96-quantum-shor-period-finding-03

Quantum Computing

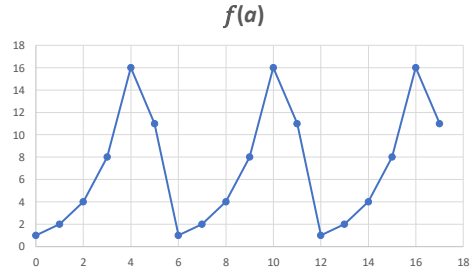
Ricerca del periodo di una funzione utilizzando la QFT e sintesi dell'algoritmo di Shor

1

1

ricerca del periodo di una funzione

- è data una funzione $f: \{0,1,\cdots,N-1\} \rightarrow S$ tale che per $r > 0$ e per ogni a abbiamo $f(a+r) = f(a)$; *trovare* il periodo r
- assumiamo che i valori di f nel periodo siano tutti diversi e, per semplicità, che N/r sia intero



in questo caso $r = 6$, $N = 18$ e il numero di ripetizioni del periodo è $N/r = 3$

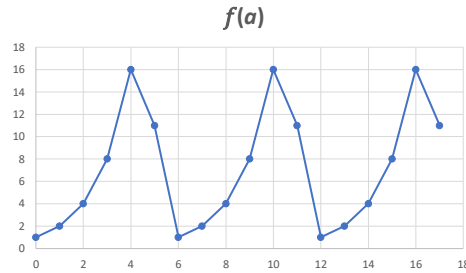
2

2

96-quantum-shor-period-finding-03

la funzione che ci interessa

- per l'algoritmo di Shor la funzione della quale ci interessa calcolare il periodo è $f(a) = x^a \pmod N$ dove x è un numero random



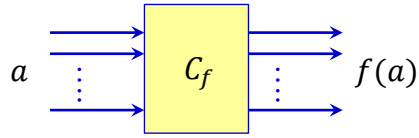
in questo caso $r = 6$, $N = 18$ e il numero di ripetizioni del periodo è $N/r = 3$

3

3

disponibilità della funzione

- possiamo implementare $f(a)$ con un C_f classico
- se cerchiamo una periodicità di f in modo tradizionale dobbiamo fornire diversi valori, magari random, al circuito sperando di trovare una collisione nei risultati (due numeri a_1 e a_2 tali che $f(a_1) = f(a_2)$)
  - troppi tentativi!



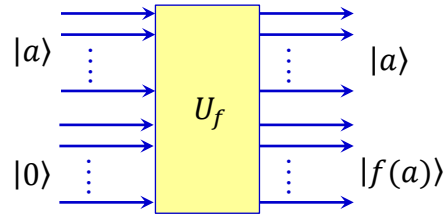
4

4

96-quantum-shor-period-finding-03

una versione quantum della funzione

- possiamo trasformare C_f in un circuito quantum U_f con i corrispondenti input e output

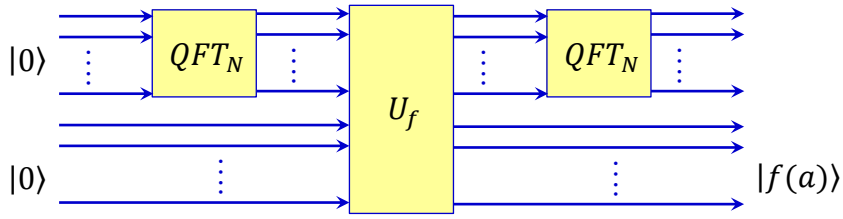


5

5

uso di U_f in un circuito più complesso

- prima e dopo facciamo la trasformata di Fourier



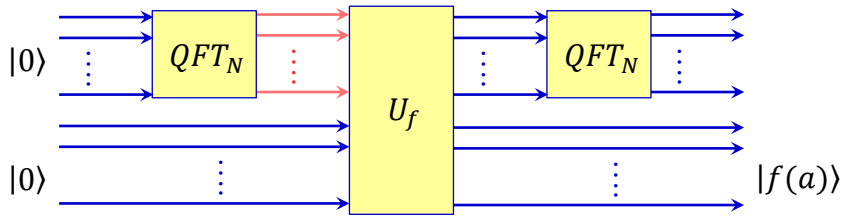
6

6

96-quantum-shor-period-finding-03

uso di U_f in un circuito più complesso

- prima e dopo facciamo la trasformata di Fourier



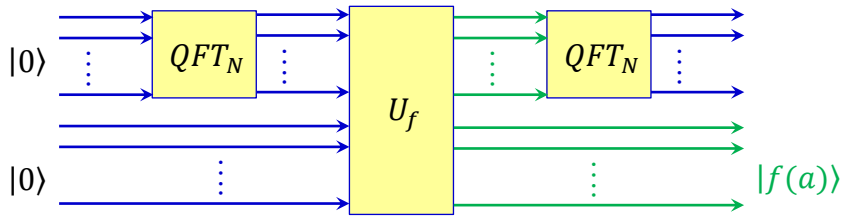
- dopo QFT_N lo stato della prima sequenza di qubit è $\frac{1}{\sqrt{N}}\sum_{a=0}^{N-1}|a\rangle$; si noti come a nella somma sia l'intero corrispondente alla codifica di uno stato base

7

7

uso di U_f in un circuito più complesso

- prima e dopo facciamo una trasformata di Fourier



- dopo U_f lo stato di tutta la sequenza di qubit è

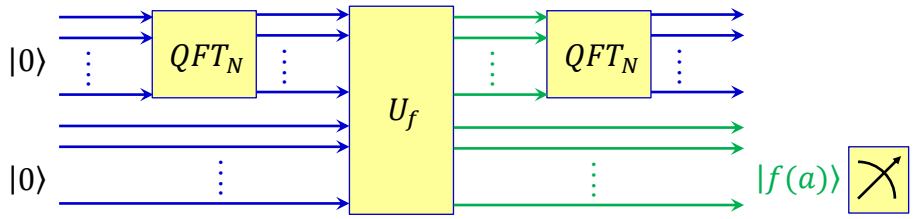
$$\frac{1}{\sqrt{N}}\sum_{a=0}^{N-1}|a\rangle|f(a)\rangle = \frac{1}{\sqrt{N}}\sum_{a=0}^{N-1}|a\rangle|x^a \pmod N\rangle$$

8

8

96-quantum-shor-period-finding-03

se effettuiamo una misura



- dopo U_f lo stato di tutta la sequenza di qubit è

$$\frac{1}{\sqrt{N}}\sum_{a=0}^{M-1}|a\rangle|x^a \pmod N\rangle$$

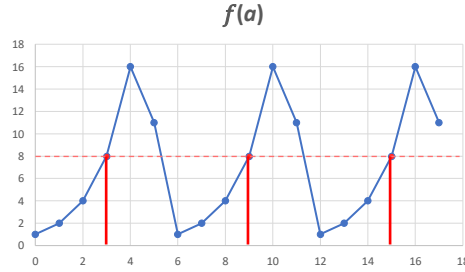
- e se facciamo una misura su $|f(a)\rangle$ otteniamo un $f(\hat{a})$ corrispondente a un qualche valore $\hat{a}$

9

9

se effettuiamo una misura – esempio

- supponiamo di aver misurato $f(\hat{a}) = 8$
- ci sono diversi valori di a con $f(a) = 8$; nell'esempio 3, 9, e 15 distanti 6 uno dall'altro, dove 6 è il periodo

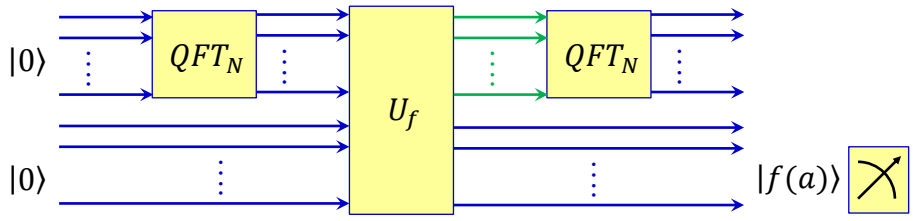


10

10

96-quantum-shor-period-finding-03

dopo la misura – gli output $|a\rangle$



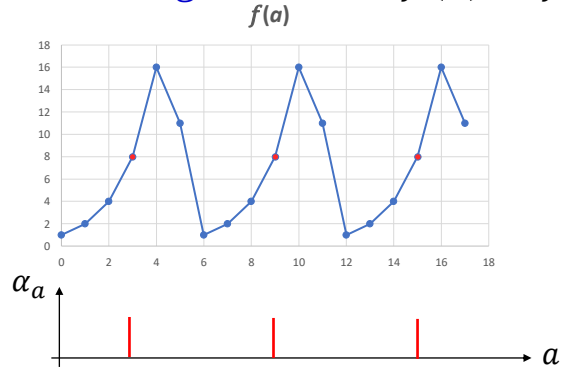
- dopo la misura cosa ne è degli output $|a\rangle$?

11

11

dopo la misura – gli output $|a\rangle$

- dopo la misura se guardiamo lo stato nel quale si trovano gli output $|a\rangle$ troviamo $\sum_{a=0}^{N-1} \alpha_a |a\rangle$ dove gli α_a sono diversi da zero solo in corrispondenza degli a tali che $f(a) = f(\hat{a})$



12

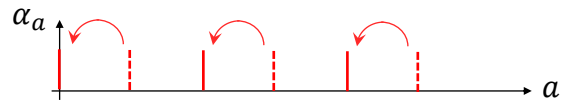
12

96-quantum-shor-period-finding-03

ricordiamo le proprietà della QFT

- abbiamo visto che se facciamo la trasformata di Fourier di questa funzione

- oppure della stessa *ruotata* (*shiftata*)



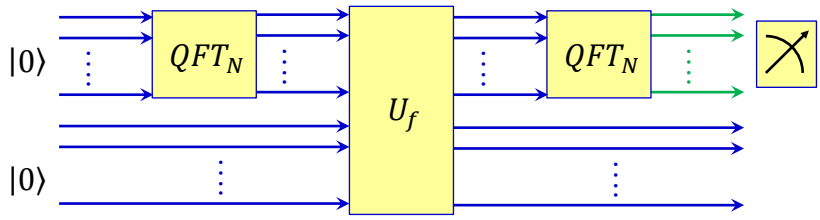
- otteniamo lo stesso risultato

13

13

effettuiamo una nuova misura

- misuriamo ora la prima sequenza di qubit



14

14

96-quantum-shor-period-finding-03

applichiamo le proprietà della QFT

- nella nuova misura otteniamo un multiplo random di $\frac{N}{r}$; nell'esempio con $r = 6$, $N = 18$ potremmo ottenere 3 oppure 6 oppure 9
- come ottenere r?
  - basta fare due volte l'esperimento (ovviamente con la stessa funzione), ottenere due multipli random di $\frac{N}{r}$ e calcolare il loro gcd, cosa che si fa in modo efficiente
  - ad esempio dai due esperimenti potremmo avere 6 e 9 e quindi calcolare $\frac{N}{r} = 3$ e, da $N = 18$ ottenere $r = 6$

15

15

algoritmo di Shor

- input: N con almeno due fattori primi distinti
- output: un divisore di N
  - scegli in modo uniforme un numero x random tra 0 e $N - 1$
  - se $\gcd(x, N) \neq 1$ allora il divisore comune di x ed N è anche un divisore di N
  - altrimenti, calcola il periodo r della funzione $f(a) = x^a \pmod N$; se r è pari ed è tale che $(x^{r/2})^2 \equiv 1 \pmod N$ allora il divisore è $\gcd(N, x + 1)$; altrimenti riprova con un altro x

16

16

96-quantum-shor-period-finding-03

algoritmo di Shor

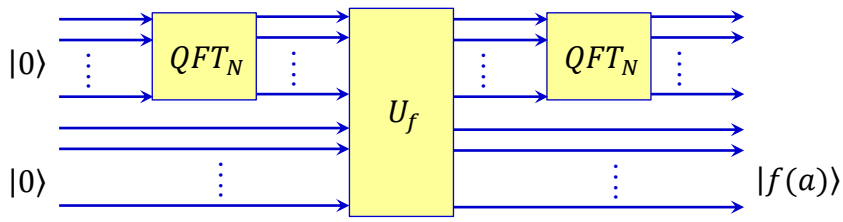
– per calcolare il periodo r della funzione $f(a) = x^a \pmod N$ usa il quantum circuit e l'algoritmo appena descritti

17

17

algoritmo di Shor – costo

- il passo critico è il calcolo del periodo



- Shor ha dimostrato che QFT_N può essere implementata usando $O(\log N^2)$ porte con uno o due ingressi

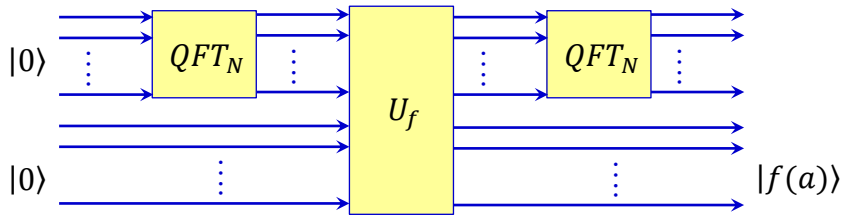
18

18

96-quantum-shor-period-finding-03

algoritmo di Shor – costo

- implementazione di U_f



- anche U_f è implementabile efficientemente
  - l'esponente può essere calcolato con il *metodo dei quadrati ripetuti*, che richiede un tempo (numero di porte) polinomiale in $\log N$

19

19