1\}$$

**Algoritmo di Deutsch-Jozsa**

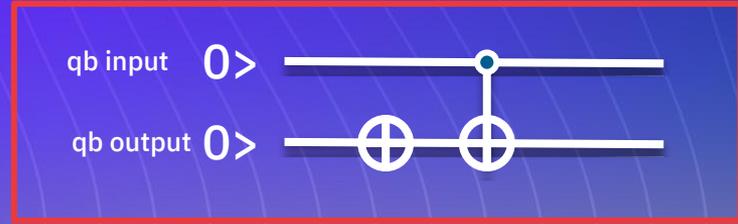
# Deutsch Algorithm

Funzione costante

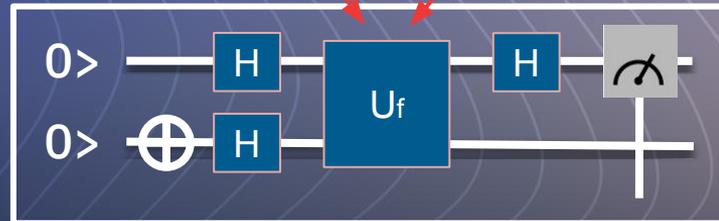


$0 \rightarrow 1$   
 $1 \rightarrow 1$

Funzione bilanciata



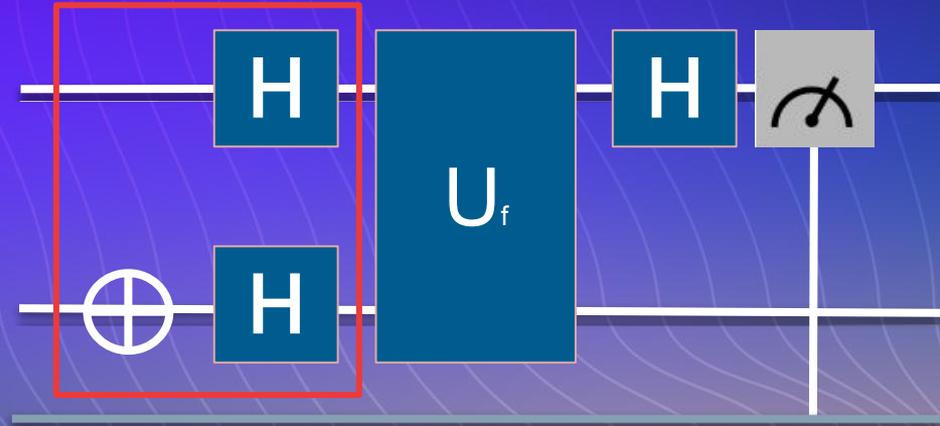
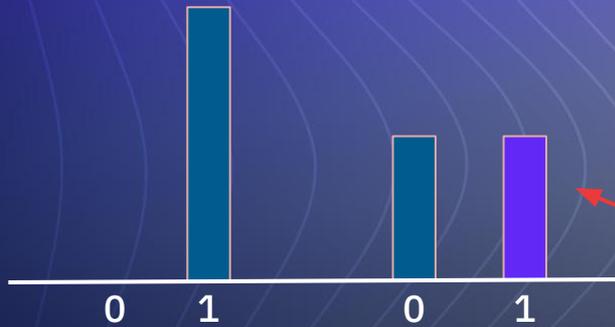
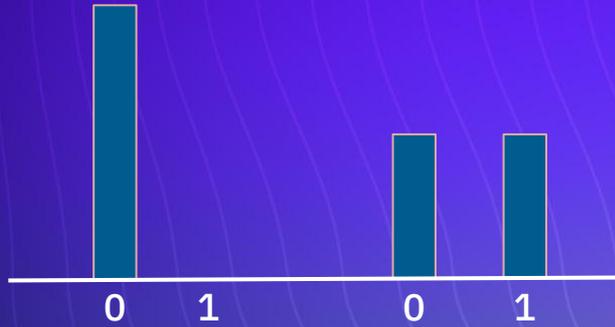
$0 \rightarrow 1$   
 $1 \rightarrow 0$



$0 \Rightarrow$  costante

$1 \Rightarrow$  bilanciata

# Deutsch Algorithm

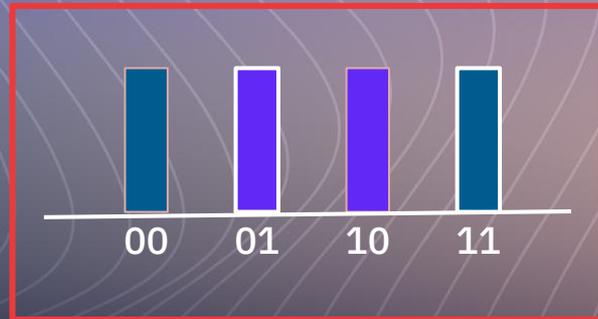
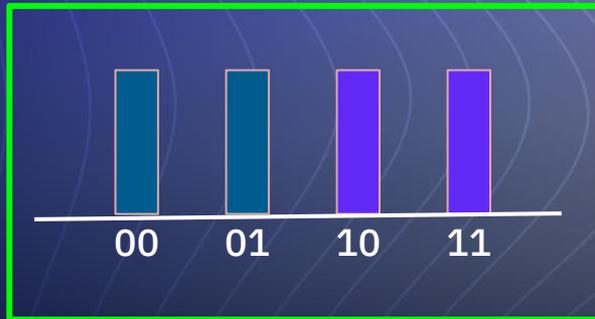
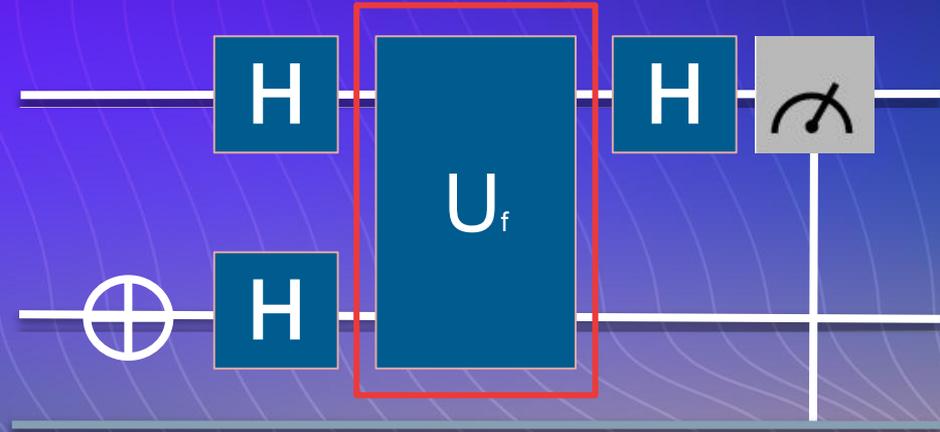


Hadamard applicato su un qubit con stato 1 ha l'effetto di negare la fase

# Deutsch Algorithm

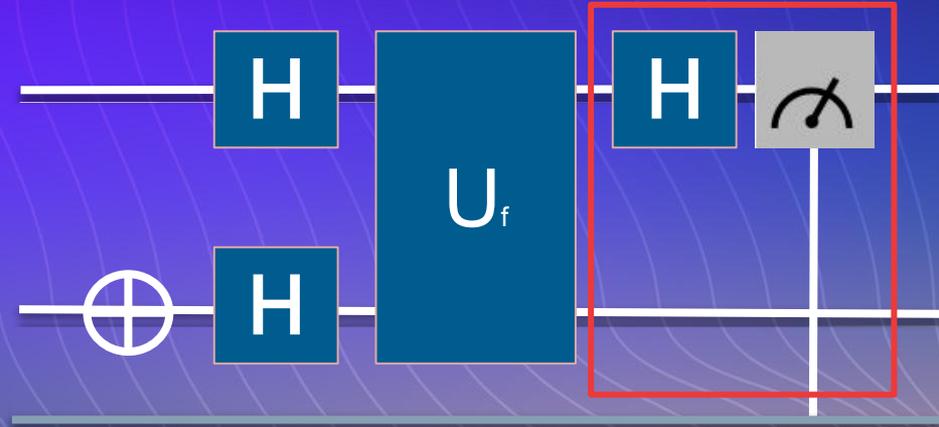
A questo punto con una singola inferenza la funzione viene testata su ogni possibile input

La funzione potenzialmente mette in entanglement i due qubit



# Deutsch Algorithm

- Caso **costante**:  
Il qubit di input non sarà stato condizionato dall'inferenza della funzione quindi riapplicando il gate H tornerà al valore 0 di partenza
- Caso **bilanciata**:  
Il qubit in input verrà influenzato dalla fase negativa del qubit in output cambiando il comportamento del gate H che porta il qubit di input a 1



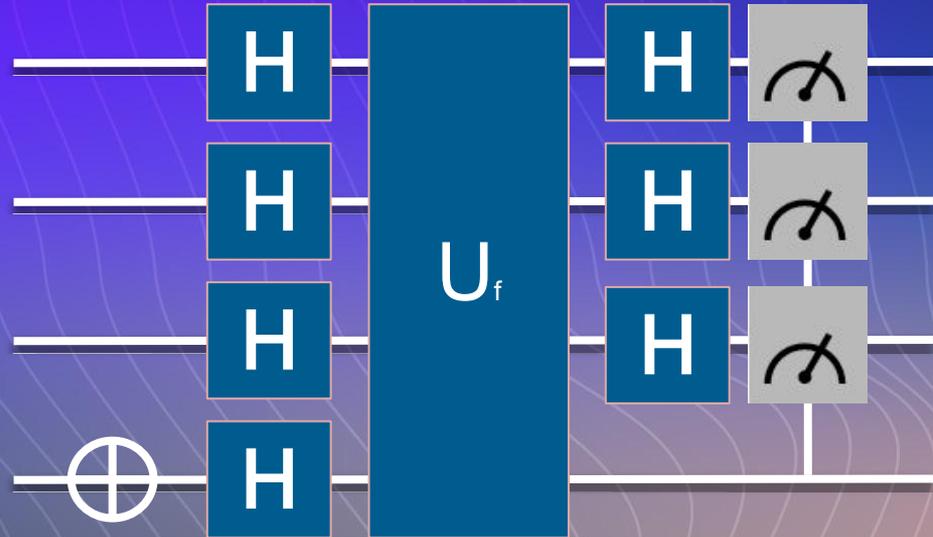
H

Hadamard è l'inversa di se stessa

# Deutsch Algorithm

conclusioni

- Esplorazione di diversi casi con una singola inferenza
- L'importanza dell'uso della componente di fase dei qubit durante la computazione



L' algoritmo è generalizzabile su funzioni binarie con input arbitrario, questo fa apprezzare maggiormente il parallelismo quantistico

# Circuiti Quantistici

## Grover algorithm

### 2° Esempio: l'algoritmo di Grover

Data una collezione di elementi (senza definire una relazione d'ordine), individuare gli elementi con determinate caratteristiche

Grover è un algoritmo di ricerca generico

Ogni stato quantistico rappresenta un elemento della nostra collezione

Dati  $N$  qubit quindi avremo  $2^N$  elementi.

Una ricerca con un computer classico avrebbe complessità esponenziale quindi...

Vedremo che con Grover lo stesso problema può essere risolto con uno speedup quadratico!

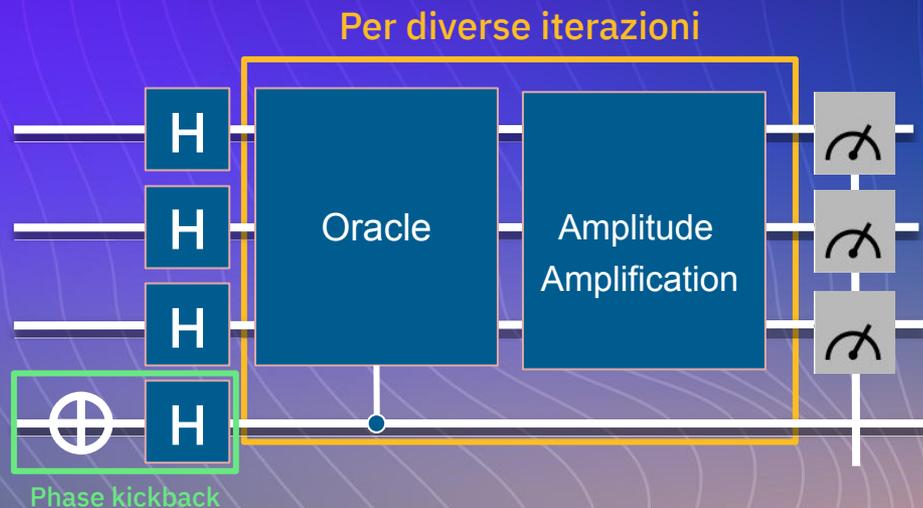
# Grover algorithm



La funzione oracolo serve a individuare l'elemento che stiamo cercando, segnandolo con fase negativa



L' Amplitude Amplification amplifica la probabilità legata allo stato segnato in precedenza

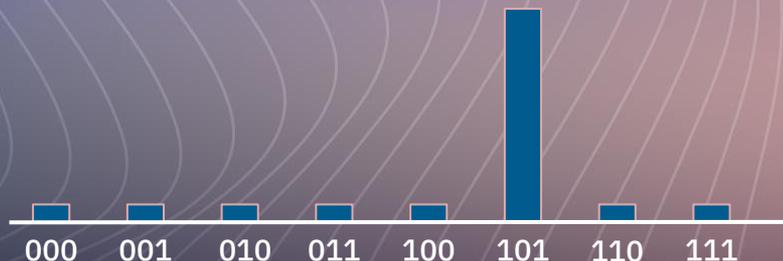
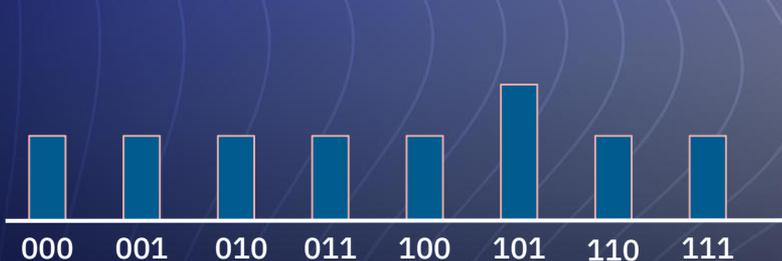
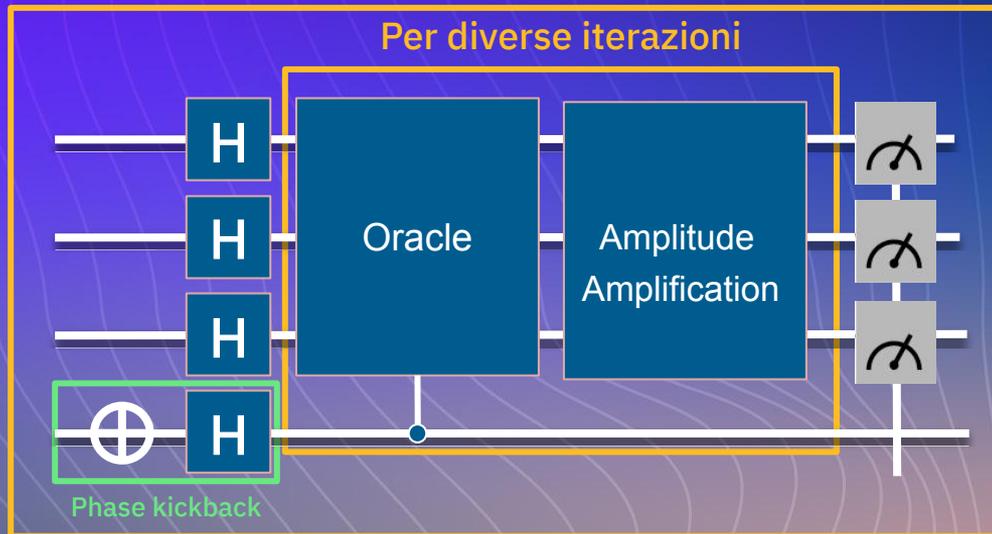


# Grover algorithm

Ripetendo il circuito oracle-amplification più volte le probabilità si polarizzano maggiormente sull'elemento cercato

Tuttavia la probabilità non arriverà mai a uno costringendo a ripetere il circuito più volte per campionare la distribuzione

Per diverse iterazione per ottenere la distribuzione



# Grover algorithm

Di quante iterazioni stiamo parlando?

Con N qubit in input e M soluzioni attese

$$\#iteration = \frac{\pi}{4} \sqrt{\frac{2^N}{M}}$$

Questo vuol dire che rispetto la controparte classica  
Grover ha uno speed up quadratico

# Grover algorithm

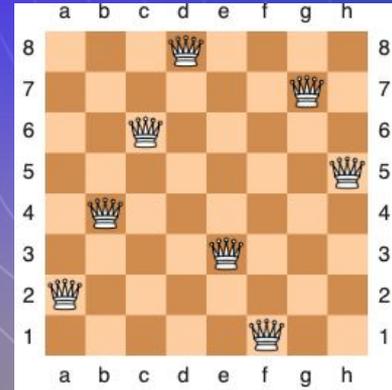
caso d'uso [problema delle N regine]

Lo scopo è posizionare N regine sulla scacchiera senza che si mangino a vicenda

Questo problema ha un'importanza teorica in informatica poiché è NP-completo

Non è possibile trovare una soluzione con certezza in meno di un numero esponenziale di passi

È possibile verificare se una scacchiera è soluzione in un tempo polinomiale



# Grover algorithm

caso d'uso [problema delle N regine]

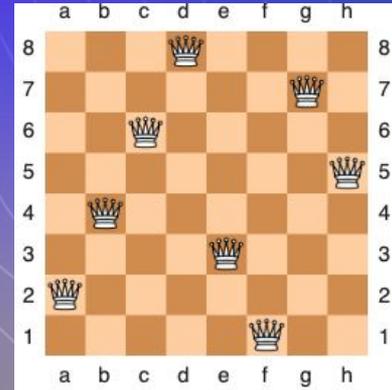
Lo scopo è posizionare N regine sulla scacchiera senza che si mangino a vicenda

Questo problema ha un'importanza teorica in informatica poiché è NP-completo

Non è possibile trovare una soluzione con certezza in meno di un numero esponenziale di passi

È possibile verificare se una scacchiera è soluzione in un tempo polinomiale

È possibile risolvere questo problema con Grover!



# Grover algorithm

caso d'uso [problema delle N regine]

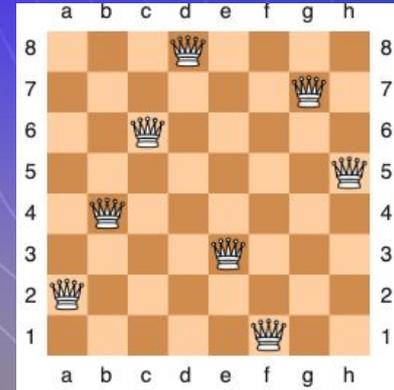
Proviamo a risolvere il problema per  $N = 4$

Possiamo rappresentare la scacchiera con 16 qubit  
Codificando con ogni stato quantistico una possibile  
scacchiera [ $2^{16}$  soluzioni candidate]

```
0000000000000000  
0000000000000001  
0000000000000010  
0000000000000011  
0000000000000100  
0000000000000101  
0000000000000110  
0000000000000111  
0000000000001000  
0000000000001001
```

⋮

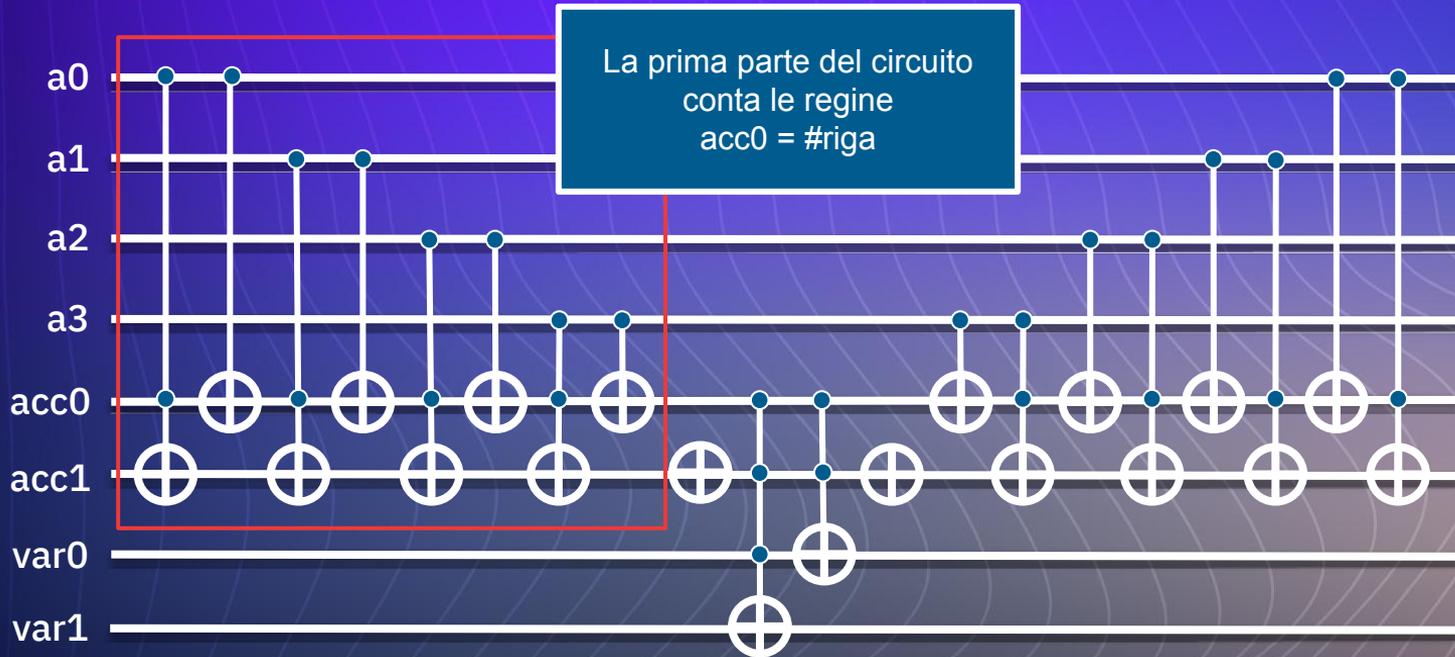
```
1111111111111010  
1111111111111011  
1111111111111100  
1111111111111101  
1111111111111110  
1111111111111111
```



# Grover algorithm

caso d'uso [problema delle N regine]

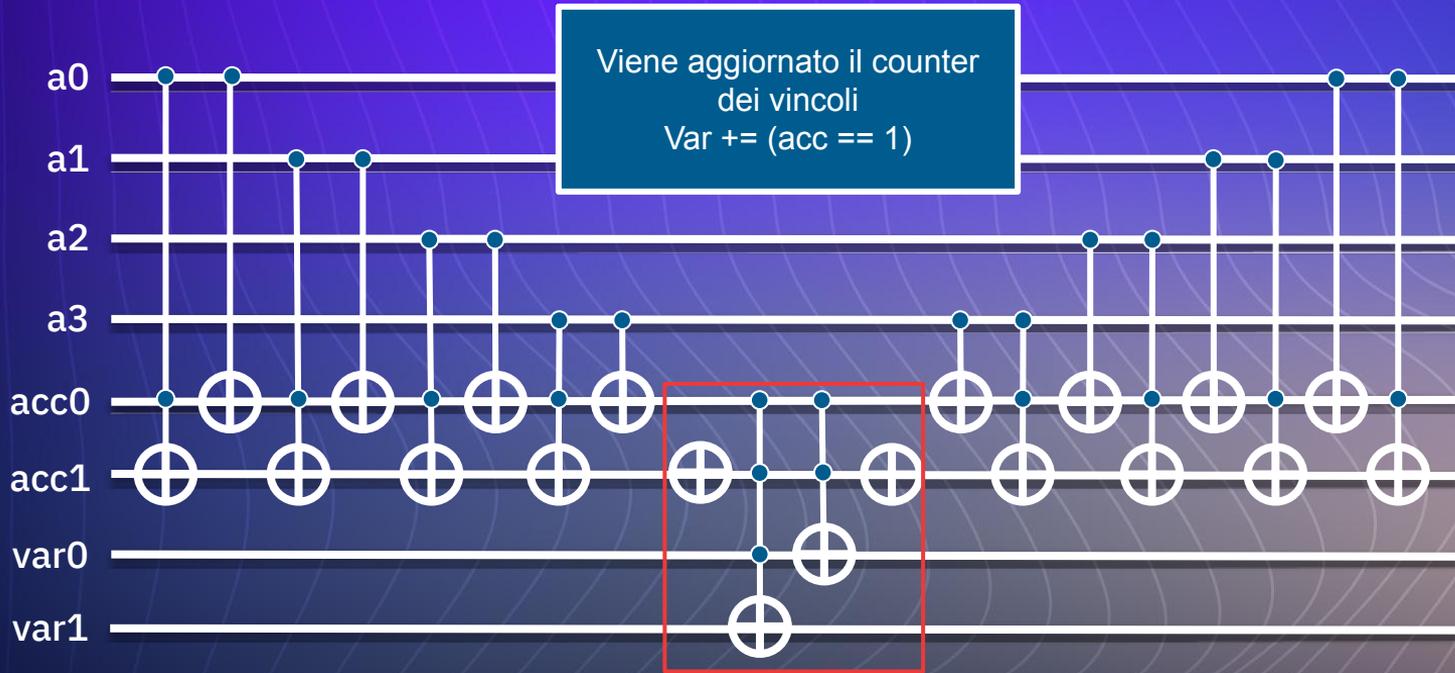
É necessario contare il numero di regine su ogni riga, colonna e diagonale  
Per essere soluzione il risultato di ogni somma deve sempre essere 1



a0	a1	a2	a3
b0	b1	b2	b3
c0	c1	c2	c3
d0	d1	d2	d3

# Grover algorithm

caso d'uso [problema delle N regine]

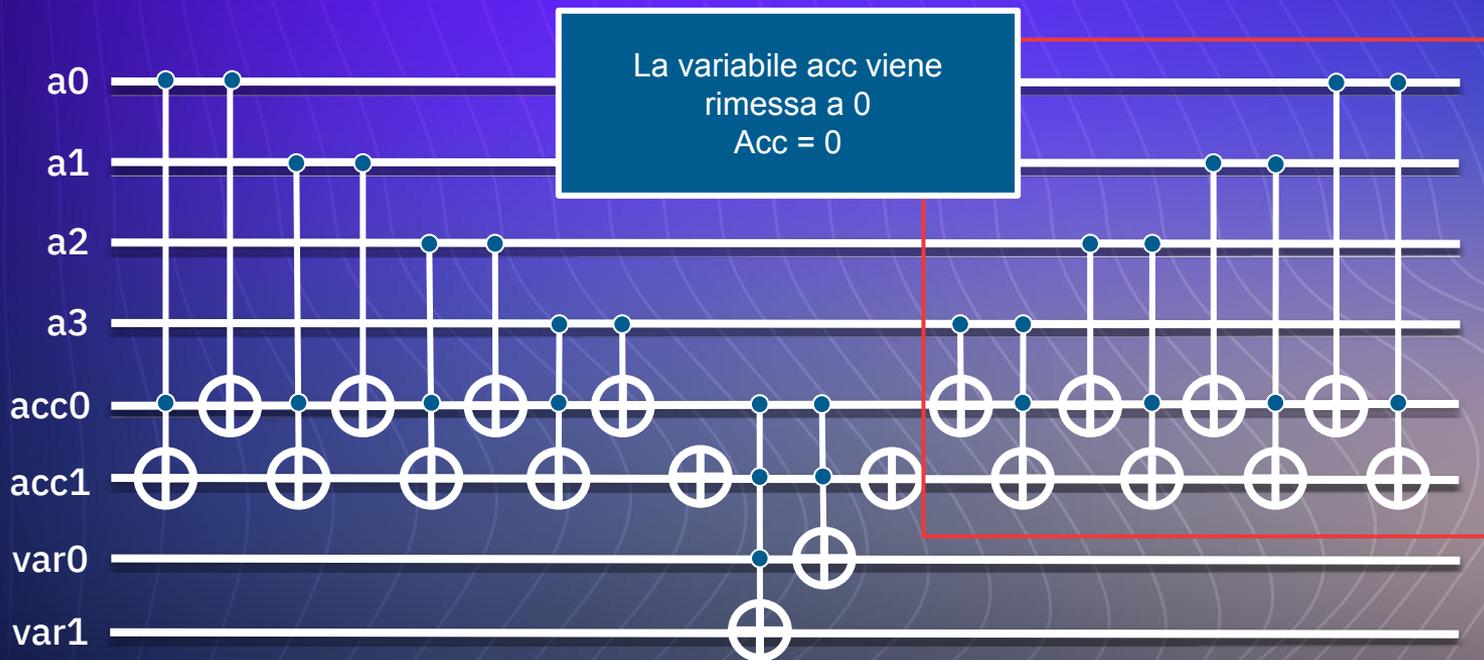


a0	a1	a2	a3
b0	b1	b2	b3
c0	c1	c2	c3
d0	d1	d2	d3

# Grover algorithm

caso d'uso [problema delle N regine]

Poiché la computazione deve essere reversibile, per riportare i qubit di acc a 0 è obbligatoria una fase di decomputazione



a0	a1	a2	a3
b0	b1	b2	b3
c0	c1	c2	c3
d0	d1	d2	d3

# Grover algorithm

caso d'uso [problema delle N regine]

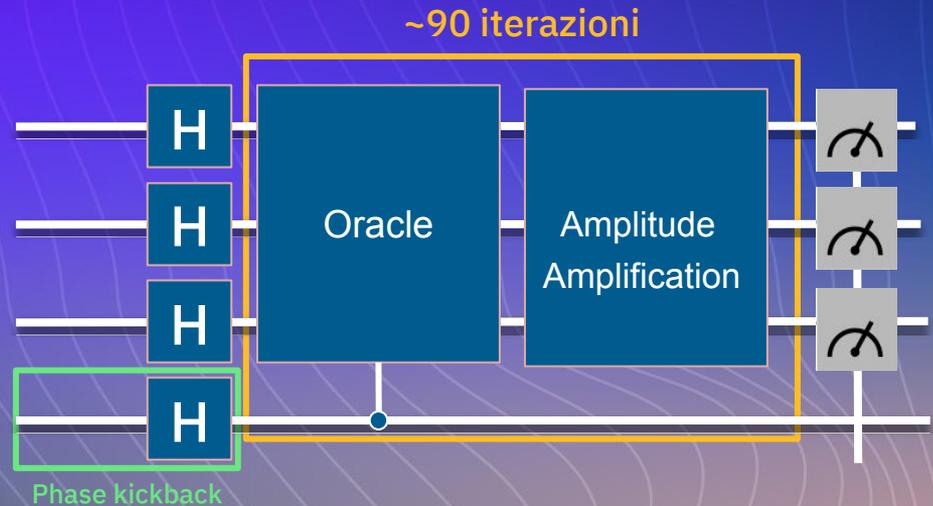
La procedura viene poi ripetuta per ogni vincolo sulla scacchiera (righe, colonne e diagonali)



a0	a1	a2	a3
b0	b1	b2	b3
c0	c1	c2	c3
d0	d1	d2	d3

# Grover algorithm

caso d'uso [problema delle N regine]

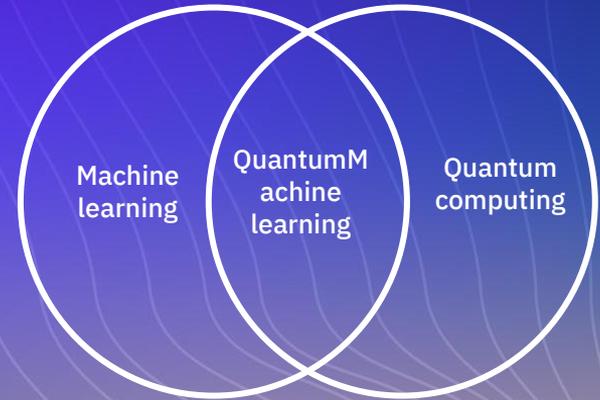


$NP \neq P$

# Quantum machine learning ?

Lo scopo della disciplina è quella di ottenere un vantaggio, in termini di speedup o modello, da un sistema quantistico

Data la situazione attuale delle macchine [NISQ] l'approccio a breve termine più efficiente è un approccio ibrido tra computer classico e quantistico

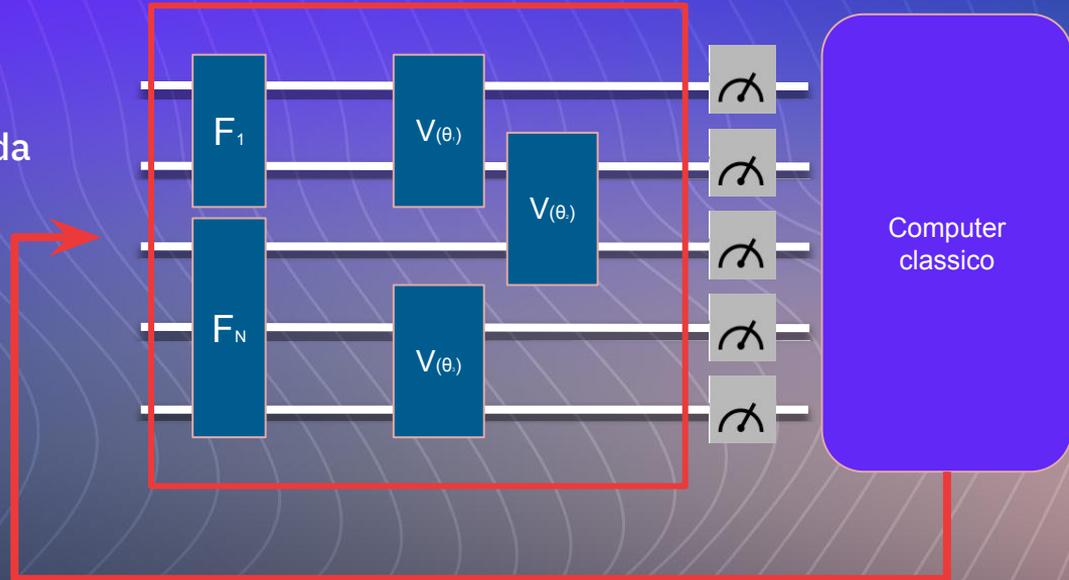


Parallelismo tra GPU e QPU

# Variational circuit

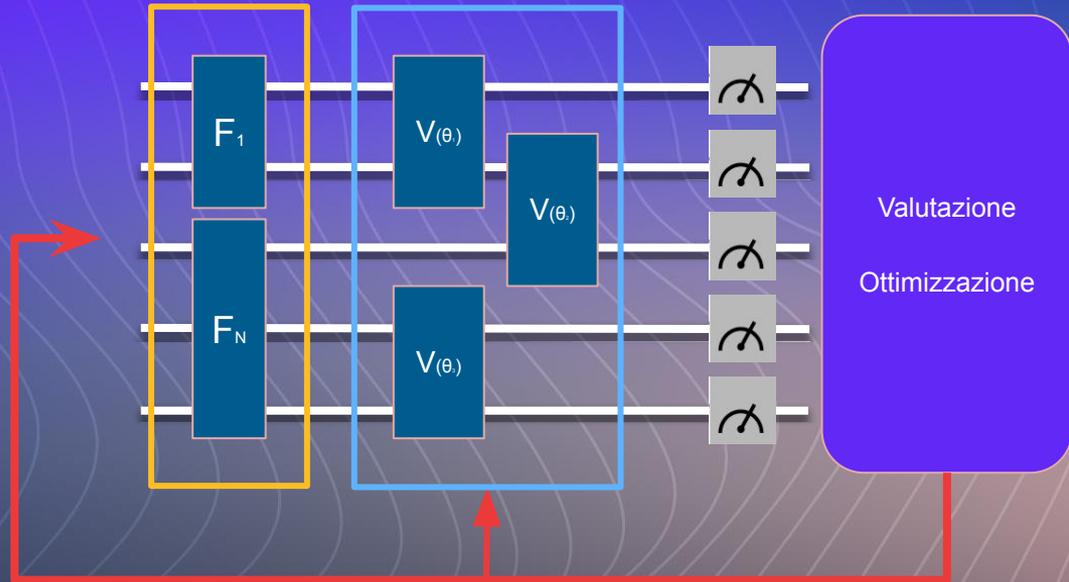
L'idea di base è quella di trattare i circuiti come fossero una black box

A ogni iterazione il modello viene valutato e aggiornato da un computer classico



# Variational circuit

1. Caricamento delle feature map
2. Esecuzione del modello
3. Valutazione e ottimizzazione
4. Update dei parametri
5. Nuova iterazione



# Variational circuit caricare il dataset

Per caricare il dataset è necessario costruire il circuito che permette di codificare nei nostri qubit i nostri dati

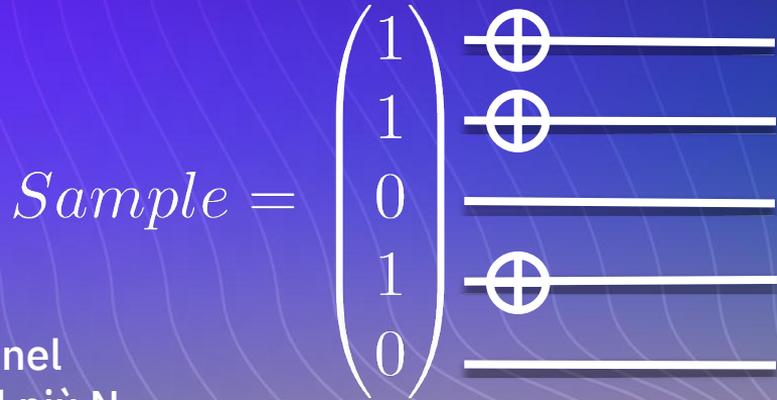
- Basis encoding
- Amplitude encoding
- Angle encoding
- Hamiltonian encoding

# Basis encoding

Il bit classico viene codificato direttamente con il qubit

Questa rappresentazione risulta efficiente nel caricamento, visto che vengono utilizzati al più N Not gate

Tuttavia risulta molto costoso in termini di qubit



# Amplitude encoding

Ogni numero viene codificato come coefficiente di uno stato quantistico

$$Sample = \begin{pmatrix} 4.343 \\ 32.3 \\ -24.01 \\ 1.00 \\ 0.2 \end{pmatrix}$$


The diagram shows a teal rectangular box labeled "Circuito altamente interlacciato" (Highly entangled circuit). It has three horizontal lines on the left side representing input qubits and three horizontal lines on the right side representing output qubits.

Questo vuol dire che è possibile rappresentare il nostro dataset in uno spazio esponenzialmente compresso

Il contro è che il circuito necessario per caricarlo usa un numero esponenziale di gate rispetto ai qubit usati (lineare rispetto ai dati)

# Angle encoding

Ogni numero viene codificato con un angolo di un qubit

$$Sample = \begin{pmatrix} 4.343 \\ 32.3 \\ -24.01 \\ 1.00 \\ 0.2 \end{pmatrix}$$


Questa rappresentazione dimostra un buon compromesso tra facilità di implementazione e quantità di qubit utilizzati

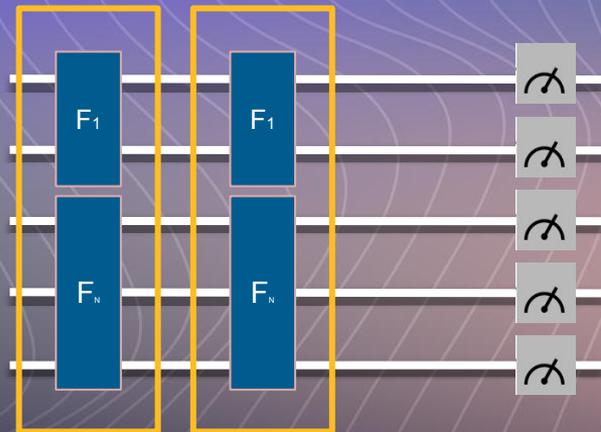
# Feature Map

L'encoding dei dati è sicuramente una parte delicata del processo

Va ricordato che una volta all'interno del formalismo quantistico sarà possibile applicare solo operatori lineari

- Basis encoding → non linear
- Amplitude encoding → linear
- Angle encoding → non linear

Le features maps possono essere combinate sia in serie che in parallelo

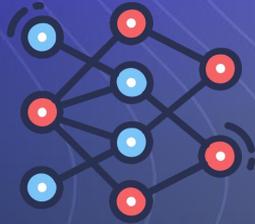


# Variational circuit conclusioni

Come possono essere interpretati questi modelli?

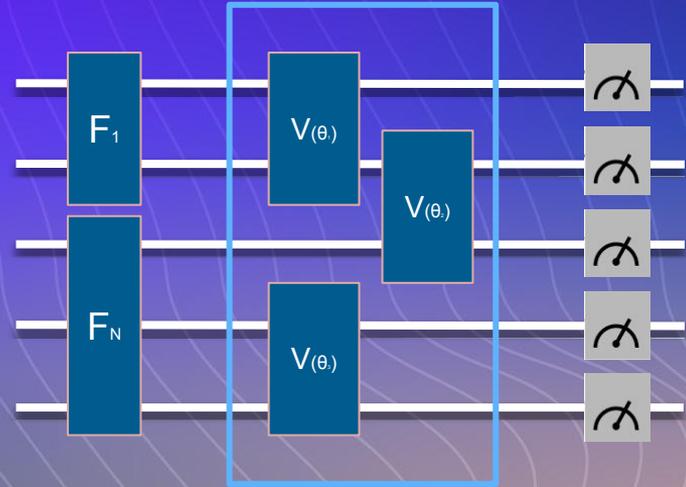
Potrebbe essere naturale pensare a un parallelismo diretto al deep learning ...

... in realtà data la mancanza di operatori non lineari non è istantanea la trasposizione



**REPEAT - UNTIL - SUCCESS**

Tramite l'uso del gate di misura si può emulare la funzione di attivazione



Un modello quantistico può essere visto come una somma di seni e coseni

# D wave - Quantum annealing

D-wave ha scelto di concentrarsi su un singolo algoritmo : Quantum annealing  
[versione quantistica del simulated annealing]

Questo algoritmo permette di risolvere problemi di ottimizzazione binaria

Inizia a diventare interessante dal punto di vista delle performance anche se rimane il collo di bottiglia del caricamento dati...

**D:wave**  
The Quantum Computing Company™

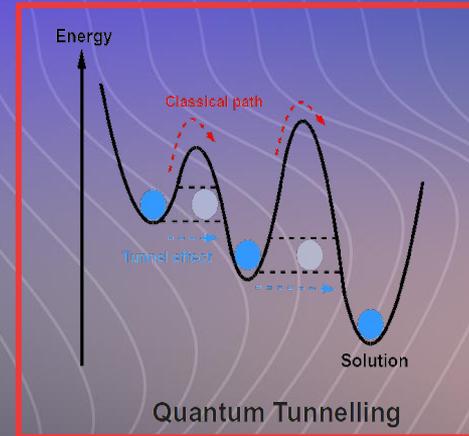


# Quantum annealing

L'algoritmo riesce a esplorare lo spazio delle soluzioni sfruttando le fluttuazione quantistiche

Questo vuol dire che, sebbene identico nello scopo, il Quantum annealing sfrutta meccanismi diversi rispetto alla sua controparte classica

Questo vuol dire che oltre a un potenziale speedup si può osservare anche un diverso comportamento

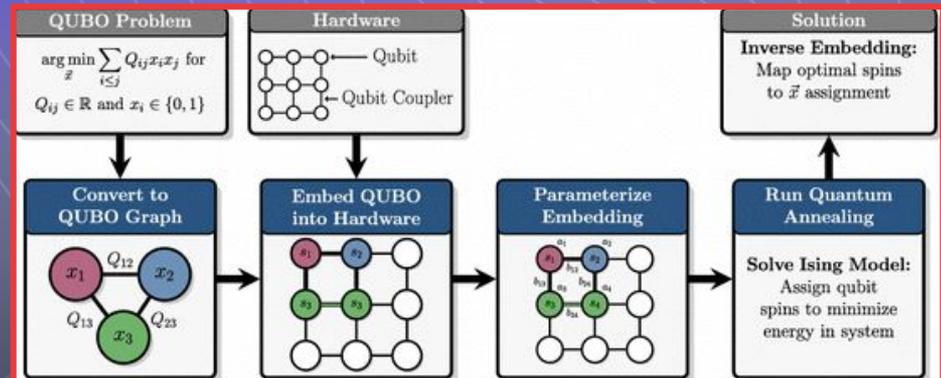


# Quantum annealing

Niente circuiti! Torniamo a ragionare con qualcosa di più familiare...

Dal punto di vista dell'utente basta solo concentrarsi sulla modellazione del problema, senza conoscenze di computazione quantistica

**QUBO**  
Quadratic Unconstrained Binary Optimization



# Quantum annealing QUBO

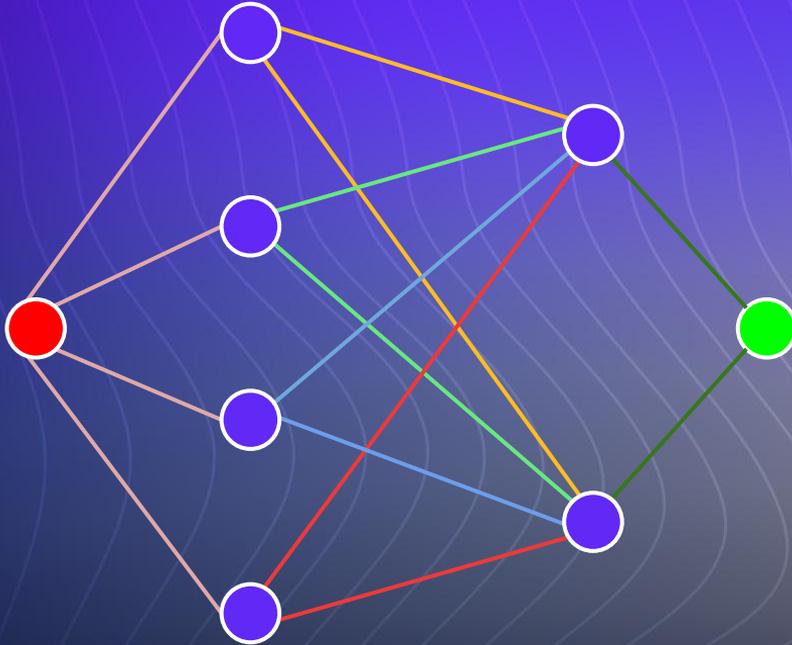
$$\min_{x_1, \dots, x_{N-1}} \sum_{i=0}^N \sum_{j=0}^N Q_{ij} x_i x_j + \sum_{i=0}^N b_i x_i$$

L'utente deve riuscire a descrivere il proprio problema definendo i pesi della matrice Q e il vettore b

# Quantum annealing

caso d'uso [routing optimization]

Dato un tempo  $T$  di viaggio, provare il percorso con costo minore, partendo dal nodo rosso verso il verde



0	1	10
1	0.5	50
2	3	5
3	4	17
4	1	23
5	1	9
6	0.1	48

7	0.3	3
8	1	24
9	2	95
10	2.4	42
11	1	80
12	0.1	4
13	2	2

○ COSTO

○ TEMPO

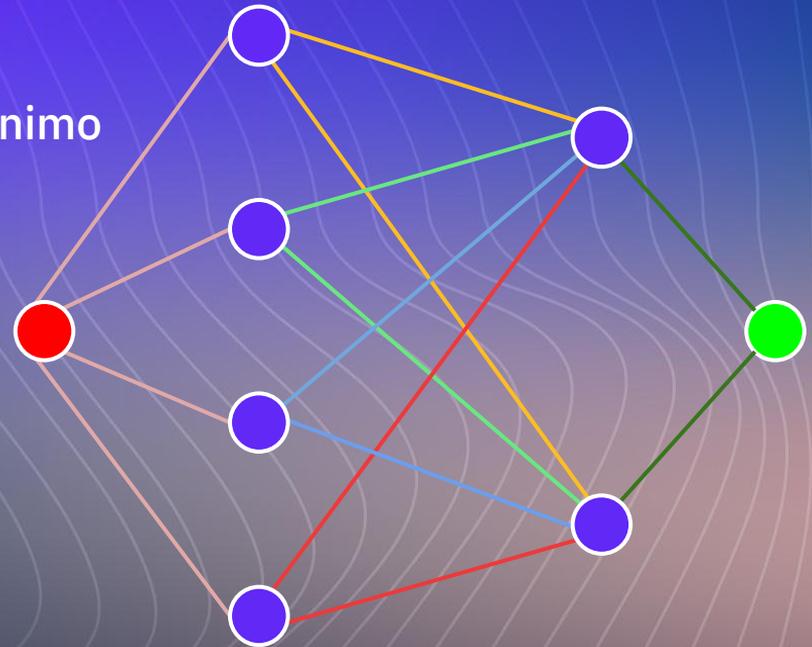
# Quantum annealing

caso d'uso [routing optimization]

La soluzione deve rispettare 3 condizioni:

- 1. Il percorso deve essere legale
- 2. Il tempo di viaggio deve tendere a T
- 3. Il costo complessivo deve essere minimo

Andando a modellare questi vincoli in  
forma QUBO...



# Quantum annealing

caso d'uso [routing optimization]

1. Il vincolo Grafico lo si modella andando a penalizzare o premiare la coesistenza tra coppie di archi

```
[  
  [0,1,1,1,-1,-1,0,0,0,0,0,0,0,0],  
  [0,0,1,1,0,0,-1,-1,0,0,0,0,0,0],  
  [0,0,0,1,0,0,0,0,-1,-1,0,0,0,0],  
  [0,0,0,0,0,0,0,0,0,0,-1,-1,0,0],  
  [0,0,0,0,0,1,1,1,1,1,1,1,-1,0],  
  [0,0,0,0,0,0,1,1,1,1,1,1,0,-1],  
  [0,0,0,0,0,0,0,1,1,1,1,1,-1,0],  
  [0,0,0,0,0,0,0,0,1,1,1,1,0,-1],  
  [0,0,0,0,0,0,0,0,0,1,1,1,-1,0],  
  [0,0,0,0,0,0,0,0,0,0,1,1,0,-1],  
  [0,0,0,0,0,0,0,0,0,0,0,1,-1,0],  
  [0,0,0,0,0,0,0,0,0,0,0,0,-1,1],  
  [0,0,0,0,0,0,0,0,0,0,0,0,0,1],  
  [0,0,0,0,0,0,0,0,0,0,0,0,0,0]  
]
```

2. Il vincolo dei tempi si ottiene sviluppando la formula diretta

$$\min \left( \sum_{i=0}^N tempo_i x_i - T \right)^2$$

$$\sum_{i=0}^N \sum_{j=i+1}^N 2tempo_i tempo_j x_i x_j + \sum_{i=0}^N (tempo_i^2 - 2T tempo_i) x_i$$

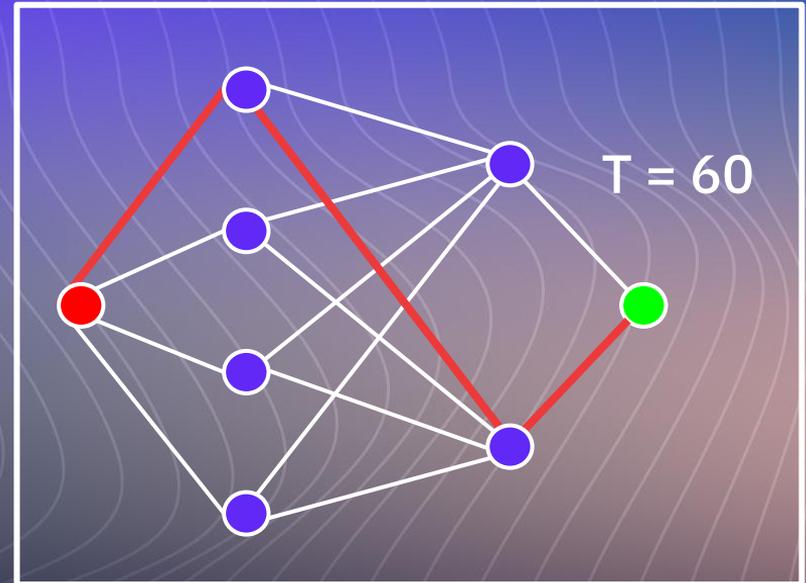
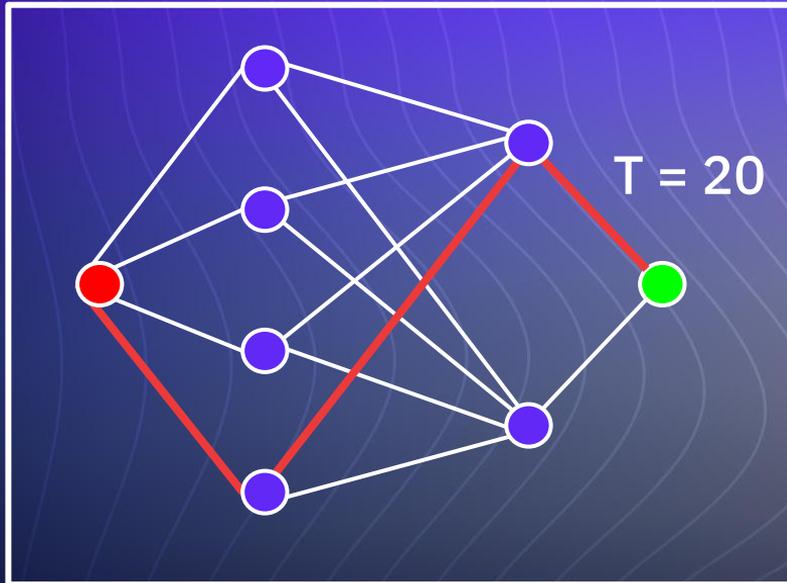
3. Il vincolo dei costi non è altro che il vettore  $b$  del problema QUBO

$$\sum_{i=0}^N x_i costo_i$$

# Quantum annealing

caso d'uso [routing optimization]

$$\alpha^2 [GRAFO] + \alpha [TEMPO] + [COSTO]$$



# Conclusioni

Il QC è oggi un fenomeno che va seguito, anche se l'impatto su temi di IA non è ancora così rilevante  
Gli investimenti sono molti e c'è molta aspettativa sul continuo miglioramento di queste macchine  
Iniziano a formarsi delle opzioni concrete di utilizzo



**Ennio Ottaviani**  
I think QC could be interesting for binary optimization using quantum search algorithms. Decision systems like job schedulers may benefit from this. Of course, a working quantum computer is needed but even with a moderate number of qbits

1 g [Mi piace](#) Rispondi 2

**Yann LeCun** ✓  
[Ennio Ottaviani](#) that's pretty much what D-Wave has been trying to sell. Exploiting quantum tunneling for more efficient combinatorial optimization. But even for that, it is far from clear that this can beat good

Scrivi una risposta...