1 Introduzione

1.1 Cos'è il Gray Code?

Il Gray Code, noto anche come codice riflessivo binario, è un sistema di codifica in cui due valori successivi differiscono per un solo bit. Questo lo rende particolarmente utile nei circuiti digitali e nella progettazione di FFQ-RAM (Fully Functional Quantum Random Access Memory), poiché riduce il numero di operazioni necessarie per la transizione tra stati successivi.

Ad esempio:

- Codifica binaria tradizionale per 2 bit: $00, 01, 10, 11$;
- Gray Code per 2 bit: $00, 01, 11, 10$.

Utilizzando il Gray Code, possiamo ridurre l'utilizzo di gate X nelle FF-QRAM.

2 Introduzione al problema

Nota: n indica sempre il numero di bit di registro.

Normalmente, per rappresentare n elementi usiamo $n \cdot 2^n$ gate X, come mostrato in Figura 1. Cercheremo di dimostrare che, utilizzando il Gray Code, per rappresentare n qubit di registro possiamo utilizzare solo:

$$n + 2^n + n - 2 = 2n + 2^n - 2.$$

2.1 Esempi di FF-QRAM

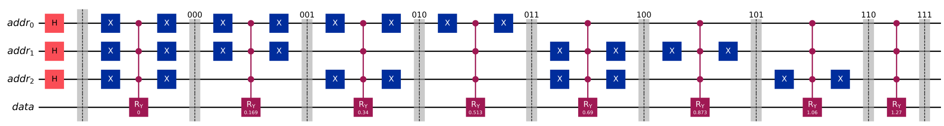

Figura 1: Esempio di FF-QRAM classica con 3 qubit di registro e 1 qubit di data.

> *Come si vede dall'immagine, ogni volta che dobbiamo mettere un bit a 0, serve un gate X e poi un altro per riportarlo allo stato precedente. Questo rappresenta uno spreco: se dobbiamo rappresentare due 0 consecutivi, possiamo utilizzare 2 gate X anziché 4. Lo stesso vale per sequenze più lunghe di 0 consecutivi.*

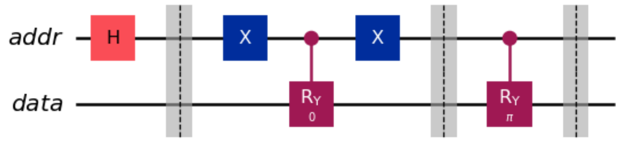

Figura 2: FF-QRAM con 1 qubit di registro e 1 qubit, ottimizzato con Gray Code.

Come mostrato in Figura 2, utilizzando il Gray Code abbiamo un unico X-gate per rappresentare il bit a 0 e un altro per portare il bit a 1. Non serve fare uncomputing poiché il bit è già nello stato desiderato.

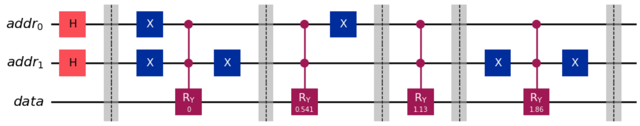

Figura 3: FF-QRAM con 2 qubit di registro e 1 qubit, ottimizzato con Gray Code.

In Figura 3, abbiamo 2 gate X per rappresentare lo stato iniziale 00*. Successivamente, servono 3 X-gate per le transizioni tra gli stati successivi. Infine, per l'uncomputing dello stato finale, sono necessari* $n - 1 = 1$ *gate X.*

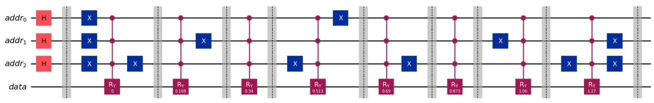

Figura 4: FF-QRAM con 3 qubit di registro e 1 qubit, ottimizzato con Gray Code.

In Figura 4, abbiamo:

- $n = 3$ *gate X per rappresentare lo stato iniziale* 000*;*
- $2^n - 1 = 7$ *gate X per rappresentare gli stati successivi;*
- $n - 1 = 2$ *gate X per l'uncomputing dello stato finale.*

In totale, servono $3 + 7 + 2 = 12$ *gate X, rispetto ai 24 necessari nella Figura 1.*

3 Ipotesi e Tesi

Ipotesi ***Ipotesi:*** *Dati n bit di registro, dimostriamo che il numero di gate X necessari è dato da*

$$P(n) = 2n + 2^n - 2.$$

Tesi: *Utilizzando il Gray Code, ogni transizione da n a $n+1$ bit di registro aggiunge $2^n + 2$ gate X. Ad esempio:*

- *Con 1 bit: 2 gate X;*
- *Con 2 bit: 6 gate X ($4 = 2^1 + 2$ in più);*
- *Con 3 bit: 12 gate X ($6 = 2^2 + 2$ in più).*

4 Dimostrazione per Induzione

Teorema 4.1. Dimostriamo che $P(n+1) = P(n) + 2^n + 2$.

Dimostrazione. **Ipotesi:** Supponiamo che la formula sia valida per $n = k$.
Tesi: Dimostrare che è valida anche per $n = k + 1$.

Abbiamo:

- $k + 1$ X-gate per rappresentare $k + 1$ bit a 0 nello stato iniziale;
- $2^{k+1} - 1$ gate X per rappresentare le transizioni tra stati successivi;
- $k + 1 - 1 = k$ gate X per l'uncomputing dello stato finale.

Sommando:

$$k + 1 + 2^{k+1} - 1 + k = 2k + 2^{k+1} - 2 + 2 = P(k) + 2^k + 2.$$

■