Alma Mater Studiorum · Università di Bologna

Scuola di Scienze
Dipartimento di Fisica e Astronomia
Corso di Laurea in Fisica

Il Random Circuit Sampling e la supremazia quantistica

Relatore:
Prof.ssa Elisa Ercolessi

Presentata da:
Pierpaolo Vesce

Anno Accademico 2022/2023

Abstract

Lo scopo di questa tesi è di fare chiarezza sul concetto di supremazia quantistica e sulle modalità con cui questa è stata reclamata da Google tramite un esperimento di Random Circuit Sampling (RCS) nel 2019. Il capitolo di apertura presenta gli strumenti fondamentali dell'informatica quantistica, il qubit e le porte quantistiche. Nel capitolo seguente viene discussa la teoria del RCS e del protocollo di benchmarking utilizzato da Google per dichiarare raggiunta la supremazia quantistica, ovvero il Cross-Entropy Benchmarking. Infine, nel capitolo conclusivo è analizzato in dettaglio l'esperimento di RCS effettuato da Google sul loro processore quantistico Sycamore e vengono discussi i risultati ottenuti.

Indice

Introduzione

Nel 1981, quando la fisica e l'informatica erano ancora visti dalla comunità scientifica come due settori relativamente separati, si teneva presso il *Massachusetts Institute of Technology* la prima edizione della *Physics of Computation Conference*, alla quale parteciparono alcuni dei fisici di spicco dell'epoca, come Freeman Dyson, John Wheeler e Richard Feynman. Quest'ultimo, durante il discorso di apertura, avanzò l'idea che simulare la natura con un computer classico fosse inefficace, nonché in un certo senso fondamentalmente sbagliato, poiché la natura stessa non era classica e quindi la simulazione doveva essere quantistica [1]. Sebbene già qualche anno prima alcuni scienziati sotto la cortina di ferro erano giunti alla conclusione che fosse necessario lo sviluppo di un computer quantistico per simulare efficacemente i fenomeni naturali (quantistici) [2], è solo dopo la conferenza tenutasi al *MIT* che la questione della simulazione quantistica esce dalla sua nicchia di interesse per entrare nel *mainstream*.

Da questa nuova necessità nata agli inizi degli anni '80 di avere uno strumento per simulare efficacemente la realtà quantistica della natura fino allo sviluppo di un computer quantistico che potesse sfidare i super-computer nel calcolo sono passati quasi quattro decenni. Nel 2019, una divisione di Google dedicata alla ricerca sull'intelligenza artificiale, Google AI, dichiara di aver raggiunto la *supremazia quantistica* con un articolo pubblicato sulla rivista scientifica Nature [3]. La supremazia quantistica è un passo necessario verso l'effettiva simulazione di sistemi quantistici arbitrariamente complicati: è il punto in cui la potenza di calcolo dei processori quantistici supera quella dei super-computer classici, anche se testata su problemi creati ad-hoc e quindi di relativa utilità. Il task su cui sono stati testati il processore quantistico *Sycamore*, sviluppato da Google AI, e alcuni tra i più potenti super-computer (del 2019), tra cui Summit, è il *Random circuit sampling*, la cui teoria e implementazione sono discusse nel dettaglio rispettivamente nei Capitoli 2 e 3 di questa tesi. Il team che ha realizzato l'esperimento dichiara di aver eseguito un calcolo che avrebbe richiesto circa 10000 anni su un super-computer classico in soli 200 secondi su Sycamore, generando immediatamente discussioni e critiche nella comunità scientifica, specialmente dall'azienda IBM, sviluppatrice del super-computer Summit [4].

È importante chiarire che il concetto di supremazia quantistica non è eterno: è possibile che la tecnologia computazionale classica abbia uno sviluppo tale superare il punto in cui si trova la tecnologia computazionale quantistica, così da far perdere il primato del vantaggio quantistico. La proprietà transitoria della supremazia quantistica è in realtà uno stimolo per i settori di ricerca tecnologica e informatica sia classici che quantistici. Il più recente esperimento di Random circuit sampling atto a dimostrare la supremazia quantistica risale

ad Aprile 2023 [5], realizzato ancora una volta dal team di Google AI, questa volta su una versione di Sycamore a 70 *qubit* (il corrispettivo quantistico del *bit* classico, descritto in dettaglio nel Capitolo 1) invece che 53 (come nell'esperimento del 2019) A rimarcare quanto esposto prima, ossia la spinta allo sviluppo tecnologico sia classico che quantistico derivante dalla supremazia quantistica, nel 2022 viene completato lo sviluppo del primo super-computer exascale *Frontier*, il quale viene subito dichiarato il più veloce al mondo; è proprio quest'ultimo super-computer ad essere messo a confronto con Sycamore-70 in [5]. I risultati sottolineano l'aspetto temporaneo della supremazia quantistica: Frontier è capace di risolvere in circa 6 secondi il task eseguito su Sycamore-53 nel 2019, ma necessiterebbe di 47 anni per completare quello realizzato su Sycamore-70.

Il problema nell'applicazione concreta della potenza dimostrata dai computer quantistici negli esperimenti citati risiede nella fragilità di questi dispositivi: i qubit sono estremamente sensibili a interferenze esterne e possono essere mantenuti operativi solo per tempi che vanno dall'ordine dei microsecondi a quello dei millisecondi. Le operazioni sui qubit inoltre sono soggette a errori relativamente grandi (discussi in Sezione 3.2), quindi i computer quantistici necessitano di un sistema di correzione degli errori per essere affidabili. Nonostante questi (ed altri) problemi nell'implementazione di computer quantistici stabili, già nel 1994 con l'algoritmo di Shor [6] (uno dei primi ad essere stato progettato per essere eseguito su un computer quantistico) si trova una possibile applicazione della capacità computazionale quantistica. Questo algoritmo offre una notevole velocità di fattorizzazione di numeri interi in fattori primi: le risorse necessarie al calcolo passano così da un numero esponenziale nel caso classico ad un numero polinomiale nel caso quantistico, rispetto al numero di cifre del numero da fattorizzare. L'algoritmo di Shor è applicabile nell'ambito della crittografia e della crittoanalisi. Altri campi di applicazione della tecnologia computazionale quantistica sono la simulazione molecolare [7], l'ottimizzazione di sistemi complessi (come la gestione del traffico [8]) e in generale tutto ciò che necessita l'analisi di grandi moli di dati.

Date anche le critiche mosse alla chiarezza della presentazione dei dati sull'esperimento di Google del 2019 e a quelle sulla trasparenza, questo lavoro di tesi vuole essere esplicativo sia per la teoria del Random Circuit Sampling e del Cross-Entropy Benchmarking, sia per i risultati presentati da Google in [3], con il proposito di far avvicinare il lettore al complesso settore dell'informatica quantistica.

In questo lavoro di tesi è stato analizzato nel dettaglio l'esperimento di Random circuit sampling realizzato da Google nel 2019, che ritengo personalmente una pietra miliare nella storia dell'informatica e della fisica. Nel Capitolo 1 vengono presentati i concetti basilari dell'informatica quantistica, quali il qubit e le porte quantistiche. Nel Capitolo 2 è descritta la teoria dietro al Random circuit sampling, che spazia dalla fisica quantistica alla statistica e alla teoria dell'informazione. Sempre nel Capitolo 2 è presentato e descritto lo strumento matematico creato dal team di Google AI per testare il proprio

processore quantistico, ossia il *Cross-Entropy Benchmarking* (*XEB*). Infine nel Capitolo 3 sono esposti i dettagli dell'esperimento sulla supremazia quantistica di Google del 2019. Nell'Appendice A si trovano le nozioni teoriche sulla realizzazione fisica del qubit usato per la costruzione del processore quantistico Sycamore, ossia il *transmon qubit*.

Capitolo 1

Basi dell'informatica quantistica

In questo capitolo verranno esposti i concetti fondamentali dell'informatica quantistica. Si farà principalmente riferimento al libro di testo standard della materia, *Quantum Computation and Quantum Information* di Nielsen e Chuang [6].

1.1 Il qubit

Il *qubit* (abbreviazione di *quantum bit*) è l'analogo quantistico del *bit* classico. A differenza del bit classico, il quale può trovarsi nello stato 0 *oppure* 1, il qubit $|\psi\rangle$ può essere in uno stato di *sovrapposizione*:

$$|\psi\rangle = \alpha_0 |0\rangle + \alpha_1 |1\rangle \tag{1.1.1}$$

il quale non è altro che una combinazione lineare degli stati $|0\rangle$ e $|1\rangle$, dove $\alpha_0, \alpha_1 \in \mathbb{C}$ sono le *ampiezze* e devono rispettare la condizione di normalizzazione:

$$|\alpha_0|^2 + |\alpha_1|^2 = 1 \tag{1.1.2}$$

Lo stato di un qubit è quindi un vettore (unitario) di uno spazio vettoriale (di Hilbert) complesso bidimensionale, una delle cui basi (ortonormali) è formata proprio dagli stati $|0\rangle, |1\rangle$. La base ortonormale $\{|0\rangle, |1\rangle\}$ è chiamata *base computazionale*.
Un'altra differenza sostanziale tra bit e qubit consiste nella misura: lo stato di un bit classico (0 o 1) può essere determinato in maniera diretta, mentre lo stato di un bit quantistico, ossia il valore delle ampiezze α_0 e α_1, non può essere osservato direttamente. Quando misurato, il qubit restituirà il risultato 0 con probabilità $|\alpha_0|^2$ oppure 1 con probabilità $|\alpha_1|^2$. Lo stato di un bit classico inoltre può essere considerato come una variabile discreta i cui unici valori possibili sono $0, 1$, mentre un qubit esiste in un *continuum* di stati tra $|0\rangle$ e $|1\rangle$ (fino a che viene osservato).

Uno dei modi in cui i qubit vengono realizzati è tramite un sistema a due stati, ad esempio quelli di un elettrone orbitante attorno ad un singolo nucleo. I due stati sono chiamati, in ordine crescente di energia, *ground state*, indicato con $|0\rangle$, e *stato eccitato*, indicato con $|1\rangle$. È possibile muovere l'elettrone dal ground state $|0\rangle$ a quello eccitato $|1\rangle$, e viceversa, applicando un fascio di luce sull'atomo, con una certa energia e per un certo lasso di tempo. Regolando il tempo di esposizione alla luce, l'elettrone può essere

spostato [6] "a metà strada" tra lo stato $|0\rangle$ e $|1\rangle$, nello stato:

$$|+\rangle \equiv \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle \tag{1.1.3}$$

uno stato in cui una misura del qubit può dare il risultato 0 o 1 con il 50% di probabilità ciascuno.
I qubit nel processore *Sycamore* usato da *Google AI* per il loro esperimento sulla supremazia quantistica sono chiamati *transmon* e alcuni dettagli sulle loro caratteristiche verranno forniti in Appendice A.

1.1.1 Sfera di Bloch

Si può riscrivere la (1.1.1) usando la rappresentazione polare dei numeri complessi:

$$|\psi\rangle = \rho_0 e^{i\phi_0}|0\rangle + \rho_1 e^{i\phi_1}|1\rangle \tag{1.1.4}$$

Dato che [9] una funzione d'onda $|\psi\rangle$ normalizzata nello spazio di Hilbert rappresenta lo stesso stato quantistico della funzione d'onda $|\psi'\rangle = c|\psi\rangle$ con $|c| = 1$, si ha che:

$$e^{-i\phi_0}|\psi\rangle = \rho_0|0\rangle + \rho_1 e^{i(\phi_1-\phi_0)}|1\rangle = \rho_0|0\rangle + \rho_1 e^{i\phi}|1\rangle \tag{1.1.5}$$

Ora considerando la (1.1.2), si ottiene:

$$1 = |\rho_0|^2 + \left|\rho_1 e^{i\phi}\right|^2 = \rho_0^2 + \rho_1^2 \tag{1.1.6}$$

dove si è tenuto conto del fatto che $\rho_0, \rho_1 \geq 0$. Si conclude il calcolo operando la sostituzione: $\rho_0 = \cos\frac{\theta}{2}$, $\rho_1 = \sin\frac{\theta}{2}$ e si ha il risultato finale:

$$|\psi\rangle = \cos\frac{\theta}{2}|0\rangle + e^{i\phi}\sin\frac{\theta}{2}|1\rangle \tag{1.1.7}$$

Si può definire adesso un punto sulla *sfera di Bloch* (Fig. 1.1), tramite i numeri θ e ϕ. Gli elementi della base computazionale sono situati ai poli opposti della sfera sull'asse z. Le operazioni sui singoli qubit possono essere visualizzate su questa sfera come rotazioni e riflessioni rispetto a piani. Da notare che nel caso di più qubit non esiste una generalizzazione semplice della sfera di Bloch, relegandola quindi a strumento utile per intuizioni e per *visualizzare* un singolo qubit.

1.1.2 Sistemi a più qubit

Come in Sezione 1.1, si effettua un parallelismo con i bit classici: prendendo ad esempio due bit ci sarebbero quattro stati possibili, 00, 01, 10 e 11. Se quindi lo stato di un

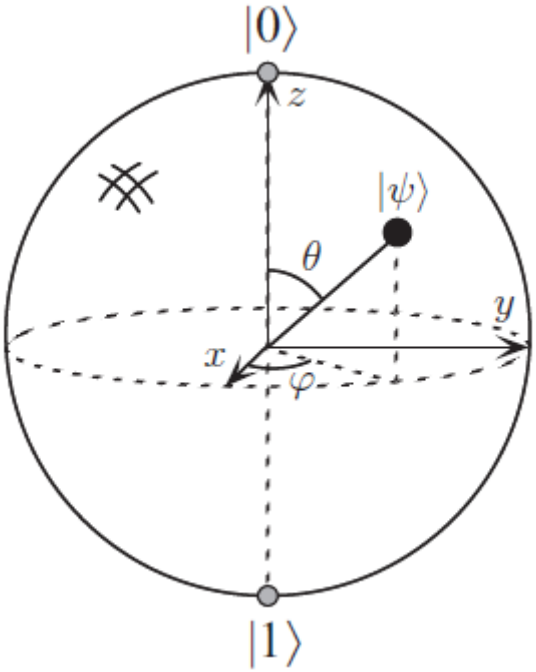


Fig. 1.1: Rappresentazione di un qubit nella sfera di Bloch. [6]

singolo qubit si esprime come combinazione lineare degli stati della base computazionale $\{|0\rangle, |1\rangle\}$, lo stato di un sistema a due qubit si scrive come combinazione lineare degli stati della base computazionale $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$:

$$|\psi\rangle = \alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle \tag{1.1.8}$$

Effettuando una misurazione lo stato del sistema dei due qubit *collassa* su uno dei quattro stati della base computazionale con probabilità $|\alpha_x|^2$ e quindi il risultato della misura è quello associato allo stato della base su cui collassa $|\psi\rangle$, ossia $|x\rangle$, con $x \in S = \{0,1\}^2 = \{00, 01, 10, 11\}$.
La condizione di normalizzazione per il sistema in esame è:

$$\sum_{x \in S} |\alpha_x|^2 = 1 \tag{1.1.9}$$

Da notare che misurando solamente un qubit del sistema, lo stato dopo la misura subisce una ulteriore normalizzazione. Ottenendo ad esempio il risultato 1 dopo la misura del secondo qubit, con probabilità $|\alpha_{01}|^2 + |\alpha_{11}|^2$, il nuovo stato del sistema si presenta in questa forma:

$$|\psi'\rangle = \frac{\alpha_{01}|01\rangle + \alpha_{11}|11\rangle}{\sqrt{|\alpha_{01}|^2 + |\alpha_{11}|^2}} \tag{1.1.10}$$

Generalizzando ad un sistema con n qubit, gli stati della base computazionale dello spazio di Hilbert (2^n-dimensionale) a cui appartiene lo stato del sistema hanno la forma

$|b_1 b_2 \dots b_n\rangle$, con $b_i \in \{0,1\}$. Uno stato di questo sistema è espresso come:

$$|\psi\rangle = \sum_{x\in\{0,1\}^n} \alpha_x |x\rangle \tag{1.1.11}$$

La condizione di normalizzazione è quindi:

$$\sum_{x\in\{0,1\}^n} |\alpha_x|^2 = 1 \tag{1.1.12}$$

Effettuando una misura sul sistema il risultato ottenuto è del tipo $x = b_1 b_2 \dots b_n \in \{0,1\}^n$. La quantità x, ossia il risultato della misura di tutti i qubit in un sistema, verrà chiamata *bitstring* nel resto della tesi.

1.2 Porte quantistiche

In questa sezione viene presentato l'analogo delle porte logiche usate nei circuiti costituenti i computer classici, ossia le porte quantistiche (o quantiche). Viene fatta una distinzione tra porte quantistiche operanti su un singolo qubit (*single qubit gates*) e quelle che operano su più di un qubit (*multiple qubit gates*).

Le porte quantiche vengono rappresentate da matrici U appartenenti al gruppo unitario $U(2^n)$ dove n indica il numero di qubit su cui opera la porta. L'unitarietà è l'unico requisito per avere una porta quantistica, come conseguenza della conservazione del prodotto scalare:

$$\langle\psi|\phi\rangle \overset{\text{unitarietà}}{=} \left\langle U^\dagger U\psi \middle| \phi \right\rangle = \langle U\psi|U\phi\rangle \tag{1.2.1}$$

Alla fine del capitolo, in Fig. 1.2 si può osservare una lista delle porte quantistiche a singolo qubit più comuni con rappresentazione matriciale e visiva sulla sfera di Bloch.

Di fondamentale importanza le operazioni X, Y, Z connesse alle matrici di Pauli. Queste infatti sono utilizzate per modellare diverse tipologie di errori (detti errori di Pauli) riscontrabili nella trasmissione di qubit [10]:

- **Errore bit-flip.** Lo stato del qubit viene invertito, ossia gli viene applicata una porta *NOT* (la matrice di Pauli $\sigma_X = X$):

  $$|\psi\rangle = (\alpha_0 |0\rangle + \alpha_1 |1\rangle) \mapsto (\alpha_1 |0\rangle + \alpha_0 |1\rangle) = X |\psi\rangle \tag{1.2.2}$$

  In pratica compie una rotazione di π radianti attorno all'asse x ($R_X(\pi)$) sulla sfera di Bloch, realizzando la mappa $|0\rangle \mapsto |1\rangle$, $|1\rangle \mapsto |0\rangle$. Per questo motivo X viene anche chiamata bit-flip.

- **Errore phase-flip.** In riferimento all'equazione (1.1.7), quando avviene questo errore lo stato subisce una rotazione di π radianti attorno all'asse z ($R_Z(\pi)$), ossia

la fase ϕ varia di π radianti. Ciò equivale all'applicazione della porta $\sigma_z = Z$ (chiamata anche *phase gate*):

$$|\psi\rangle = (\alpha_0 |0\rangle + \alpha_1 |1\rangle) \mapsto (\alpha_0 |0\rangle - \alpha_1 |1\rangle) = Z |\psi\rangle \tag{1.2.3}$$

L'effetto è quello della mappa $|0\rangle \mapsto |0\rangle$, $|1\rangle \mapsto -|1\rangle$.

- **Entrambi.** La mappa dell'errore è quasi equivalente alla combinazione degli altri due, a meno di un fattore i: $|0\rangle \mapsto i|1\rangle$, $|1\rangle \mapsto -i|0\rangle$. Questo è il risultato dell'applicazione della porta $\sigma_Y = Y$:

$$|\psi\rangle = (\alpha_0 |0\rangle + \alpha_1 |1\rangle) \mapsto (i\alpha_1 |0\rangle - i\alpha_0 |1\rangle) = Y |\psi\rangle \tag{1.2.4}$$

In linea con le altre porte di Pauli, anche questa equivale ad una rotazione di π radianti sull'asse y ($R_Y(\pi)$).

Le porte singole utilizzate nell'esperimento in esame nel capitolo 3 sono:

$$\sqrt{X} \equiv R_X(\frac{\pi}{2}) = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & -i \\ -i & 1 \end{bmatrix} \tag{1.2.5}$$

$$\sqrt{Y} \equiv R_Y(\frac{\pi}{2}) = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix} \tag{1.2.6}$$

$$\sqrt{W} \equiv R_{X+Y}(\frac{\pi}{2}) = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & -\sqrt{i} \\ \sqrt{-i} & 1 \end{bmatrix} \tag{1.2.7}$$

Le porte quantistiche a due qubit (chiamate anche porte doppie nel resto della tesi) sono rappresentate da matrici unitarie 4×4: a differenza delle porte logiche classiche, le quali ricevendo in input 2 bit ne restituiscono 1, le porte doppie quantistiche ricevono in input 2 qubit e restutiscono 2 qubit come output.

Nel capitolo 3 saranno di cruciale importanza la porta doppia *controlled-Z*(CZ) o *controlled-phase*:

$$CZ = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} \tag{1.2.8}$$

la quale è un esempio di porta *controllata*: operando su uno stato $|x_1 x_2\rangle = |x_1\rangle \otimes |x_2\rangle$, viene applicata una porta Z sul qubit $|x_2\rangle$ solo se $|x_1\rangle = |1\rangle$.

Un'altra porta doppia fondamentale per il capitolo 3 è la porta *iSWAP*:

$$iSWAP = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & i & 0 \\ 0 & i & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1.2.9}$$

Infine, la porta più importante forse di tutto l'esperimento di RCS esposto nel capitolo 3 è la porta *fSim*, da *fermionic simulator* [3], un ibrido tra CZ e $iSWAP$:

$$fSim(\theta, \phi) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -i\sin\theta & 0 \\ 0 & -i\sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & e^{-i\phi} \end{bmatrix} \tag{1.2.10}$$

Una questione importante è quella del set di porte universale: nel caso classico, la porta *NAND* o la porta *NOR* sono considerate porte universali, nel senso in cui è possibile realizzare qualsiasi circuito utilizzando una sola di queste. Nel caso quantistico si dice che un insieme di porte è universale se qualsiasi operazione unitaria può essere *approssimata* con precisione arbitraria da un circuito quantistico formato solo da porte dell'insieme [6]. È stato dimostrato in [11] che l'insieme di porte $\{CZ, \sqrt{X}, \sqrt{W}\}$ è un insieme di porte quantistiche universale e come conseguenza su Sycamore è possibile realizzare qualsiasi tipo di circuito quantistico.

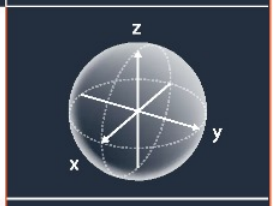
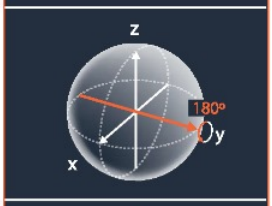
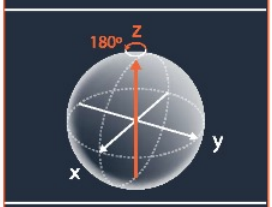
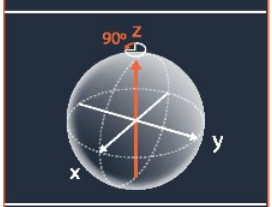
GATE	CIRCUIT REPRESENTATION	MATRIX REPRESENTATION	TRUTH TABLE	BLOCH SPHERE
I Identity-gate: no rotation is performed.		$I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$	Input / Output $\|0\rangle$ / $\|0\rangle$ $\|1\rangle$ / $\|1\rangle$	
X gate: rotates the qubit state by π radians (180°) about the x-axis.		$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$	Input / Output $\|0\rangle$ / $\|1\rangle$ $\|1\rangle$ / $\|0\rangle$	
Y gate: rotates the qubit state by π radians (180°) about the y-axis.		$Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$	Input / Output $\|0\rangle$ / $i\|1\rangle$ $\|1\rangle$ / $-i\|0\rangle$	
Z gate: rotates the qubit state by π radians (180°) about the z-axis.		$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$	Input / Output $\|0\rangle$ / $\|0\rangle$ $\|1\rangle$ / $-\|1\rangle$	
S gate: rotates the qubit state by $\frac{\pi}{2}$ radians (90°) about the z-axis.		$S = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{2}} \end{pmatrix}$	Input / Output $\|0\rangle$ / $\|0\rangle$ $\|1\rangle$ / $e^{i\frac{\pi}{2}}\|1\rangle$	
T gate: rotates the qubit state by $\frac{\pi}{4}$ radians (45°) about the z-axis.		$T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{4}} \end{pmatrix}$	Input / Output $\|0\rangle$ / $\|0\rangle$ $\|1\rangle$ / $e^{i\frac{\pi}{4}}\|1\rangle$	
H gate: rotates the qubit state by π radians (180°) about an axis diagonal in the x-z plane. This is equivalent to an X-gate followed by a $\frac{\pi}{2}$ rotation about the y-axis.		$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$	Input / Output $\|0\rangle$ / $\frac{\|0\rangle + \|1\rangle}{\sqrt{2}}$ $\|1\rangle$ / $\frac{\|0\rangle - \|1\rangle}{\sqrt{2}}$	

Fig. 1.2: Esempi di porte logiche a un qubit, tratto da [12]

Capitolo 2

Random Circuit Sampling

In questo capitolo verrà illustrato in cosa consiste il *Random Circuit Sampling* (RCS) e perché è stato utilizzato per dimostrare la supremazia quantistica. Verranno inoltre forniti gli strumenti matematici necessari alla comprensione dell'esperimento, seguendo in gran parte l'utile guida scritta da Sean Mullane, del Dipartimento di Fisica Applicata dell'Università di Stanford [13].

Verrà dimostrato come campionare circuiti dalla *misura di Haar* (si veda Sezione 2.2) induce una distribuzione delle probabilità degli output che tende alla distribuzione di *Porter-Thomas* (discussa in Sezione 2.3), la quale verrà utilizzata, insieme alla proprietà di equipartizione asintotica (*AEP*, da *asymptotic equipartition property*, trattata nella Sezione 2.4) per dimostrare che le bitstring campionate da un circuito quantistico sono e volte più probabili di quelle campionate dalla simulazione classica.
Nella sezione 2.5 si dimostrerà che:

$$\frac{\prod_{x\in S} |\langle x|\psi\rangle|^2}{\prod_{x\in S_{cl}} |\langle x|\psi\rangle|^2} \sim e^m \tag{2.0.1}$$

dove un campione S di m misurazioni del circuito quantistico U con risultati $x \in \{0,1\}^n$ nella base computazionale ha probabilità $\prod_{x\in S} |\langle x|\psi\rangle|^2$. Con S_{cl} viene indicato il campione di m bitstring risultato dalla simulazione classica e con $|\psi\rangle = U|\psi_0\rangle$ lo stato finale del sistema.
È doveroso notare che la (2.0.1) è valida in caso di circuiti *error-less*: nel pre-print di [14] il membro a destra della (2.0.1) è riportato come $e^{me^{-rg}}$, dove r è un parametro relativo al *per-gate error rate* e g è il numero totale di porte quantistiche, direttamente proporzionale al numero di qubit n e alla profondità del circuito d. Nel resto del capitolo viene assunto che $r \approx 0$.

Verranno usati poi dei risultati parziali ottenuti nel corso della dimostrazione della (2.0.1) per calcolare la differenza di entropia incrociata (*XED*, da *cross entropy difference*), strumento principale del protocollo di *benchmark* chiamato appunto *Cross-entropy benchmark* (*XEB*).

2.1 Cosa è il Random Circuit Sampling?

Si consideri un sistema di n-qubit in uno spazio di Hilbert 2^n dimensionale $\mathcal{H}^{\otimes n}$. Nella sezione 1.2 è stato illustrato come un circuito quantistico può essere rappresentato matematicamente da una matrice unitaria U che fa evolvere il sistema di qubit considerato dallo stato iniziale $|\psi_0\rangle$ allo stato finale $|\psi\rangle = U|\psi_0\rangle$. Misurando nella base computazionale ognuno degli n qubit dello stato finale si ottiene la bitstring $x = b_1 b_2 ... b_n$ dove $b_i \in \{0, 1\}$. Alla bitstring ottenuta corrisponde quindi uno stato della base $|x\rangle = |b_1 b_2 ... b_n\rangle$ del sistema di n-qubit considerato. La probabilità che ha lo stato finale del sistema $|\psi\rangle$ di *collassare* nello stato della base $|x\rangle$ a seguito della misurazione è dato da:

$$p_U(x) = |\langle x|\psi\rangle|^2 \tag{2.1.1}$$

L'espressione sopra indica la probabilità di osservare lo stato della base $|x\rangle$ a seguito della misurazione dello stato $|\psi\rangle$ del sistema fatto evolvere con un circuito definito dalla matrice unitaria U. Questo si intende con campionare (l'output di) un circuito quantistico: far evolvere lo stato iniziale del sistema con una matrice unitaria U per poi misurare i qubit dello stato finale ed ottenere una bitstring x.

Il processo computazionale in esame, ossia il campionamento dell'output di un circuito quantistico (pseudo-)casuale, prevede appunto che i circuiti campionati siano casuali e di conseguenza occorre che gli operatori unitari U siano scelti in modo casuale. Per evitare "preferenze" sulle U selezionate, queste dovranno essere prese da una distribuzione uniforme sullo spazio delle matrici unitarie $\mathbb{U}^{2^n \times 2^n}$.

Si considera un operatore unitario casuale, tralasciando per il momento il modo in cui lo si ottiene. Si applica il circuito U allo stato iniziale degli n-qubit (di solito $|\psi_0\rangle = |0\rangle^{\otimes n}$), viene misurato lo stato finale $|\psi\rangle = U|\psi_0\rangle$ e si ottiene una bitstring s. Ripetendo questa operazione m volte (con lo stesso operatore U) si estrae un insieme di bitstring $S = \{x_1, x_2, ..., x_m\}$. La probabilità di avere S dopo m campionamenti è:

$$\Pr(S) = \prod_{x \in S} p_U(x) = \prod_{x \in S} |\langle x|\psi\rangle|^2 \tag{2.1.2}$$

dove si è indicato con $\langle x|$ lo stato della base associato alla bitstring x.

Il fulcro del *task* è dato dal confronto tra il caso quantistico e quello classico, in questo caso consistente nella simulazione tramite classici (super)computer. In teoria occorrerebbe semplicemente ricreare tramite un normale computer lo stato iniziale del sistema $\left|\psi_{0,cl}\right\rangle$, il circuito U_{cl} e quindi lo stato finale $|\psi_{cl}\rangle$ da misurare per ottenere una bitstring x_{cl}.

Il problema nel realizzare ciò è che la dimensione dello spazio degli stati per un sistema di n qubit è *esponenziale* in n, per cui oltre un certo numero di qubit diventa impossibile per un computer classico simulare perfettamente il sistema (se $n \simeq 50$ il sistema è de-

scritto da $2^{50} \approx 10^{15}$ numeri complessi). Allora la simulazione tramite classici computer sarà un'approssimazione *polinomiale*, ossia l'algoritmo impiegato per la simulazione userà risorse *polinomiale* in n: ciò significa che il circuito simulato classicamente non sarà identico all'originale quantistico e pertanto i campioni ricavati saranno in qualche maniera "diversi".

Un modo per quantificare la differenza tra gli insiemi di campioni ottenuti dal computer quantistico e dalla simulazione classica è confrontare le probabilità di ottenere i due set di sample; se si indica con $S_{cl} = \{x_1^{cl}, x_2^{cl}, ..., x_m^{cl}\}, \left\langle x^{cl} \right|$ rispettivamente l'insieme dei campioni ottenuti e lo stato della base relativi alla simulazione classica, allora:

$$\Pr(S_{cl}) = \prod_{x^{cl} \in S_{cl}} p_{cl}(x^{cl}) = \prod_{x^{cl} \in S_{cl}} \left| \left\langle x^{cl} \middle| \psi \right\rangle \right|^2 \tag{2.1.3}$$

è la probabilità di ottenere l'insieme di campioni S_{cl} dall'algoritmo classico se la simulazione del circuito fosse perfetta. Si nota infatti che il prodotto interno è eseguito tra lo stato della base $\left\langle x^{cl} \right|$ e $|\psi\rangle$, ossia lo stato finale del circuito quantistico originale e non $|\psi_{cl}\rangle$ ovvero il circuito simulato classicamente. È comune inoltre nella teoria del caos (quantistico) assumere [14] che i campioni raccolti dalla simulazione non sono correlati alla distribuzione definita da $|\psi\rangle$.

La differenza tra i due set $\Pr(S)$ e $\Pr(S_{cl})$ è osservabile fintanto che l'algoritmo classico utilizzi risorse *polinomiali* in n (se i computer classici avessero abbastanza risorse simulerebbero perfettamente il circuito quantistico e il concetto stesso di *quantum supremacy* non avrebbe senso) e soprattutto finché il circuito quantistico abbia un *error rate* basso. Assumendo un *error rate* accettabile si dimostrerà nelle sezioni successive che:

$$\mathbb{E}_U[\log \Pr(S) - \log \Pr(S_{cl})] \approx m \tag{2.1.4}$$

dove m è il numero di bitstring campionate e il valore di aspettazione è calcolato su un insieme di operatori unitari casuali. Questa equazione ci rivela l'importante proprietà di un generico sample S di rappresentare una firma (*signature*) di quel circuito [14].

2.2 Misura di Haar

In questa sezione vengono forniti elementi di teoria delle matrici casuali, principalmente tratti da un articolo del fisico Freeman Dyson [15].

Com'è possibile selezionare in maniera casuale delle matrice unitarie dallo spazio $\mathbb{U}^{2^n \times 2^n}$? La costruzione di una distribuzione uniforme di matrici unitarie parte dal considerare qualsiasi matrice complessa $A \in \mathbb{C}^{2^n \times 2^n}$ come un punto di uno spazio Euclideo $2N^2$-dimensionale, con $N = 2^n$. Quindi, essendo $\mathbb{U}^{2^n \times 2^n} \subset \mathbb{C}^{2^n \times 2^n}$, una matrice unitaria equivale ad un punto $(p_1, ..., p_M)$, con $M = 2^{2n+1}$ [13]. Nel corso dello studio degli spettri

energetici dei nuclei di atomi pesanti, il fisico Freeman Dyson si occupò della formalizzazione di alcuni insiemi della meccanica statistica; quello di maggiore interesse in questo caso è il Circular Unitary Ensemble (CUE o E_2(N)), utilizzato per sistemi non-invarianti per inversioni temporali. Il primo risultato di interesse per questa tesi che si trova in [15] è la probabilità che un sistema di E_2 (ossia una matrice unitaria) appartenga all'elemento di volume dS:

$$\mathrm{P}(dS) = (V_2)^{-1}\mu(dS) \tag{2.2.1}$$

dove

$$V_2 = \int \mu(dS) \tag{2.2.2}$$

è il volume totale dello spazio T_2.

L'altro risultato importante è il seguente teorema:

L'insieme unitario E_2 è unicamente definito, nello spazio T_2 delle matrici unitarie, dalla proprietà di essere invariante sotto ogni automorfismo

$$S \to USW \tag{2.2.3}$$

di T_2 in se stesso, dove U, W sono due matrici qualsiasi di T_2.

La quantità P(dS) viene chiamata la *misura di Haar* dell'insieme E_2 e denotata con $\eta(S)$.

L'importanza della misura è data dalla condizione di uniformità, la quale porterà la distribuzione delle probabilità $p_U(x)$ a tendere a una distribuzione chiamata di *Porter-Thomas*.

2.3 Distribuzione di Porter-Thomas

In questa sezione si dimostra, facendo riferimento principalmente a [13] e [16], che la distribuzione delle probabilità $p = p_U(x) = p(x) = |\langle x|\psi\rangle|^2$ è una distribuzione Beta:

$$\Pr(p) = (N-1)(1-p)^{N-2} \tag{2.3.1}$$

dove $N = 2^n$ è la dimensione dello spazio di Hilbert. Per $N >> 1$ una buona approssimazione è quella che viene chiamata distribuzione di *Porter-Thomas*:

$$\Pr(p) = Ne^{-Np} \tag{2.3.2}$$

Si considera un generico stato

$$|\psi\rangle = \sum_x (a_x + ib_x)\,|x\rangle \tag{2.3.3}$$

nella base computazionale $\{|x\rangle\}$.

Scegliere uno stato in maniera casuale equivale allo scegliere casualmente una matrice unitaria U dalla misura di Haar e applicarla ad uno stato iniziale $|0^n\rangle$. Avere uno stato scelto casualmente equivale a dire che gli $a_x, b_x \in \mathbb{R}$ sono casuali, soggetti al vincolo (unico) della normalizzazione:

$$\sum_x a_x^2 + b_x^2 = 1 \tag{2.3.4}$$

Per calcolare la probabilità di ottenere $|x_0\rangle$ da $|\psi\rangle$ si calcola $p(x_0) = |\langle x_0|\psi\rangle|^2 = a_{x_0}^2 + b_{x_0}^2$. Per calcolare invece la probabilità $\Pr(p)$ di ottenere $p(x_0)$ si opera sui volumi:

$$\Pr(p) = \frac{\mathrm{Vol}(H_{p,1})}{\mathrm{Vol}(H_1)} \tag{2.3.5}$$

dove $\mathrm{Vol}(H_{p,1})$ è il volume del sottospazio degli stati normalizzati con la condizione $p(x) = a_{x_0}^2 + b_{x_0}^2$ e $\mathrm{Vol}(H_1)$ è il volume del sottospazio degli stati normalizzati totali. In particolare:

$$\mathrm{Vol}(H_{p,1}) = \int_{-\infty}^{\infty} \prod_x da_x db_x \, \delta\left(\sum_x a_x^2 + b_x^2 - 1\right) \delta\left(a_{x_0}^2 + b_{x_0}^2 - p\right) \tag{2.3.6}$$

$$\mathrm{Vol}(H_1) = \int_{-\infty}^{\infty} \prod_x da_x db_x \, \delta\left(\sum_x a_x^2 + b_x^2 - 1\right) \tag{2.3.7}$$

Le funzioni delta sono utilizzate per imporre le condizioni di normalizzazione e la condizione su $p(x_0)$. Il calcolo degli integrali prosegue usando un risultato standard delle delta di Dirac:

$$\delta\left(\sum_x a_x^2 + b_x^2 - 1\right) = \frac{1}{2\pi} \int_{-\infty}^{\infty} dt \, e^{-it} \prod_x e^{it(a_x^2 + b_x^2)} \tag{2.3.8}$$

$$\delta\left(a_{x_0}^2 + b_{x_0}^2 - p\right) = \frac{1}{2\pi} \int_{-\infty}^{\infty} du \, e^{iu(a_{x_0}^2 + b_{x_0}^2) - iup} \tag{2.3.9}$$

per cui i due integrali assumono la forma:

$$\begin{aligned} \mathrm{Vol}(H_{p,1}) = \frac{1}{2\pi} \int_{-\infty}^{\infty} dt \, e^{-it} \left(\int_{-\infty}^{\infty} da_x db_x \, e^{it(a_x^2 + b_x^2)}\right)^{N-1} \cdot \\ \cdot \frac{1}{2\pi} \int_{-\infty}^{\infty} du \, e^{-iup} \int_{-\infty}^{\infty} da_x db_x \, e^{i(t+u)(a_x^2 + b_x^2)} \end{aligned} \tag{2.3.10}$$

$$\mathrm{Vol}(H_1) = \frac{1}{2\pi} \int_{-\infty}^{\infty} dt \, e^{-it} \left(\int_{-\infty}^{\infty} da_x db_x \, e^{it(a_x^2 + b_x^2)}\right)^{N} \tag{2.3.11}$$

Da notare che:

$$\int_{-\infty}^{\infty} da_x db_x \, e^{it(a_x^2 + b_x^2)} = \int_{-\infty}^{\infty} da_x \, e^{ita_x^2} \cdot \int_{-\infty}^{\infty} db_x \, e^{itb_x^2} = \left(\frac{i\pi}{t}\right) \tag{2.3.12}$$

dove è stato utilizzato un risultato ottenibile dall'integrale di Gauss. Allora utilizzando il

risultato sopra i volumi dei sottospazi prendono la forma:

$$\mathrm{Vol}(H_{p,1}) = \frac{1}{2\pi}\int_{-\infty}^{\infty} dt\, e^{-it}\left(\frac{i\pi}{t}\right)^{N-1} \cdot \frac{1}{2\pi}\int_{-\infty}^{\infty} du\, e^{-iup}\left(\frac{i\pi}{t+u}\right) \tag{2.3.13}$$

$$\mathrm{Vol}(H_1) = \frac{1}{2\pi}\int_{-\infty}^{\infty} dt\, e^{-it}\left(\frac{i\pi}{t}\right)^{N} \tag{2.3.14}$$

dove sull'integrale $\mathrm{Vol}(H_{p,1})$ viene effettuata la sostituzione:

$$\begin{aligned} u = -t - \frac{i}{2\pi} &\Longrightarrow \frac{1}{t+u} = -\frac{2\pi}{i} \\ &\Longrightarrow e^{-iup} = e^{itp}e^{-\frac{p}{2\pi}} \end{aligned} \tag{2.3.15}$$

notando che $p \approx 0 \Rightarrow e^{-\frac{p}{2\pi}} \approx 1$, si ottiene:

$$\mathrm{Vol}(H_{p,1}) = \frac{(i\pi)^N}{2\pi i}\int_{-\infty}^{\infty} dt\, e^{-it(1-p)}\frac{1}{t^{N-1}} \tag{2.3.16}$$

$$\mathrm{Vol}(H_1) = \frac{(i\pi)^N}{2\pi}\int_{-\infty}^{\infty} dt\, \frac{e^{-it}}{t^N} \tag{2.3.17}$$

A questo punto si utilizza il teorema dei residui di Cauchy per calcolare gli integrali (di contorno) in t, i calcoli per esteso e i cammini su cui vengono eseguiti gli integrali si trovano in [13], mentre qui vengono riportati i risultati finali:

$$\mathrm{Vol}(H_{p,1}) = \frac{\pi^N}{(N-2)!}(1-p)^{N-2} \tag{2.3.18}$$

$$\mathrm{Vol}(H_1) = \frac{\pi^N}{(N-1)!} \tag{2.3.19}$$

e quindi la distribuzione che si ottiene dal rapporto tra i due volumi dei sottospazi considerati risulta:

$$\Pr(p) = \frac{\mathrm{Vol}(H_{p,1})}{\mathrm{Vol}(H_1)} = \frac{(N-1)!}{(N-2)!}(1-p)^{N-2} = (N-1)(1-p)^{N-2} \tag{2.3.20}$$

Secondo [13] la distribuzione ottenuta può essere approssimata con la distribuzione di *Porter-Thomas* con grande accordo quando $N \geq 32$, corrispondente a un numero di qubit relativamente esiguo ($n = 5$). Per cui, per N grandi:

$$\Pr(p) \longrightarrow Ne^{-Np} \tag{2.3.21}$$

Questa distribuzione ci permetterà nella sezione successiva di ottenere un'espressione esplicita per $\log \Pr(S)$ e $\log \Pr(S_{cl})$ tramite questo risultato enunciato in [16] e dimostrato in

[13]:

$$\sum_p f(p) = N \cdot \mathbb{E}_U[f(p)] \tag{2.3.22}$$

dove il valore di aspettazione è calcolato sulla distribuzione di *Porter-Thomas*.

2.4 Proprietà di equipartizione asintotica

In questa sezione vengono forniti alcuni strumenti della Teoria dell'Informazione, quali la legge debole dei grandi numeri (*WLLN*, da *weak law of large numbers*) e la proprietà di equipartizione asintotica (*AEP*), con i quali si completerà la dimostrazione della (2.0.1). Nella sezione si farà riferimento alle nozioni illustrate nel libro *Elements of Information Theory* [17].

La *WLLN* stabilisce che per delle variabili x_i indipendenti e identicamente distribuite (*iid*) come $p(x_i)$, la loro media $\langle x_n \rangle = \frac{1}{n}\sum_{i=1}^n x_i$ converge *in probabilità* al loro valore di aspettazione $\mathbb{E}_p[x] = \sum_{i=1}^n x_i p(x_i)$. Formalmente:

$$\langle x_n \rangle \xrightarrow[n\to\infty]{P} \mathbb{E}_p[x] \tag{2.4.1}$$

dove la notazione $\xrightarrow[n\to\infty]{P}$ indica la convergenza *in probabilità* al tendere di n all'infinito, ossia:

$$\lim_{n\to\infty} \Pr(\Big|\langle x_n \rangle - \mathbb{E}_p[x]\Big| > \epsilon) = 0 \tag{2.4.2}$$

per ogni valore di $\epsilon > 0$. Si può trovare una dimostrazione nel Capitolo 4 di [18].

Una diretta conseguenza di questa legge è la proprietà di equipartizione asintotica:

Teorema. *(AEP) se $x_1, x_2, ...$ sono iid $\sim p(x)$, allora*

$$-\frac{1}{m}\log p(x_1, x_2, ..., x_m) \xrightarrow[m\to\infty]{P} H(p(x)) \tag{2.4.3}$$

dove $H(p(x)) = -\sum_{i=1}^N p(x_i)\log p(x_i)$ è l'entropia di Shannon (o entropia dell'informazione) e $N = 2^n$ il numero totale di bitstring possibili.
Da notare che l'argomento del logaritmo nel membro a sinistra della relazione è $\Pr(S)$ e le variabili $x_1, ..., x_m$ sono le misurazioni del campione S:

$$\begin{aligned} -\frac{1}{m}\log p(x_1, x_2, ..., x_m) = -\frac{1}{m}\log \Pr(S) = -\frac{1}{m}\log \prod_{x\in S} p(x) = -\frac{1}{m}\sum_{x\in S}\log p(x) = \\ = -\langle \log p(x) \rangle \end{aligned} \tag{2.4.4}$$

Usando questo risultato, senza perdere di generalità si può dimostrare la *AEP*:

Dimostrazione. (AEP)
Se le x_i sono iid allora anche i $\log p(x_i)$ lo sono. Segue dalla *WLLN* che

$$-\langle \log p(x) \rangle \xrightarrow[m\to\infty]{P} \mathbb{E}_p[-\log p(x)] = H(p(x)) \tag{2.4.5}$$

□

Si ottiene quindi questo risultato utile per quando il numero delle misurazioni per campione è molto grande (si consideri che nell'esecuzione del task, Google ha effettuato misurazioni in numero $\gtrsim 10^6$ per circuito [3]):

$$\log \Pr(S) \approx -mH(p(x)) \tag{2.4.6}$$

Considerando il caso del campione S_{cl} ottenuto tramite simulazione classica il risultato è simile:

$$\log \Pr(S_{cl}) \approx -mH(p_{cl}, p) \tag{2.4.7}$$

dove $H(p_{cl}, p) = \sum_{i=1}^{N} p_{cl}(x_i) \log p(x_i)$ è l'*entropia incrociata* (o *cross-entropy*) e $p_{cl}(x) = \left|\left\langle x \middle| \psi^{cl} \right\rangle\right|^2$ è la distribuzione ottenuta dalla simulazione classica.
Il termine di entropia incrociata compare nel momento in cui, avendo considerato le misurazioni simulate classicamente $x \sim p_{cl}(x)$, si va a calcolare il valore di aspettazione a cui tende la funzione $f(x) = -\langle \log p(x) \rangle$:

$$-\langle \log p(x) \rangle \xrightarrow[m\to\infty]{P} \mathbb{E}_{p_{cl}}[-\log p(x)] \tag{2.4.8}$$

è quindi logico calcolare il valore di aspettazione della funzione $f(x)$ rispetto alla distribuzione $p_{cl}(x)$ dato che le x sono estratte da questa.

2.5 Cross-entropy benchmarking (XEB)

In questa sezione verranno presentati i principali risultati di [13], [14] e [16]. Si completerà la dimostrazione della (2.0.1) attraverso il calcolo esplicito dell'entropia $H(p) = H(p(x))$ e del valore di aspettazione per l'entropia incrociata $\mathbb{E}_U[H(p_{cl}, p)]$. Si utilizza ora il risultato (2.3.22) per calcolare esplicitamente il valore di $H(p)$ [13]:

$$H(p) = -\sum_{i=1}^{N} p(x_i) \log p(x_i) = -N \cdot \int_0^\infty dp\, Npe^{-Np} \log p = \log N - 1 + \gamma \tag{2.5.1}$$

dove γ è la costante di Eulero-Mascheroni. Infine, usando la (2.4.6) si ottiene:

$$\log \Pr(S) = -m(\log N - 1 + \gamma) \tag{2.5.2}$$

In [14] si assume, in linea con gli standard della teoria del caos, che la distribuzione p_{cl} è (quasi) non-correlata alla distribuzione p. L'assunzione è supportata dall'estrema sensibilità alle perturbazione (tipico di sistemi caotici): la distribuzione ottenuta dopo un singolo errore casuale applicato alle porte quantistiche X o Z è *quasi* non-correlata alla distribuzione precedente all'errore $p_U(x)$. Quindi, a meno che un algoritmo classico non utilizzi risorse *esponenziali* in n, il suo *output* è *quasi* statisticamente non-correlato con la distribuzione degli *output* di un circuito quantistico di sufficiente profondità.
Detto ciò, se gli *output* degli algoritmi (quantistico e classico) non sono correlati, si possono considerare p_{cl} e p come variabili casuali indipendenti. Allora, utilizzando la (2.4.7) si calcola il seguente valore di aspettazione sull'insieme di matrici unitarie casuali:

$$\mathbb{E}_U[\log \Pr(S_{cl})] = -m \cdot \mathbb{E}_U[H(p_{cl}, p)] \tag{2.5.3}$$

e, date le precedenti considerazioni:

$$-m \cdot \mathbb{E}_U[-\sum_{i=1}^{N} p_{cl}(x_i) \log p(x_i)] = m \cdot \sum_{i=1}^{N} \mathbb{E}_U[p_{cl}(x_i)] \cdot \mathbb{E}_U[\log p(x_i)] \tag{2.5.4}$$

Il valore di aspettazione di $\log p(x_i)$ è calcolato sulla distribuzione di *Porter-Thomas*:

$$\mathbb{E}_U[\log p(x_i)] = \int_0^\infty dp\, N e^{-Np} \log p = -\log N + \gamma \tag{2.5.5}$$

essendo inoltre l'algoritmo classico non correlato con il circuito quantistico, il valore di aspettazione di p_{cl} coincide con p_{cl} stessa. Allora si ottiene:

$$\mathbb{E}_U[\log \Pr(S_{cl})] = -m(\log N + \gamma) \tag{2.5.6}$$

avendo usato $\sum_{i=1}^{N} p_{cl}(x_i) = 1$.
Unendo i risultati (2.5.2) e (2.5.6) e considerando che essendo $\log \Pr(S) = -m(\log N - 1 + \gamma)$ indipendente da p il suo valore di aspettazione coincide con $\log \Pr(S)$ stesso, si può dimostrare la (2.1.4):

$$\begin{aligned} \mathbb{E}_U[\log \Pr(S) - \log \Pr(S_{cl})] &= \mathbb{E}_U[\log \Pr(S)] - \mathbb{E}_U[\log \Pr(S_{cl})] \simeq \\ &\simeq -m(\log N - 1 + \gamma) + m(\log N + \gamma) = \\ &= m \end{aligned} \tag{2.5.7}$$

Di conseguenza si dimostra anche la (2.0.1):

$$m \approx \frac{\log \Pr(S)}{\log \Pr(S_{cl})} \Rightarrow e^m \approx \frac{\Pr(S)}{\Pr(S_{cl})} = \frac{\prod_{x \in S} |\langle x|\psi\rangle|^2}{\prod_{x \in S_{cl}} |\langle x|\psi\rangle|^2} \tag{2.5.8}$$

Alla fine della Sezione IIa di [14] viene fatto notare che, considerando l'ultimo membro

della (2.5.8), essendo $e^m >> 1$ si ha che $\Pr(S) >> \Pr(S_{cl})$: questo perché il numeratore è *dominato* da misurazioni $x \in S$ con $p(x) > \frac{1}{N}$, mentre il denominatore, essendo formato da misurazioni simulate $x \in S_{cl}$ non correlate con la distribuzione del circuito, è *dominato* dal supporto della distribuzione di *Porter-Thomas* con $p < \frac{1}{N}$.
Boixo et al. continuano notando che:

$$-\frac{\mathbb{E}_U[\Pr(S_{cl})]}{m} = \mathbb{E}_U[H(p_{cl}, p)] = \log N + \gamma \equiv H_0 \tag{2.5.9}$$

è l'entropia incrociata di un algoritmo che campiona uniformemente bitstring casuali (ossia un campionatore da distribuzione uniforme $p_0(x) = 1/N$).
Viene definito quindi un nuovo strumento matematico, la *differenza di entropia incrociata*, o *XED* (da *cross-entropy difference*):

$$\begin{aligned} \Delta H(p_A) &\equiv H_0 - H(p_A, p) = \\ &= \sum_{i=1}^{N} \left(\frac{1}{N} - p_A(x_i) \right) \frac{1}{\log p(x_i)} \end{aligned} \tag{2.5.10}$$

dove p_A è la distribuzione dell'algoritmo A, p è la distribuzione ideale del circuito quantistico. La XED viene proposta come strumento per misurare quanto efficacemente un algoritmo $A(U)$ può simulare l'output di un (tipico) circuito quantistico casuale U. Si nota che la XED è nulla per la distribuzione uniforme e uguale a 1, poiché:

$$\begin{aligned} &\Delta H(p_0) = H(p_0) - H(p_0) = 0 \\ &\Delta H(p) = H(p_0) - H(p) = logN + \gamma - logN - \gamma + 1 = 1 \end{aligned} \tag{2.5.11}$$

Passando dalla teoria ad un apparato sperimentale, si identifica $A = A_{exp}(U)$ e lo si associa alla distribuzione di probabilità p_{exp}. Si definisce la XED sperimentale:

$$\Delta H(p_{exp}) = H_0 - H(p_{exp}, p) \tag{2.5.12}$$

A questo punto si può introdurre la *fidelity* teorica α:

$$\alpha \equiv \mathbb{E}_U[\Delta H(p_{exp})] \tag{2.5.13}$$

Questo parametro è il più importante nell'esperimento realizzato da Google, in quanto determina l'accuratezza con la quale il computer quantistico opera: richiamando le equazioni (2.5.11), per un computer quantistico perfetto si ottiene $\alpha = 1$, mentre per uno difettoso si ha $\alpha = 0$. Applicando il teorema del limite centrale alla (2.5.10) si ottiene [14]:

$$\alpha \approx H_0 - \frac{1}{m} \sum_{j=1}^{m} \log \frac{1}{p(x_j^{exp})} \tag{2.5.14}$$

dove m è la dimensione del campione sperimentale $S_{exp} = \{x_1^{exp}, ..., x_m^{exp}\}$. La stima della performance dell'algoritmo è chiamata *cross-entropy benchmarking* (*XEB*) e il procedimento è il seguente [14]:

1. Viene campionato dall'insieme universale di porte quantistiche un circuito casuale U.
2. Si raccoglie un campione di bitstring $S_{exp} = \{x_1^{exp}, ..., x_m^{exp}\}$ nella base computazionale, con m sufficientemente grande ($\sim 10^6$).
3. Si calcolano le grandezze $\log \frac{1}{p(x_j^{exp})}$ su un supercomputer classico.
4. Si stima il parametro α con l'equazione (2.5.14).

Si parla di supremazia quantistica quando:

$$C < \alpha \leq 1 \tag{2.5.15}$$

dove $C = \mathbb{E}_U[\Delta H(p^*)]$, con p^* distribuzione degli output del miglior algoritmo A^* eseguibile su un classico (super)computer.
Il concetto stesso di supremazia quantistica è quindi strettamente legato alla potenza degli algoritmi e dei computer classici: se questi hanno abbastanza risorse da simulare perfettamente un circuito quantistico allora $C = 1$ e la supremazia quantistica è impossibile. Nello specifico, nel caso di un numero di qubit $n \lesssim 48$ o di circuiti con profondità d non sufficiente la simulazione diretta è possibile e $C = 1$.
Infatti, allo stato attuale (2023) [19], il supercomputer *pre-exascale "Leonardo"*, presso il Tecnopolo di Bologna, ha a disposizione un modulo di calcolo (*modulo Booster*) progettato per massimizzare la capacità computazionale con una memoria RAM di 2.654 Petabytes (PB) (il modulo è formato da 3456 nodi ciascuno dei quali possiede una RAM DDR4 da 512GB e 4 schede grafiche NVidia da 64GB). Considerando che per memorizzare nella memoria RAM una funzione d'onda in *single-digit precision* occorrono $2^n \cdot 2 \cdot 4 = 2^{n+3}$ bytes. Per $n = 48$ occorreranno quindi $2^{51} \simeq 2.252$ PB, che il supercomputer considerato copre con buono scarto.
Per circuiti $n \gtrsim 48$ e con profondità sufficiente, il calcolo numerico delle $p_U(x), x \in S_{exp}$ non è più possibile e $C \simeq 0$, rendendo possibile la supremazia quantistica. Tuttavia, l'impossibilità di calcolare queste quantità impedisce la stima diretta del parametro α, che *Boixo et al.* sostengono potrà essere estrapolato da circuiti specifici meno grandi e che, nella pratica, bisognerà tenere conto di possibili *bias* sperimentali nella scelta dei circuiti usati per l'estrapolazione.

Capitolo 3

Supremazia quantistica su Sycamore-53

In questo capitolo verrà illustrato il primo esperimento eseguito con il fine di mostrare la supremazia quantistica, realizzato dalla divisione di Intelligenza Artificiale di Google nel 2019. La maggior parte dei risultati presentati di seguito sono estratti quindi dall'articolo pubblicato dal team di Google [3].

L'esperimento consiste nel campionare l'output di un circuito quantistico pseudo-casuale (ossia il *Random Circuit Sampling* descritto nel capitolo 2) con il processore quantistico progettato e costruito dal team di Google e con un supercomputer (in questo caso è stato utilizzato Summit, che al tempo era il super-computer più potente al mondo [3]), per poi confrontarne i risultati.

3.1 Linear Cross-Entropy Benchmarking Fidelity

L'obiettivo dell'esperimento è quello di ottenere un valore della *Linear Cross-Entropy Benchmarking Fidelity* (in simboli $\mathcal{F}_{XEB}$, chiamata anche solo *fidelity*) abbastanza alto da non essere replicabile in tempi e risorse accettabili tramite un super-computer. La $\mathcal{F}_{XEB}$ viene calcolata a partire dalle bitstring misurate $\{x_i\}$ ed è espressa come [3]:

$$\mathcal{F}_{XEB} \equiv 2^n \langle P(x_i) \rangle_i - 1 \tag{3.1.1}$$

dove n è il numero dei qubit, $P(x_i)$ è la probabilità di ottenere la bitstring x_i.
La fidelity $\mathcal{F}_{XEB}$ (o anche α_f in [14]) è derivata in [14] e in [3] (Supplementary Information) a partire da un modello per il rumore quantistico chiamato *depolarizing channel* (canale di depolarizzazione) [3]:

$$\rho_U = \alpha_f |\psi_U\rangle \langle\psi_U| + (1 - \alpha_f)\frac{\mathbb{1}}{2^n} \tag{3.1.2}$$

dove α_f è la fidelity sperimentale, n è il numero di qubit, $|\psi_U\rangle = U|\psi_0\rangle$ è il sistema evoluto tramite il circuito U e $|\psi_0\rangle$ è lo stato iniziale del sistema (che di solito viene inizializzato a $|\psi_0\rangle = |0\rangle^{\otimes n} = |0 \dots 0\rangle$). La (3.1.2) descrive un modello teorico secondo il quale il canale quantistico rumoroso trasforma lo stato iniziale del sistema $\rho_0 = |\psi_0\rangle \langle\psi_0|$ in uno stato composto da due fattori: quello dello stato corrispondente all'evoluzione esatta $|\psi_U\rangle$ con probabilità α_f e quello dello stato massimamente misto $\mathbb{1}/2^n$ con probabilità $(1 - \alpha_f)$.

Riprendendo la $p_{exp}(x_j)$ della Sezione 2.5, in [14] viene descritta come:

$$p_{exp}(x_j) = \left\langle x_j \right| \rho_U \left| x_j \right\rangle \tag{3.1.3}$$

da cui, inserendo l'espressione di ρ_U vista in (3.1.2):

$$p_{exp}(x_j) = \alpha_f p(x_j) + (1 - \alpha_f)\frac{1}{N} \tag{3.1.4}$$

dove si è posto $N = 2^n$. Adesso quindi si può calcolare esplicitamente l'entropia incrociata sperimentale dell'equazione (2.5.12):

$$\begin{aligned} H(p_{exp}, p) &= \sum_{j=1}^{m} \left(\alpha_f p(x_j) + (1 - \alpha_f)\frac{1}{N} \right) \frac{1}{\log p(x_j)} \\ &= \alpha_f \sum_{j=1}^{m} p(x_j) \frac{1}{\log p(x_j)} + (1 - \alpha_f) \sum_{j=1}^{m} \frac{1}{N} \frac{1}{\log p(x_j)} = \\ &= \alpha_f H(p) + (1 - \alpha_f) H_0 \end{aligned} \tag{3.1.5}$$

dove H_0 si ricorda essere l'entropia incrociata descritta in (2.5.9), $x_j \in S_{exp}$ sono le bitstring misurate nell'esperimento e $\left|S_{exp}\right| = m$ è la dimensione del campione sperimentale. Il valore atteso di questa entropia incrociata sperimentale è:

$$\mathbb{E}_U[H(p_{exp}, p)] = \alpha_f(\mathbb{E}_U[H(p)] - H_0) + H_0 \tag{3.1.6}$$

Calcolando il valore atteso della XED sperimentale (2.5.12) si ottiene un'espressione per la fidelity sperimentale α_f:

$$\begin{aligned} \mathbb{E}_U[\Delta H(p_{exp}] &= \mathbb{E}_U[H_0 - H(p_{exp}, p)] = \\ &= H_0 - \alpha_f(\mathbb{E}_U[H(p)] - H_0) - H_0 = \alpha_f(H_0 - \mathbb{E}_U[H(p)]) \end{aligned} \tag{3.1.7}$$

da cui:

$$\alpha_f = \frac{H_0 - \mathbb{E}_U[H(p_{exp}, p)]}{H_0 - \mathbb{E}_U[H(p)]} \tag{3.1.8}$$

Utilizzando l'approssimazione resa possibile dalla distribuzione di Porter-Thomas (Sezione 2.5) si ottiene un'espressione calcolabile per la (3.1.8) [16]:

$$\alpha_f \approx \log N + \gamma - \mathbb{E}_U[H(p_{exp}, p)] \tag{3.1.9}$$

dove γ è la costante di Eulero-Mascheroni. Ricordando che $H_0 = \log N + \gamma$ e usando la linearità del valore atteso:

$$\begin{aligned} H_0 - \mathbb{E}_U[H(p_{exp}, p)] &= \mathbb{E}_U[H_0 - H(p_{exp}, p)] = \\ &= \mathbb{E}_U[\Delta H(p_{exp})] \equiv \alpha \end{aligned} \tag{3.1.10}$$

dove è stata usata la definizione di fidelity teorica (2.5.13). Quindi è dimostrato che la fidelity sperimentale α_f è uguale in approssimazione alla fidelity teorica α e che quindi può essere calcolata come descritto in Sezione 2.5, ossia calcolando le quantità $\log \frac{1}{p(x_j^{exp})}$ (si veda (2.5.14)).

Nella (3.1.1) viene riportata l'espressione della fidelity lineare, dove le quantità da calcolare sono solo le probabilità $p(x_j^{exp})$ [3].

3.2 Calibrazione di Sycamore

In questa sezione vengono esposti i dettagli del processore quantistico (anche denotato come QPU, ossia *Quantum Processing Unit*, analogo della CPU dei computer classici) Sycamore e dei circuiti utilizzati nell'esperimento. Verranno forniti alcuni dettagli relativi alle fasi precedenti al sampling degli output dei circuiti quantistici casuali.
La maggior parte del contenuto della sezione sarà quindi estrapolato dall'articolo pubblicato da *Google AI Quantum and collaborators* [3] e dalle relative *Supplementary information*.

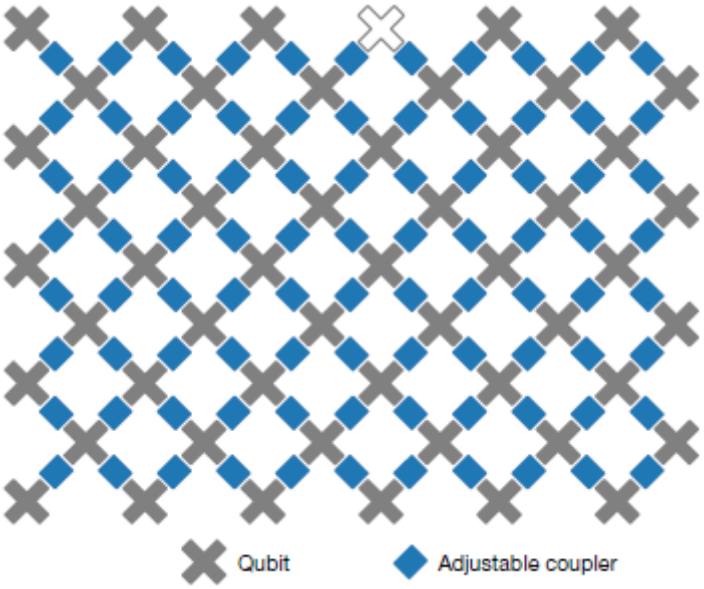


Fig. 3.1: Schema dei qubit nel processore Sycamore. [3]

Sycamore è un processore quantistico formato da 142 *transmon* qubit (descritti in appendice A), dei quali 54 possono essere regolati singolarmente in frequenza e impulso ricevuto e possono essere misurati individualmente e sono i "veri e propri" qubit del processore. Gli altri 88 qubit agiscono come *accoppiatori* (*couplers*, informazioni dettagliate in [12]), fissati al ground state durante il calcolo. I qubit sono connessi tra loro tramite i coupler in una griglia diagonale 2D. I couplers permettono di effettuare porte a 2 qubit (come il CZ definito in equazione (1.2.8)). Nel task eseguito uno dei 54 qubit è risultato

Pauli and measurement errors

Average error	Isolated	Simultaneous
Single-qubit (e_1)	0.15%	0.16%
Two-qubit (e_2)	0.36%	0.62%
Two-qubit, cycle (e_{2c})	0.65%	0.93%
Readout (e_r)	3.1%	3.8%

Fig. 3.2: Errori di Pauli e di misura. Nella prima colonna ci sono i tipi di errori come descritti nel testo; nelle ultime due colonne ci sono i valori relativi all'operazione di benchmarking sui qubit isolati o simultanea. [3].

difettoso e quindi il dispositivo ha usato effettivamente 53 qubit e 86 couplers (il qubit mal funzionante è colorato in bianco in Fig. 3.1).

Prima di iniziare l'esperimento si effettua un benchmark della performance delle porte quantiche (realizzate come impulsi di microonde della durata di $25ns$) a uno e due qubit per misurare la probabilità di errore durante l'esecuzione delle operazioni.

Per quanto riguarda le operazioni a singolo qubit, su ognuno di questi si applicano singolarmente m porte quantiche casualmente scelte e si calcola la fidelity media (la stessa dell'equazione (3.1.1) con $n = 1$). Al crescere di m la fidelity cala con la legge:

$$p = \left[1 - \frac{e_1}{1 - \frac{1}{2(2n)}}\right]^m \tag{3.2.1}$$

dove p è l'effettiva polarizzazione dello stato del qubit ([3], *Supplementary Information*), e_1 è la probabilità di un errore di Pauli (descritti in Sezione 1.2). Viene fatto presente che il parametro p è identificabile con l'effettiva fidelity del circuito in esame (α_f). Una seconda azione di benchmarking sui gate a singolo qubit viene eseguita simultaneamente su tutti i qubit, con risultati lievemente peggiori rispetto alla versione isolata del task. I valori dei Pauli error $e1$ vengono ricavati quindi dal *decay* della fidelity (numeri in tabella in Fig. 3.2).

Per effettuare il benchmarking delle porte a due qubit, questi vanno prima posti in uno stato di entanglement tramite le porte quantiche fSim, applicate a qubit connessi allo stesso coupler, attivato per brevissimo tempo ($12ns$) per l'occasione. Si eseguono quindi m *cicli* di circuiti a due qubit: con ciclo si intende l'esecuzione in sequenza di una singola porta quantica casuale per ogni qubit e una porta a due qubit fissata. Si confrontano poi le probabilità stimate con quelle ideali. I dati della XEB fidelity vengono quindi ottimizzati [3], ricavando così gli errori per ciclo e_{2c} riportati alla terza riga nella tabella in Fig. 3.2. Anche in questo caso viene ripetuto l'esperimento in simultanea su tutti i qubit del processore.

Gli errori sulle singole porte a due qubit e_2 sono estratti sottraendo a e_{2c} due volte (poiché in ogni ciclo si eseguono due porte a un qubit, una per ciascun qubit) gli errori delle porte a un qubit e_1, ottenendo i risultati riportati alla seconda riga della tabella in Fig. 3.2.

Viene infine effettuato il benchmarking sulla lettura finale dei qubit, singolarmente e simultaneamente (risultati nell'ultima riga della tabella in Fig. 3.2). Una tabella completa dei parametri del processore preparato per l'esperimento di RCS (quali frequenza dei qubit, anarmonicità, errori) si trova a pagina 22 ("TABLE II. Aggregate system parameters") di *Supplementary Information for Quantum supremacy using a programmable superconducting processor* [3].

3.3 Modello sperimentale

Il task del campionamento di circuiti quantistici casuali (RQCs, da *random quantum circuits*) è modellato per soddisfare due obiettivi:

P.1 Dimostrare la supremazia quantistica effettuando un calcolo non replicabile da un super-computer classico.

P.2 Valutare la performance di Sycamore attraverso il calcolo della fidelity.

Una delle maggiori difficoltà dell'esperimento è data dal contrasto tra **P.1** e **P.2**. Infatti, per dimostrare la supremazia quantistica (**P.1**) occorre che il campionamento effettuato dal quantum-computer Sycamore, con una certa fidelity, non sia replicabile (con la stessa fidelity) in tempi adeguati da un super-computer classico. Il calcolo della fidelity (**P.2**) va però effettuato classicamente, raccogliendo le ampiezze esatte dalla funzione d'onda in output $|\psi_U\rangle$: le risorse computazionali necessarie al calcolo della fidelity scalano con il numero di qubit n e con la profondità dei circuiti m e possono arrivare anche nell'ordine di milioni di anni [3] per il tempo e (tenendo presente le considerazioni a fine capitolo 2) a $2^{56} \approx 72$ PB di memoria RAM, molto lontana dalla capacità RAM di un super-computer come ad esempio *Leonardo* (2.654 PB [19]).

La soluzione a questo problema è realizzare delle *varianti* semplificate dei circuiti utilizzati per **P.1**, sulle quali è possibile **P.2**. Le varianti semplificate dei circuiti originali (anche detti *interi*) sono simulabili classicamente ed è possibile calcolare la loro fidelity. Le fidelity calcolate vengono poi confrontate con la previsione teorica, ottenuta moltiplicando le probabilità di assenza di errori delle porte quantistiche a uno e a due bit e di errori di misura sulla lettura dei qubit [3]:

$$F \approx \prod_{g \in G_1} (1 - e_g) \prod_{g' \in G_2} (1 - e_{g'}) \prod_{q \in Q} (1 - e_q) \tag{3.3.1}$$

dove G_1, G_2, Q sono rispettivamente gli insiemi (relativi ai circuiti interi) delle porte quantiche a singolo qubit, a doppio qubit e l'insieme dei qubit mentre $e_g, e_{g'}, e_q$ sono rispettivamente gli errori di Pauli individuali delle porte a un qubit, a due qubit e l'errore sulla preparazione dello stato e/o sulla misura dei qubit singoli.

L'esperimento quindi può essere schematizzato nelle seguenti fasi:

Fase 1: circuiti verificabili. I circuiti in questa fase hanno un numero di qubit (la larghezza o *width* del circuito) in input variabile da $n = 12$ a $n = 53$ e numero di cicli (la profondità o *depth* del circuito) fissato a $m = 14$. La sequenza di esecuzione dei cicli è *semplificata* (si veda Sezione 3.4) e per cui anche i circuiti interi in questa fase prendono il nome di circuiti interi semplificati.

- **CV1.** Viene eseguito il campionamento sui circuiti interi semplificati su Sycamore.
- **CV2.** Viene eseguito il campionamento sulle varianti dei circuiti interi semplificati su Sycamore.
- **CV3.** Per ogni circuito intero semplificato viene calcolata la fidelity prevista (3.3.1).
- **CV4.** Vengono realizzate simulazioni classiche dei circuiti interi semplificati e delle sue varianti per calcolare la fidelity di ognuno di questi tramite un super-computer.
- **CV5.** Si confrontano le fidelity ottenute sia tra i circuiti interi semplificati e le sue varianti, sia con la previsione teorica.

Dato che l'accordo tra le fidelity dei circuiti interi semplificati e delle sue varianti è buono (tutte in linea con la previsione teorica), si può ragionevolmente pensare che rimanga vero anche per circuiti più complessi, così da utilizzare i risultati delle varianti dei circuiti interi per stimare le fidelity dei circuiti interi (non semplificati).

Fase 2: circuiti per la supremazia. In questa fase i circuiti hanno un numero di qubit in input fisso a $n = 53$ e una profondità che va da un minimo di $m = 12$ a un massimo di $m = 20$. Viene rimossa la semplificazione utilizzata nella fase precedente.

- **CS1.** Viene eseguito il campionamento sui circuiti interi su Sycamore.
- **CS2.** Viene eseguito il campionamento sulle varianti dei circuiti interi su Sycamore.
- **CS3.** Si calcola la fidelity prevista dalla (3.3.1) per i circuiti interi.
- **CS4.** Si realizzano simulazioni classiche solo delle varianti dei circuiti interi (poiché questi ultimi non sono simulabili classicamente in tempi adeguati) per valutarne la fidelity su un super-computer.
- **CS5.** Si confrontano le fidelity ottenute e in base a quanto osservato nella fase precedente, si stima la fidelity dei circuiti interi.

I dati ottenuti al punto **CS1.** sono stati archiviati poiché la verifica della fidelity avrebbe richiesto troppo tempo. Sono stati poi estrapolati dei tempi di simulazione classica dell'esperimento di sampling per confrontarli con il tempo impiegato da Sycamore al fine di dichiarare raggiunta la supremazia quantistica [3].

3.4 Circuiti dell'esperimento

I circuiti dell'esperimento sono formati da m cicli, ognuno dei quali è diviso in due fasi: una prima in cui ad ogni qubit viene applicata una porta singola casuale dall'insieme $\{\sqrt{X}, \sqrt{Y}, \sqrt{W}\}$ ed una seconda in cui viene applicata una porta doppia su alcune coppie di qubit. Per ognuna delle coppie di qubit collegate da un coupler (86 in totale), la porta doppia è fissata ed appartiene a una famiglia di porte decomponibili in quattro porte Z e una $fSim(\theta, \phi)$ (descritta in equazione (1.2.10)) con parametri θ e ϕ fissati inizialmente a $\theta \approx \frac{\pi}{2}$ e $\phi \approx \frac{\pi}{6}$, poi corretti usando il XEB su ogni coppia di qubit presa in esame per minimizzare gli errori [3]. È doveroso notare che per **P.2** è necessario conoscere l'entropia incrociata della distribuzione ideale degli output del circuito, ricavata in Sezione 2.5 per circuiti con misure di probabilità distribuite secondo la distribuzione di Porter-Thomas. In [3] viene fatto presente che i loro circuiti quantistici casuali soddisfano il requisito quando la profondità del circuito $m > 12$, per cui tutti i circuiti dell'esperimento hanno questo limite inferiore alla profondità.

I circuiti vengono realizzati secondo i criteri **P.1** e **P.2**: quelli implementati per soddisfare la prima prendono il nome di circuiti per la supremazia, mentre quelli che soddisfano la seconda sono i circuiti verificabili.

Circuiti per la supremazia. I circuiti di questa categoria sono realizzati appositamente per non essere replicabili da un super-computer. Si tratta di circuiti con $n = 53$ qubit e profondità variabile tra $m = 12$ e $m = 20$. Una caratteristica che li differenzia dai circuiti verificabili è il *pattern* di esecuzione delle porte doppie. I coupler attorno ad un generico qubit di questi circuiti sono divisi in quattro gruppi A,B,C,D e vengono attivati secondo la sequenza di 8 cicli ABCDCDAB. La sequenza da 8 cicli è modellata per aumentare l'entanglement del circuito. Dopo gli m cicli è presente un mezzo ciclo costituito solo da porte singole prima della misurazione simultanea dei qubit. In Fig. 3.3 è schematizzato quanto descritto sopra. Il circuito più grande su cui l'esperimento di sampling è stato effettuato (e utilizzato ai fini di **P.1**) è formato da $n = 53$ qubit, 1113 porte singole, 430 porte doppie, dispositivo di lettura per ogni qubit e $m = 20$ cicli. Su questo circuito, la fidelity prevista secondo il modello (3.3.1) era dello 0.2%. Sebbene sia un valore apparentemente piccolo bisogna considerare che una singola implementazione di una porta quantistica ha una fidelity $\approx 99\%$; ad esempio, per citare uno studio dello stesso anno della pubblicazione dell'articolo di Google (2019), in *Realisation of high-fidelity*

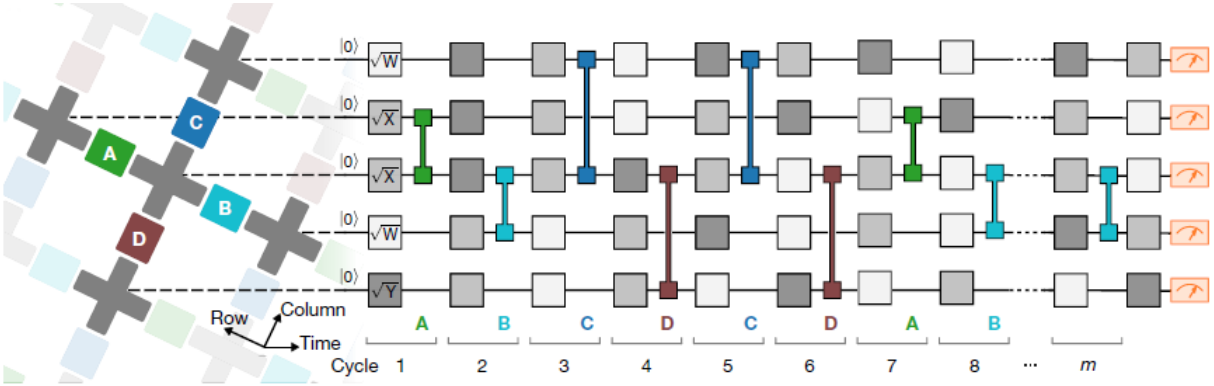


Fig. 3.3: Struttura di un circuito per la supremazia. Le porte singole sono scelte casualmente da $\{\sqrt{X}, \sqrt{Y}, \sqrt{W}\}$, mentre le porte doppie dipendono esclusivamente dal coupler (quindi dalla coppia di qubit specifica) e dal ciclo. La sequenza di attivazione dei coupler è ABCDCDAB. [3]

.

nonadiabatic CZ gates with superconducting qubits [20] viene realizzata una porta CZ con la fidelity massima raggiunta del $(99.54 \pm 0.08)\,\%$. Quindi, considerando il gran numero di porte utilizzate nel circuito più grande ($\sim$ 1500 porte), dopo un grezzo calcolo si trova $0.99^{1500} \approx 0.00003\%$ da cui si può apprezzare la fidelity raggiunta nell'esperimento trattato.

Circuiti verificabili. Questa categoria è composta da circuiti simulabili classicamente in tempi adeguati. Il numero dei qubit utilizzati varia da un minimo di $n = 12$ ad un massimo di $n = 53$, mentre la profondità è fissata a $m = 14$. A differenza dei circuiti per la supremazia, il *tiling* (ossia il *piastrellamento*) dei coupler attorno ad un generico qubit è semplificato: questi sono divisi nei quattro gruppi E,F,G,H e vengono attivati nella sequenza di 4 cicli EFGH. La sequenza è è implementata per permettere al circuito di formare delle strutture tra qubit (*wedges*, discussi più avanti nella sezione) adatte a rallentare la diffusione dell'entanglement all'interno.

3.4.1 Varianti dei circuiti interi

Nella Sezione 3.3 è stato introdotto il concetto di varianti per i circuiti. Le varianti prendono il nome di *elided circuit* e *patch circuit*. La ratio con cui sono state realizzate queste varianti è quella di renderle simulabili attraverso l'algoritmo di *Schrödinger-Feynman* (*SFA*). Il SFA è un ibrido tra la *full state vector simulation* (o algoritmo di Schrödinger (SA)) e la *tensor network simulation* (o algoritmo di Feynman) i quali hanno difficoltà ad essere eseguiti quando n, m crescono, rispettivamente [3]. Il risultato è quindi un algoritmo che lavora bene su circuiti a grande profondità e con molti qubit in questo modo: riceve come input un circuito a n-qubit e un *taglio* (*cut*) che divide i qubit in input in

due partizioni adiacenti con n_1 e n_2 qubit, $n_1 + n_2 = n$. La scelta del cut è fondamentale poiché esclusivamente da questa dipende il tempo di esecuzione (*runtime*) dell'algoritmo di Schrödinger-Feynman [3]:

$$T_{SFA} \sim (2^{n_1} + 2^{n_2})r^g \tag{3.4.1}$$

dove g è il numero delle porte che attraversano le partizioni (*cross-partition gates*) e r è il *rango di Schmidt*, indice della non separabilità di uno stato [21]. Il fattore r^g è denominato in [3] *bond dimension* dei tagli, associato alla partizione dei circuiti. I tagli con n_1, n_2, g tali che la simulazione è realizzabile sono detti *promising cuts*, tra i quali quello che nell'esperimento ha dato i risultati migliori e che quindi è stato scelto per realizzare le varianti dei circuiti è stato quello parallelo all'asse più corto del reticolo di qubit, nei pressi del qubit malfunzionante (nel riquadro (a) di Fig. 3.4, il taglio è indicato come una linea tratteggiata nello schema del *patch circuit*). I coupler cross-partition lungo questo taglio sono 7. È chiaro dalla (3.4.1) che il runtime della simulazione aumenta esponenzialmente con il numero delle porte cross-partition, che quindi nelle varianti ideate per stimare la fidelity dei circuiti non riproducibili classicamente giocano un ruolo fondamentale. I dettagli dell'algoritmo SFA si trovano in [22].

Le due trasformazioni utilizzate per ridurre il costo della simulazione (attraverso la riduzione della bond dimension) sono [3]:

- *Gate elision* (Rimozione porte). Questa operazione consiste nel rimuovere un certo numero di porte cross-partition lungo il promising cut scelto. Ogni porta rimossa riduce la bond dimension di un fattore 2 o 4 a seconda di quanti termini sono presenti nella decomposizione di Schmidt (ossia il rango) della porta , solitamente 4. I circuiti con alcune di queste porte rimosse sono gli *elided circuits*, mentre quelli con tutte le porte cross-partition rimosse sono i *patch circuits*. Questi ultimi diventano due circuiti separati e non interagenti (le due partizioni) che vengono eseguiti in parallelo. I circuiti iniziali (senza nessuna rimozione) sono detti *full circuits*.

- *Wedge formation* (Formazione di cunei). Un cuneo, o *wedge*, è una struttura formata da 2 porte doppie consecutive che hanno in comune un qubit. La decomposizione di Schmidt della matrice unitaria connessa all'operazione sui 3 qubit ha 4 termini, contribuendo quindi con un fattore 4 alla bond dimension, invece che 8 come nel caso di due porte doppie separate. Questo riduce l'entanglement tra partizioni e rende possibile la simulazione tramite SFA. Come già accennato in precedenza, la sequenza EFGH favorisce la formazione dei cunei e ai circuiti modellati su questa viene aggiunto il suffisso *simplified* (semplificato) nel nome. Da notare che i circuiti semplificati fanno parte della categoria dei circuiti verificabili, ossia simulabili classicamente.

Riassumendo quanto sopra: dato un circuito intero, rimuovendo alcune porte doppie lungo il promising cut si ottiene un circuito elided, mentre rimuovendo tutte le porte

doppie lungo il taglio si ottiene un circuito patch. In aggiunta, è possibile rendere i circuiti "semplificati" tramite il pattern di attivazione dei coupler EFGH, così da ottenere i circuiti interi semplificati, dai quali applicando la rimozione delle porte appena descritta si ottengono i circuiti patch semplificati e elided semplificati.

L'accordo sulla fidelity tra la versione intera del circuito e la versione patch presenta una deviazione minima ($\sim 5\%$), dovuta al mancato entanglement tra le due partizioni del circuito patch. Il vantaggio dei circuiti patch risiede nel minore costo computazionale rispetto alla versione del circuito intero: il XEB viene operato su due sistemi non interagenti (le partizioni del circuito patch) con una dimensione pari alla metà del circuito intero. Per questo motivo vengono utilizzati come un veloce indicatore della performance del sistema e per monitorare la stabilità del sistema [3].

I circuiti elided invece sono una versione più sofisticata rispetto a quelli patch: le due partizioni non sono più totalmente isolate, ma data la rimozione di alcune porte lungo il promising cut del circuito la diffusione dell'entanglement è rallentata. Le due partizioni sono quindi simulate tramite il metodo di Schrödinger e poi vengono connesse con il criterio degli integrali sui cammini di Feynman. Il rapporto medio tra le fidelity dei circuiti elided e quella dei circuiti interi è risultato essere di 1.01, dimostrando un perfetto accordo tra le due varianti [3].

3.5 RCS su Sycamore

In questa sezione vengone le operazioni dell'esperimento ed i suoi risultati. Lo schema del procedimento è descritto in Sezione 3.3.

3.5.1 Campionamento dei circuiti verificabili

La prima fase consiste nel generare dei circuiti verificabili (si veda sezione 3.4, in modo da mostrare l'accordo sulla fidelity tra le varianti proposte (patch, elided, intero semplificato (*full simplified* in Fig. 3.4)). Vengono quindi generati 10 circuiti casuali sul processore quantistico con un numero di qubit da $n = 12$ a $n = 53$ e con la profondità fissata a $m = 14$, per poi campionare 5×10^5 bitstring in uscita da ogni circuito, per un totale di $N_s = 5 \times 10^6$ bitstring totali. Si ricorda che i circuiti verificabili presentano un pattern di porte doppie EFGH (si veda la sezione 3.4). Il XEB viene eseguito su diversi dispositivi e con diverse modalità a seconda della grandezza (il numero dei qubit n) dei circuiti. Per i sistemi da 12 a 37 qubit viene utilizzato l'algoritmo di Schrödinger su un server con 88 *hyper-thread* (CPU sviluppata da Intel per il multithreading [23], tecnica per eseguire più operazioni in simultanea su un processore) e 1.5TB di memoria. Per i circuiti a 38 qubit la full state vector simulation viene realizzata da un *n1- ultramem-160 VM* (potente macchina di Google [24]) sul cloud di Google, con 160 hyper-thread e 3.8TB di memoria.

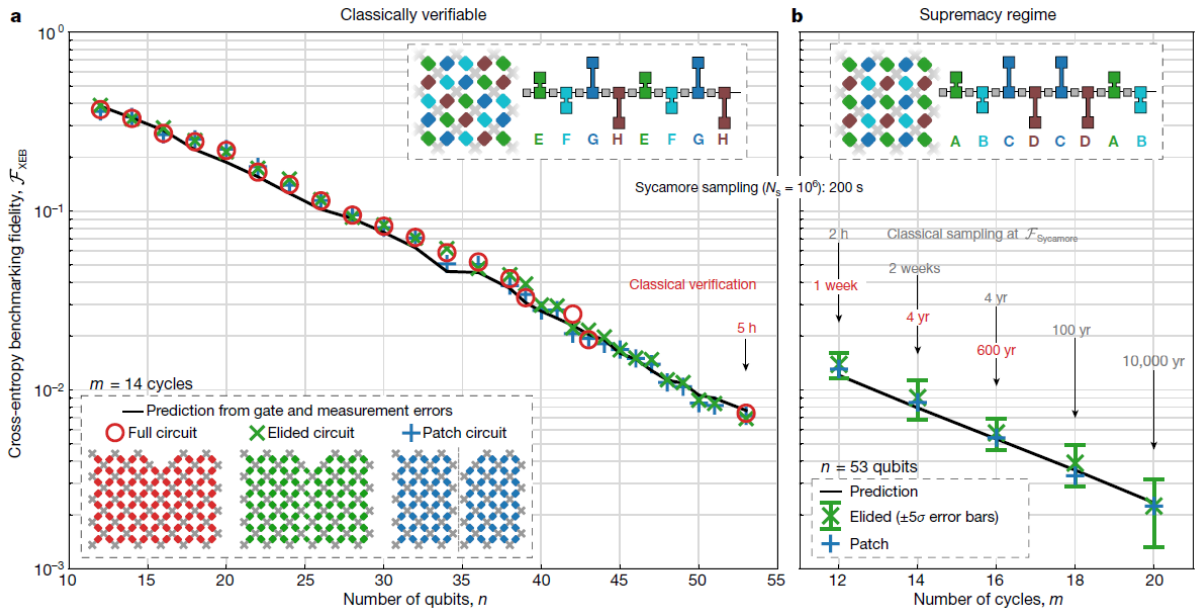


Fig. 3.4: Risultati della stima della $\mathcal{F}_{XEB}$. In (a) i circuiti sono semplificati (pattern EFGH) e la profondità è ridotta a $m = 14$, mentre in (b) i circuiti sono in regime di supremazia quantistica. In rosso sono indicati i tempi di analisi del XEB per i circuiti eseguiti su Sycamore. In grigio sono riportati i tempi estrapolati per una eventuale simulazione classica. In nero viene tracciata la previsione teorica della fidelity ottenuta tramite l'equazione (3.3.1). [3]

L'algoritmo di Schrödinger viene sostituito dallo SFA superati i 38 qubit, utilizzato sul *large-scale cluster computing* di Google [3].

I risultati sono riassunti in Fig. 3.4: i valori della fidelity ottenuta dai circuiti verificabili sono stati comparati alla fidelity prevista (3.3.1) e sono riportati visivamente (Fig. 3.4, linea nera). Da notare come nonostante la previsione teorica della (3.3.1) sia relativamente semplice riesca a predire i risultati sperimentali con sorprendente precisione. Nello specifico, i dati dei circuiti interi semplificati sono riportati in Fig. 3.4 da cerchi in rosso, quelli dei circuiti elided sono croci colorati in verde mentre i risultati dei circuiti patch sono indicati da croci blu.

Come previsto, i tempi per il calcolo della fidelity sono risultati accettabili, con l'analisi del XEB per il circuito intero semplificato più grande ($n = 53$ e $m = 14$) che ha impiegato 5h (mostrato come *classical verification* in Fig. 3.4 e indicato anche come un cerchio rosso in 3.6(b)).

L'osservazione più importante di questa parte dell'esperimento è che i risultati dei circuiti interi semplificati e delle sue varianti sono molto simili: è giustificabile quindi usare i risultati dei circuiti elided e patch per trarre conclusioni sui circuiti interi anche in regime di supremazia quantistica.

3.5.2 Campionamento dei circuiti per la supremazia

Dopo aver verificato che le probabilità $p(x_i)$ delle bitstring x_i ottenute dai circuiti fossero distribuite secondo la distribuzione di Porter-Thomas (si veda 2.3) tramite la statistica di *Kolmogorov-Smirnov* (descritta in [3]), si procede all'esperimento sui circuiti in regime di supremazia quantistica ($n = 53$, $m = 20$). Vengono generati 10 circuiti casuali con rimozione di porte (patch, elided) poiché il XEB sui circuiti interi di queste dimensioni è proibitivo. Sono state campionate $N_s = 3 \times 10^6$ bitstring da ciascuno dei circuiti (per un ammontare di 30×10^6 bitstring) e sono state calcolate le probabilità p_i per ciascuna bitstring x_i (ottenuta dai circuiti patch, elided). In Fig. 3.4(b) sono illustrati i dati relativi ai circuiti per la supremazia: come per i valori dei circuiti verificabili, i risultati delle fidelity dei circuiti elided sono disegnati in verde mentre quelli dei circuiti patch in blu. Non sono presenti nel grafico i risultati dei circuiti interi, essendo stati archiviati per l'impossibilità di analisi XEB data l'intrattabile mole di dati. Quindi questi dati sono stati dichiarati essere in regime di supremazia quantistica. I cerchi rossi in Fig. 3.4(a) sono stati sostituiti da scritte in rosso in Fig 3.4(b), le quali indicano i tempi di analisi XEB ottenuti tramite un'estrapolazione e necessari a ricavare un risultato per la fidelity: questi vanno da 1 settimana nel caso del circuito per la supremazia più piccolo ($n = 53$, $m = 12$), a 600 anni per il circuito per la supremazia con $n = 53$ e $m = 16$, fino a milioni di anni per il circuito per la supremazia più grande ($n = 53$, $m = 20$). Quest'ultimo valore non è indicato nelle figure ma è solamente citato in [3]. I risultati estrapolati di questi circuiti intrattabili classicamente sono anche illustrati in Fig. 3.6(c), indicati come stelline rosse. La fidelity media calcolata per i circuiti elided, dopo l'analisi delle incertezze statistiche e sistematiche, è [3]:

$$\mathcal{F}_{XEB} = (2.24 \pm 0.21) \times 10^{-3} \tag{3.5.1}$$

Dato quindi l'ottimo accordo (osservabile in Fig. 3.4) tra i risultati della fidelity per le varianti dei circuiti e i circuiti interi semplificati, è plausibile aspettarsi che anche i circuiti interi non semplificati abbiano risultati simili.

Tramite un cluster di computer di Google è stato estrapolato un tempo di simulazione dell'esperimento di sampling per un circuito intero da $n = 53$ qubit con $m = 20$ cicli pari a circa 10000 anni (testo grigio in Fig. 3.4).

In Fig. 3.5 sono mostrati altri runtime per vari circuiti (corrispondenti alle scritte in grigio della Fig. 3.4).

Null hypotesis. È stato stimato dagli autori di [3] che un computer classico con una CPU, la cui potenza è equivalente a 1 milione di core, necessita di 5000 anni per campionare l'output (1 milione di volte) di un circuito quantistico random in regime di supremazia quantistica ($n = 53$, $m = 16$) con una fidelity uguale a $F = 10^{-3}$. A questo punto viene formulata l'ipotesi nulla che la fidelity di Sycamore sia $F \leq 10^{-3}$ e l'ipotesi alternativa

qubits, n	cycles, m	total #paths	fidelity	run time
53	12	$4^{17}2^4$	1.4%	2 hours
53	14	$4^{21}2^4$	0.9%	2 weeks
53	16	$4^{25}2^3$	0.6%	4 years
53	18	$4^{28}2^3$	0.4%	175 years
53	20	$4^{31}2^4$	0.2%	10000 years

Fig. 3.5: Valori estrapolati dei tempi di esecuzione delle simulazioni di sampling. [3]

che $F > 10^{-3}$, così che se quest'ultima fosse vera allora sarebbe possibile affermare che un computer classico non può eseguire lo stesso campionamento effettuato da un computer quantistico. Usando il risultato (3.5.1) si rifiuta l'ipotesi nulla con una significatività di 6σ, dove $\sigma = 1/\sqrt{N_s}$ [3] e reclamando così il primato della supremazia quantistica.

Per completezza vengono riportati in Fig. 3.6 le curve dei costi di analisi del XEB in funzione della dimensione e della profondità dei circuiti.

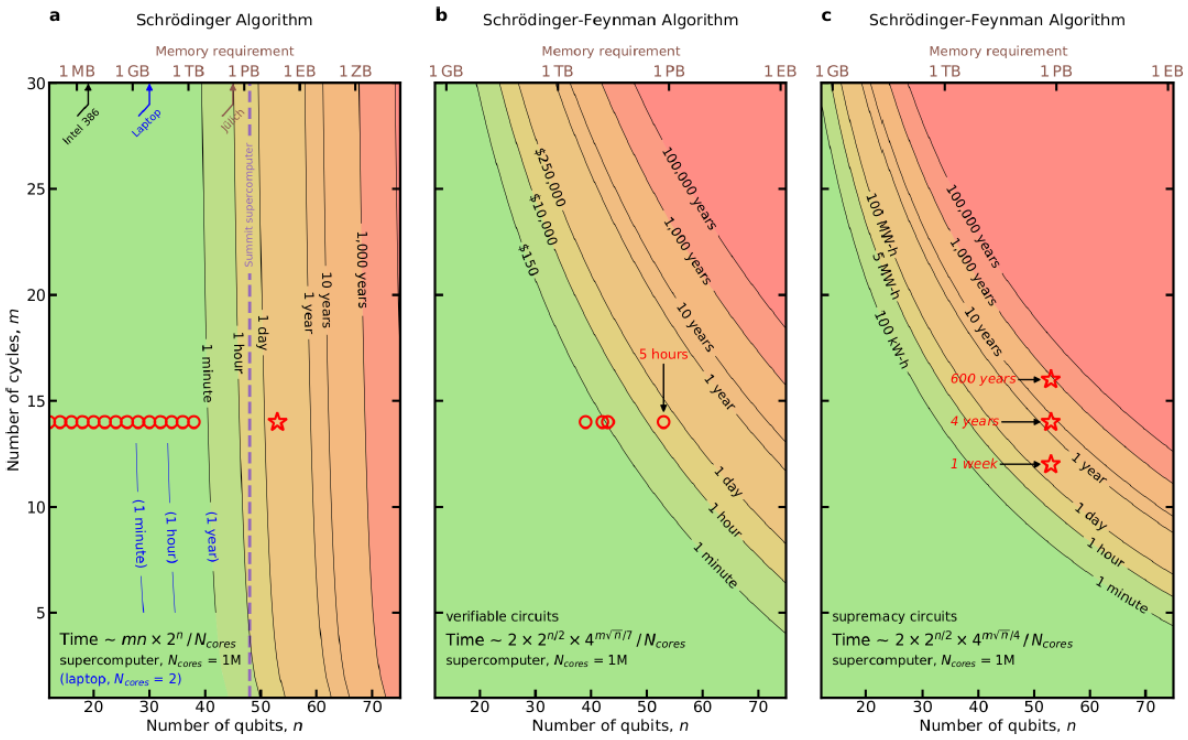


Fig. 3.6: Grafici delle relazioni tra m, n, tempi di simulazione e memoria necessaria al calcolo del XEB. I simboli in rosso (cerchi per le simulazioni effettuate, stelle per le estrapolazioni) corrispondono a quelli in Fig. 3.4. In (a) e (b) i circuiti sono verificabili, con la differenza che nel primo la simulazione usa il SA e nel secondo il SFA. In (c) sono mostrate le curve dei costi per i circuiti per la supremazia, mostrando come questi ultimi siano intrattabili anche dai più potenti super-computer del tempo (2019). [3]

Conclusioni

La dichiarazione di supremazia quantistica avanzata da Google nel 2019 è subito stata oggetto di dibattito e critiche: essendo Summit il super-computer più veloce a quel tempo proprietà di IBM, azienda rivale, quest'ultima fu la prima a sollevare perplessità a riguardo. Gli scienziati di IBM sostenevano che modificando il modo in cui Summit affronta il task del RCS fosse possibile ridurre i tempi da 10000 anni (ipotesi sostenuta da Google) a 2.5 giorni [4]. Altri scienziati hanno criticato il concetto stesso di supremazia quantistica e in generale lo sviluppo di grandi computer quantistici, avanzando l'ipotesi che il calcolo informatico quantistico sia reso impossibile dalla necessaria correzione degli errori per mantenerlo stabile; tra questi, un team dell'Università Ebraica di Gerusalemme ha pubblicato un articolo in cui criticava sia la metodologia di analisi statistica dei dati dell'esperimento del team di Google AI del 2019, sia la trasparenza con cui questo ha comunicato i dati sulla calibrazione del processore [25].

Nonostante le critiche ricevute da concorrenti e non, la ricerca e lo sviluppo di tecnologie quantistiche in grado di surclassare nel calcolo i più potenti super-computer non si sono arrestati: nell'Aprile 2023 Google ha di nuovo reclamato la supremazia quantistica del suo processore Sycamore-70 contro il primo *exascale* computer della storia e più veloce del mondo al momento, *Frontier* [5]. Per controbattere alle critiche sull'impossibile scalabilità dei processori quantistici a causa della natura degli errori a cui sono soggetti, sempre ad Aprile 2023 viene pubblicato uno studio su *Nature* dove un team di scienziati dell'Università di Yale dichiara di aver raddoppiato la durata operativa di un transmon qubit tramite la *Quantum Error Correction* (QEC), che quindi si dimostra realizzabile [26].

Di cruciale importanza per il progresso della tecnologia computazionale quantistica sarebbe riscire a dimostrare la supremazia quantistica per protocolli di effettiva utilità, come ad esempio la fattorizzazione in numeri primi di grandi numeri interi tramite l'algoritmo di Shor: al momento il numero intero più grande fattorizzato tramite questo algoritmo è stato $N = 21$ nel 2012 [27]. Un tentativo di fattorizzare il numero $N = 35$ è stato compiuto nel 2019, fallito per il grande quantitativo di errori [28]. Nel 2023 uno studio conclude che l'algoritmo di Shor non riesce a fattorizzare grandi numeri interi in presenza di rumore quantistico [29], di fatto rimettendo il problema nel campo della QEC.

In conclusione, il problema della supremazia quantistica è ancora aperto e le compagnie produttrici di computer quantistici, tra le quali figurano Google, IBM e Microsoft, stanno competendo tra loro per ottenerne il primato.

Appendice A

Il transmon qubit

Una delle tecnologie maggiormente utilizzate per lo sviluppo di processori quantistici è quella dei superconduttori e in questa sezione verranno accennate le basi del funzionamento e dell'utilizzo di questi ultimi nell'implementazione di circuiti quantistici, tra i quali il transmon.

A.1 La giunzione Josephson

Alla base della superconduttività ci sono le *coppie di Cooper*, ossia due elettroni accoppiati in uno stato legato [30]: a temperature estremamente basse nei metalli, le vibrazioni degli atomi del reticolo cristallino (*fononi*) causano una attrazione tra gli elettroni. Le coppie di Cooper quindi, pur essendo formate da fermioni (gli elettroni), si comportano come dei bosoni, poiché scambiando gli elettroni all'interno della coppia si cambia il segno della funzione d'onda due volte, lasciandola invariata [31]. Trovandosi a bassissima temperatura (i circuiti *quanto-elettrodinamici (QED)* operano a $T \sim 10mK$) le particelle si portano ai livelli energetici più bassi; essendo dei bosoni, le coppie di Cooper non vengono toccate dal principio di esclusione di Pauli e quindi "condensano" allo stato di energia più basso [32]. Per una trattazione completa sui circuiti superconduttori si rimanda a *Introduction to quantum electromagnetic circuits* [33].
Un componente fondamentale dei transmon qubit è la *giunzione Josephson* (*Josephson Junction, JJ*). Si tratta di un sistema formato da uno strato superconduttore, uno strato sottilissimo ($\sim 1nm$) di dielettrico e un altro strato di superconduttore (generalmente la JJ si compone con due strati di alluminio e uno di ossido di alluminio): per effetto tunnel le coppie di Cooper attraversano il dielettrico e creano una super-corrente, la quale può essere non nulla anche in assenza di differenza di potenziale tra i due strati di superconduttore [34, 35].
Nella descrizione hamiltoniana [36] della Josephson Junction vengono introdotte le variabili coniugate ϕ, n, rispettivamente la differenza di fase lungo la giunzione e il numero di coppie di Cooper nell' "isola" (si veda figura A.3 cosa si intende per isola) della giunzione (schematizzata in figura A.1):

$$\phi = \frac{2\pi(\Phi_1 - \Phi_2)}{\Phi_0} \tag{A.1.1}$$

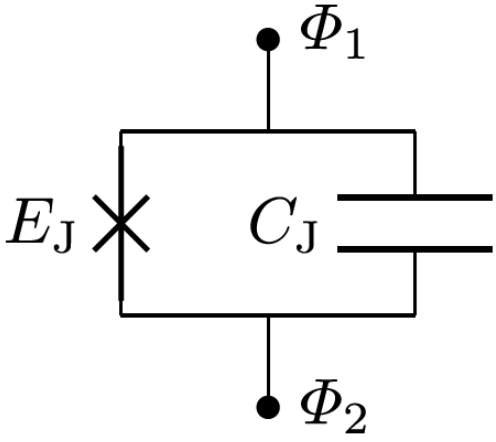


Fig. A.1: Schema circuitale di una giunzione Josephson con energia E_J e capacità parassita C_J. [36].

dove le Φ sono i *flussi ai nodi* (*node fluxes* [36]), $\Phi_0 = h/2e$ è il quanto di flusso magnetico, h è la costante di Planck, e è la carica elementare. La giunzione viene descritta dalle equazioni di Josephson:

$$I = I_c \sin\phi \tag{A.1.2}$$

$$\frac{\partial\phi}{\partial t} = \frac{2e}{\hbar}V(t) \tag{A.1.3}$$

dove I è la super-corrente che attraversa la giunzione, $V(t)$ la tensione lungo la giunzione, $\hbar$ è la costante di Planck ridotta, $I_c = 2eE_J/\hbar$ è la corrente critica oltre la quale la JJ si comporta come una giunzione dissipativa [35] (il valore è determinato dalla geometria del dispositivo e solitamente è di circa $40nA$ per i transmon, il che equivale ad una $L_{J0} \approx 8nH$ [37]) e $E_J = \hbar I_c/2e$ è l'energia Josephson [36].

Considerando la nota relazione $V = L\dot{I}$ si ricava un'espressione non lineare per l'induttanza della JJ [36]. Questo componente infatti si comporta come un induttore dipendente dalla corrente [37]:

$$L_J(I) = \frac{L_{J0}}{\sqrt{1 - \frac{I^2}{I_c^2}}} \tag{A.1.4}$$

dove $L_{J0} = \hbar/2e$ è il quanto ridotto di flusso magnetico.
L'energia associata a questa induttanza è:

$$H_L = -E_J \cos\phi \tag{A.1.5}$$

che diventerà il potenziale anarmonico nell'Hamiltoniana.
Data la struttura fisica della giunzione di Josephson [32], nell'espressione dell'energia totale è presente un termine legato ad una capacità parassita (in parallelo all'elemento induttore). La carica totale Q contenuta in questa è $2en$ (la carica viene misurata in unità di coppie di Cooper, ossia 2 elettroni) e considerando l'energia di carica di un singolo

elettrone $E_C = e^2/2C$, l'energia totale $Q^2/2C$ associata al condensatore della giunzione nell'Hamiltoniana è:

$$H_C = 4E_C n^2 \tag{A.1.6}$$

Questo porta ad un'espressione della Hamiltoniana del sistema:

$$H = H_C + H_L = 4E_C n^2 - E_J \cos\phi \tag{A.1.7}$$

Per comprendere l'importanza della non linearità nell'implementazione di un qubit, consideriamo un circuito LC con frequenza di oscillazione $\omega_r = 1/\sqrt{LC}$ nella condizione in cui $T << 1K$ e quindi $\hbar\omega_r >> k_B T$, ossia quando il rumore termico non disturba il sistema quantistico [36]. A questo punto il circuito LC può essere considerato un oscillatore armonico quantistico (*QHO*), con livelli di energia equispaziati:

$$E_n = \hbar\omega_r(n + \frac{1}{2}) \tag{A.1.8}$$

Quindi l'energia E_{01} per portare il sistema dal ground state $|0\rangle$ al primo stato eccitato $|1\rangle$ è la stessa necessaria per evolvere il sistema dallo stato $|1\rangle$ allo stato $|2\rangle$, o in generale dallo stato $|j\rangle$ allo stato $|j+1\rangle$. Questo fa sì che il circuito LC non sia adatto a costituire un qubit, poiché è necessario che il sistema resti nel sottospazio generato dai livelli $\{|0\rangle, |1\rangle\}$ [36]. Tradotto in termini di potenziale, il circuito LC (quantizzato) ha un potenziale quadratico (quello del QHO), mentre quello di un circuito con giunzione Josephson, ad esempio un circuito JJC (un prototipo del transmon, il *Cooper Pair Box, CPB*), presenta un potenziale cosinusoidale anarmonico visto in (A.1.5) (rappresentazione grafica dei livelli energetici in figura A.2).

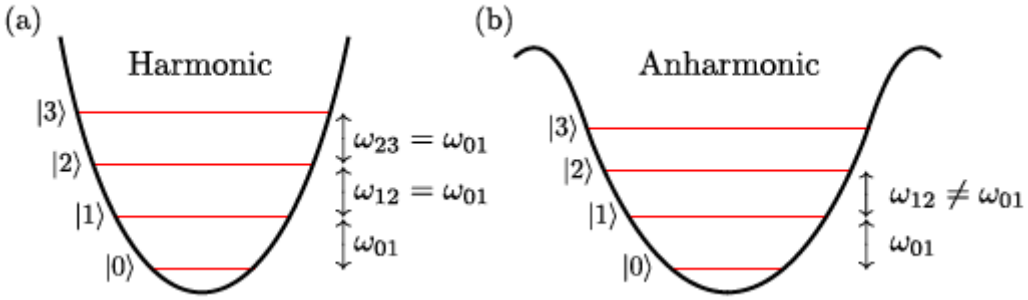


Fig. A.2: Le ω_{ij} sono le frequenze di transizione tra i livelli j, k. (a) Il potenziale del QHO (o del circuito LC quantizzato). I livelli energetici sono equidistanti. (b) Il potenziale anarmonico di un circuito dominato dalla componente JJ. L'energia di transizione tra livelli adiacenti non è costante. [36].

Quantitativamente, i qubit basati sulle JJ hanno frequenze di transizione nell'intervallo 4-8 GHz (i transmon di Sycamore operano nel range 5-7 GHz [3]) e anarmonicità in 150-

300 MHz rispettivamente [37]. Questa piccola differenza tra le energie di transizione tra livelli è sufficiente a contenere il sistema nel sottospazio computazionale dei primi due livelli.

A.2 Dal CPB al transmon

Il transmon è un *qubit di carica* (*charge qubit*): ciò significa che il suo stato ($|0\rangle$ o $|1\rangle$) è determinato da un valore della carica (rispetto ad altri tipi di qubit superconduttori il cui stato è determinato dal flusso o dalla fase, rispettivamente il *flux qubit* e il *phase qubit* [36]). Il primo qubit di carica realizzato è il già citato Cooper Pair Box, di cui viene fornita di seguito uno schema circuitale (figura A.3). Si noti che la giunzione Josephson viene rappresentata come un condensatore (la capacità parassita) in parallelo con un elemento "X" contenente la parte caratterizzata dalle equazioni di Josephson (A.1.2),(A.1.3).

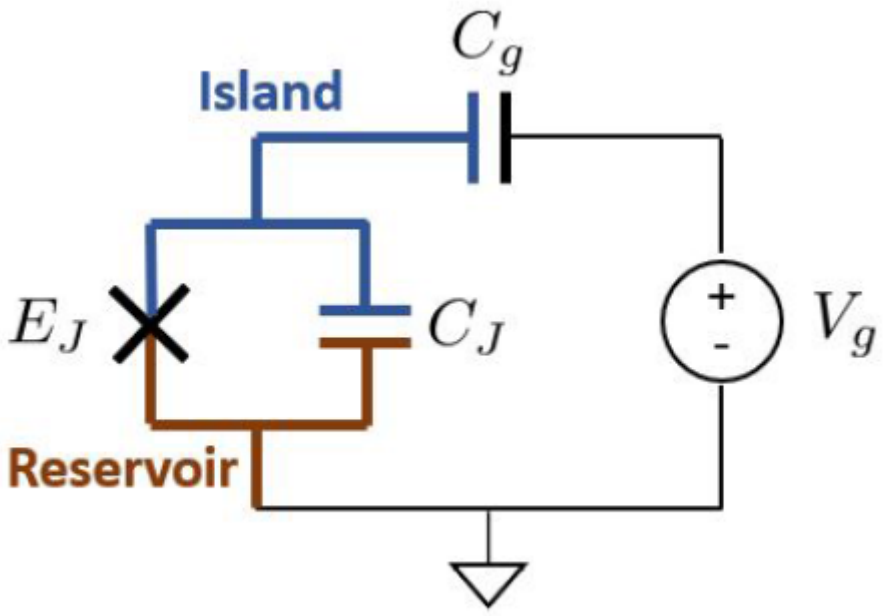


Fig. A.3: Schema di un Cooper Pair Box. La JJ (E_J, C_J in parallelo) è connessa ad un generatore di tensione V_g tramite il condensatore C_g (capacità di gate). [32].

Il CPB è formato da una "isola" connessa ad un "serbatoio" (*island, reservoir* in figura A.3) tramite una JJ. L'isola non è direttamente connessa con circuiti esterni, di conseguenza è molto sensibile al numero di coppie di Cooper che attraversano la giunzione ed è per questo che questo tipo di qubit sono chiamati qubit di carica [32]. L'Hamiltoniana del sistema CPB è molto simile alla (A.1.7):

$$H_{CPB} = 4E_C(n - n_g)^2 - E_J \cos\phi \tag{A.2.1}$$

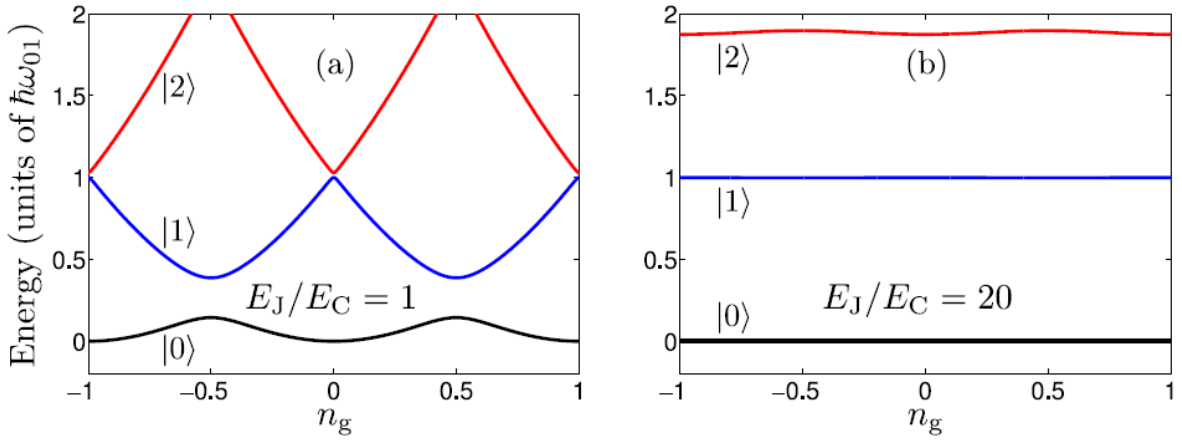


Fig. A.4: Livelli di energia dei primi 3 stati del CPB (a) e del transmon (b) normalizzati dall'energia di transizione tra i primi due stati. [36].

dove n_g è la carica indotta dal generatore di tensione e tiene conto dell'effetto della capacità di gate e $C = C_J + C_g$ è la capacità associata a E_C.

Per una descrizione quantistica del sistema, le variabili coniugate vengono "promosse" ad operatori [32, 36] (si omette la notazione con i cappelli per semplicità) e rispondono alla relazione di commutazione:

$$[e^{i\phi}, n] = 1 \tag{A.2.2}$$

In particolare, il principio di indeterminazione nella forma [36]:

$$\Delta\phi\Delta n \geq 1 \tag{A.2.3}$$

indica che non è possibile misurare con arbitraria precisione in contemporanea le osservabili associate alle due variabili (adesso operatori). Entra in gioco allora il parametro E_J/E_C: quando questo valore è $\ll 1$ il numero di cariche n è ben definito, a discapito della fase ϕ [36]. Tipicamente un CPB è caratterizzato da un rapporto $E_J/E_C < 1$, è quindi opportuno riscrivere l'Hamiltoniana in (A.2.1) nella base ortogonale [32] degli autostati $|N\rangle$ dell'operatore di carica n [33]:

$$H_{CPB} = \sum_{N=-\infty}^{+\infty} \left[4E_C(N - n_g)^2 \, |N\rangle \langle N| - \frac{1}{2} E_J(|N+1\rangle \langle N| + |N\rangle \langle N+1|) \right] \tag{A.2.4}$$

si noti che il parametro n_g rimane una variabile classica. L'operatore $|N+1\rangle \langle N|$ rappresenta il trasferimento *coerente* di una coppia di Cooper dal serbatoio all'isola; l'operatore $|N\rangle \langle N+1|$ indica il trasferimento opposto, ossia dall'isola al serbatoio [32]. Dal grafico (a) in figura A.4 (dove viene usata l'approssimazione $E_J/E_C = 1$ per semplicità) si può notare la dipendenza dei livelli energetici dal parametro n_g. Per valori interi di n_g si ha una grande separazione tra il ground state e i livelli eccitati, condizione favorevole alla

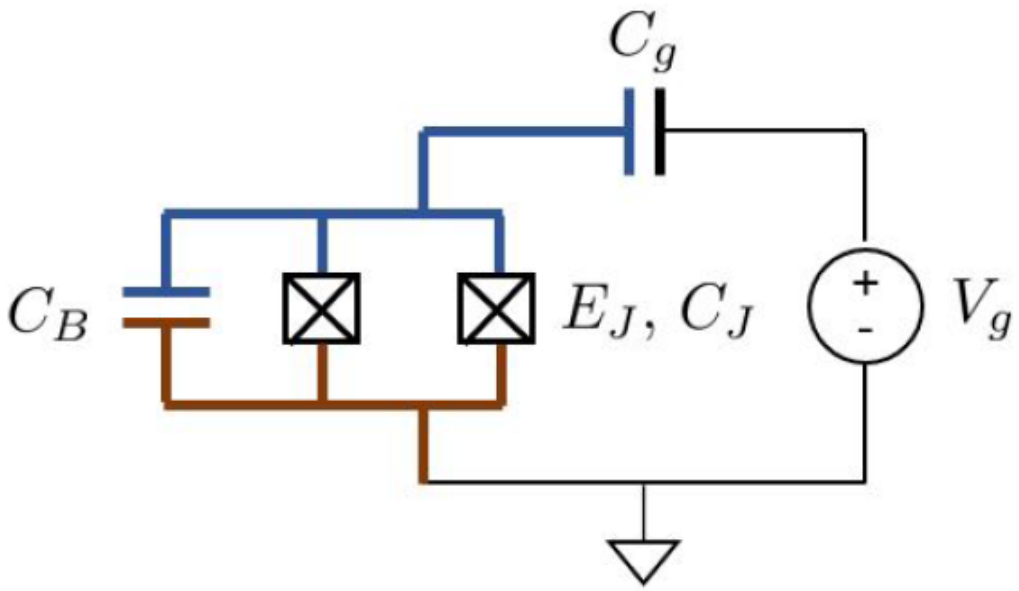


Fig. A.5: Schema circuitale del transmon qubit. [32].

inizializzazione del CPB nello stato $|0\rangle$ [32]. Tuttavia per questi valori, i primi due livelli eccitati sono molto vicini, rendendo pericolosamente probabile la transizione spontanea dal livello $|1\rangle$ al livello $|2\rangle$, quindi per le operazioni sui qubit si utilizzano valori semi-interi $n_g = m + \frac{1}{2}$, $m \in \mathbb{Z}$. Questi valori, chiamati anche *sweet spots* [32, 36], garantiscono grande separazione tra i livelli energetici dei primi 2 stati eccitati e una marcata anarmonicità, rendendo possibile la transizione tra il ground state ed il primo livello eccitato usando veloci ($\sim 10ns$) impulsi di microonde [32]. Inoltre, i valori semi-interi di n_g garantiscono una riduzione delle fluttuazioni di carica, o *rumore di carica* (*charge noise*), rispetto al CPB con valori di n_g interi [32][36]. Ciononostante, il CPB rimane un dispositivo molto sensibile al rumore di carica, motivo per cui è stato sviluppato il transmon. Prima di arrivare al transmon bisogna passare dallo *SQUID*, ossia un *superconducting quantum interference device*: in breve si tratta di due JJ in parallelo, rappresentate in maniera condensata da due "X" racchiuse da un quadrato (si veda figura A.5). Il grande vantaggio di utilizzare lo SQUID al posto di una singola JJ risiede nella possibilità di regolare l'energia Josephson E_J (non regolabile nella giunzione Josephson singola) e quindi controllare l'importante parametro E_J/E_C. Per approfondimenti sullo SQUID si rimanda a [32]. Combinando le varie tecnologie discusse in questo capitolo si giunge al *transmission-line shunted plasma oscillation qubit* [36], ossia il transmon. Osservando lo schema del circuito in figura A.5 si osserva la struttura: uno SQUID (E_J, C_J) connesso in parallelo ad una capacità C_B, il tutto connesso ad un generatore di tensione V_g (in realtà omesso nel design pratico [32]) tramite la capacità di gate C_g. Aggiungendo il condensatore C_B si diminuisce l'energia $E_C = e^2/2C$, aumentando di conseguenza il rapporto E_J/E_C che arriva a valori $\approx 10^2$.

Il qubit così passa da avere una n definita ad avere una fase ϕ ben definita [36]. I livelli energetici osservabili in figura A.4(b) non sono più influenzati dal parametro n_g: il rumore diminuisce esponenzialmente a discapito di una perdita (di ordine minore rispetto all'ordine esponenziale [32]) di anarmonicità del circuito, che tuttavia rimane tale da poter effettuare operazioni sul qubit con rapidi impulsi di microonde. La forma dell'hamiltoniana del transmon è la stessa di quella relativa al CPB (A.2.1) [32] e, considerando $E_J >> E_C$, si ottengono (in approssimazione, seguendo la teoria perturbativa) i livelli di energia del circuito, da cui si estrapolano rispettivamente la frequenza di transizione ω_{01} tra i primi 2 livelli del qubit e l'anarmonicità $\omega_{12} - \omega_{01}$ [36]:

$$\omega_{01} = \frac{1}{\hbar}\left(\sqrt{8E_J E_C} - E_C\right) \tag{A.2.5}$$

$$\omega_{12} - \omega_{01} = -\frac{E_C}{\hbar} \tag{A.2.6}$$

Bibliografia

[1] Richard P. Feynman. *Simulating phsyics with computers*. In: *International Journal of Theoretical Physics* 21.6 (1982), pp. 467–488. DOI: 10.1007/BF02650179.

[2] Roman P.Poplavskii. *Thermodynamical models of information processing*. In: *Uspekhi Fizicheskikh Nauk* 115.3 (1975), pp. 465–501. DOI: 10.1070/PU1975v018n03ABEH001955.

[3] Frank Arute et al. *Quantum supremacy using a programmable superconducting processor*. In: *Nature* 574.7779 (2019), pp. 505–510. DOI: 10.1038/s41586-019-1666-5.

[4] Edwin Pednault et al. *Leveraging Secondary Storage to Simulate Deep 54-qubit Sycamore Circuits*. 2019. arXiv: 1910.09534 [quant-ph].

[5] Alexis Morvan et al. *Phase transition in Random Circuit Sampling*. 2023. arXiv: 2304.11119 [quant-ph].

[6] Michael A. Nielsen e Isaac L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2011. ISBN: 9781107002173.

[7] Hans Hon Sang Chan et al. *Grid-based methods for chemistry simulations on a quantum computer*. In: *Science Advances* 9.9 (2023), eabo7484. DOI: 10.1126/sciadv.abo7484.

[8] Christopher D. B. Bentley et al. *Quantum computing for transport optimization*. 2022. arXiv: 2206.07313 [quant-ph].

[9] Albert Messiah. *Quantum Mechanics*. Dover Publications, 1999. ISBN: 9780486409245.

[10] John Preskill. *Lecture Notes for Ph219/CS219: Quantum Information, Chapter 3*. California Institute of Technology. 2018. URL: http://theory.caltech.edu/~preskill/ph219/chap3_15.pdf.

[11] David P. DiVincenzo. *Two-bit gates are universal for quantum computation*. In: *Physical Review A* 51.2 (1995), pp. 1015–1022. DOI: 10.1103/physreva.51.1015.

[12] Philip Krantz et al. *A quantum engineer's guide to superconducting qubits*. In: *Applied Physics Reviews* 6.2 (2019). DOI: 10.1063/1.5089550.

[13] Sean Mullane. *Sampling random quantum circuits: a pedestrian's guide*. 2020. arXiv: 2007.07872 [quant-ph].

[14] Sergio Boixo et al. *Characterizing quantum supremacy in near-term devices*. In: *Nature Physics* 14.6 (2018), pp. 595–600. DOI: 10.1038/s41567-018-0124-x.

[15] Freeman J. Dyson. *Statistical Theory of the Energy Levels of Complex Systems I*. In: *Journal of Mathematical Physics* 3.1 (2004), pp. 140–156. DOI: 10.1063/1.1703773.

[16] Sergio Boixo et al. *Supplementary information for "Characterizing quantum supremacy in near-term devices"*. In: *Nature Physics* 14.6 (2018), pp. 595–600. DOI: 10.1038/s41567-018-0124-x.

[17] Thomas M. Cover e Joy A. Thomas. *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience, 2006. ISBN: 0471241954.

[18] Thomas S. Robinson. *10 Fundamental Theorems for Econometrics*. 2022. URL: https://bookdown.org/ts_robinson1994/10EconometricTheorems/.

[19] Sito ufficiale del super-computer Leonardo. URL: https://leonardo-supercomputer.cineca.eu/it/leonardo-hpc-system/.

[20] Shaowei Li et al. *Realisation of high-fidelity nonadiabatic CZ gates with superconducting qubits*. In: *npj Quantum Inf* 5.84 (2019). DOI: 10.1038/s41534-019-0202-7.

[21] Gabriele Sicuro. *Introduzione all'informatica quantistica*. Appunti dal corso tenuto dal prof. V.Giovannetti (Scuola Normale Superiore di Pisa). 2012. URL: https://gabrielesicuro.files.wordpress.com/2012/08/information1.pdf.

[22] Igor L. Markov e Yaoyun Shi. *Simulating Quantum Computation by Contracting Tensor Networks*. In: *SIAM Journal on Computing* 38.3 (2008), pp. 963–981. DOI: 10.1137/050644756.

[23] Sito ufficiale di Intel. URL: https://www.intel.com/content/www/us/en/gaming/resources/hyper-threading.html.

[24] Blog ufficiale di Google Cloud. URL: https://cloud.google.com/blog/products/gcp/introducing-ultramem-google-compute-engine-machine-types.

[25] Gil Kalai et al. *Questions and Concerns About Google's Quantum Supremacy Claim*. 2023. arXiv: 2305.01064 [quant-ph].

[26] Sivak et al. *Real-time quantum error correction beyond break-even*. In: *Nature* 616 (2023), pp. 50–55. DOI: https://doi.org/10.1038/s41586-023-05782-6.

[27] Enrique Martín-López et al. *Experimental realization of Shor's quantum factoring algorithm using qubit recycling*. In: *Nature Photon* 6 (2012), pp. 773–776. DOI: https://doi.org/10.1038/nphoton.2012.259.

[28] Mirko Amico et al. *Experimental study of Shor's factoring algorithm using the IBM Q Experience*. In: *Phys. Rev. A* 100 (1 2019), p. 012305. DOI: 10.1103/PhysRevA.100.012305.

[29] Jin-Yi Cai. *Shor's Algorithm Does Not Factor Large Integers in the Presence of Noise*. 2023. arXiv: 2306.10072 [quant-ph].

[30] Leon N. Cooper. *Bound Electron Pairs in a Degenerate Fermi Gas*. In: *Phys. Rev.* 104 (1956), pp. 1189–1190. DOI: 10.1103/PhysRev.104.1189.

[31] Richard P. Feynman et al. *The Feynman Lectures on Physics, Vol. III: The New Millennium Edition: Quantum Mechanics*. Basic Books, 2011. ISBN: 9780465025015.

[32] Thomas E. Roth et al. *An Introduction to the Transmon Qubit for Electromagnetic Engineers*. 2021. arXiv: 2106.11352 [quant-ph].

[33] Uri Vool e Michel Devoret. *Introduction to quantum electromagnetic circuits*. In: *International Journal of Circuit Theory and Applications* 45.7 (2017), pp. 897–934. DOI: 10.1002/cta.2359.

[34] John M. Martinis et al. *Energy-Level Quantization in the Zero-Voltage State of a Current-Biased Josephson Junction*. In: *Physical Review letters* 55 (1985).

[35] Mahdi Naghiloo. *Introduction to Experimental Quantum Measurement with Superconducting Qubits*. 2019. arXiv: 1904.09291 [quant-ph].

[36] Anton Frisk Kockum e Franco Nori. *Fundamentals and Frontiers of the Josephson Effect*. Springer International Publishing, 2019, pp. 703–741. DOI: 10.1007/978-3-030-20726-7.

[37] Joseph C. Bardin et al. *Microwaves in Quantum Computing*. In: *IEEE Journal of Microwaves* 1.1 (2021), pp. 403–427. DOI: 10.1109/jmw.2020.3034071.

Voglio ringraziare la mia famiglia per avermi sostenuto durante questi anni.