UNIVERSITÀ DEGLI STUDI
DI GENOVA

Università degli Studi di Genova

Fondamenti di Computazione Quantistica

Lorenzo Vaccarecci

Anno Accademico 2024/2025

Indice

Capitolo 1

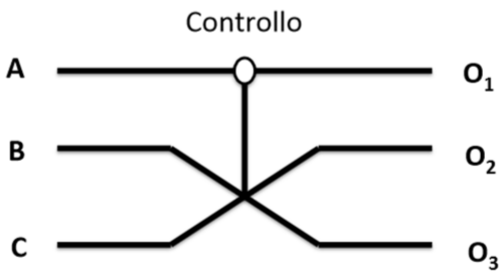
Introduzione

1.1 Porte logiche universali

- `NOT(A)` $\equiv \bar{A}$
- `AND(A,B)` $\equiv A \cdot B$ oppure $A \wedge B$
- `OR(A,B)` $\equiv A + B$ oppure $A \vee B$
- `XOR(A,B)` $\equiv A \oplus B = (A + B) \mod 2$
- `NAND(A,B)` $\equiv A \bar{\cdot} B$ oppure $A \bar{\vee} B$
- `NOR(A,B)` $\equiv A \bar{+} B$ oppure $A \bar{\wedge} B$

L'insieme di `AND` e `NOT` oppure di `OR` e `NOT` sono insiemi universali. Questo significa che, ad esempio, usando solo combinazioni di porte `AND` e `NOT` è possibile implementare una qualsiasi funzione booleana. Pur formando set universali, le porte `AND`, `OR`, `NAND` e `NOR` sono però **irreversibili**. A livello concettuale è interessante introdurre delle porte logiche che siano **reversibili**. Questo vuol dire che se combiniamo in sequenza una porta logica reversibile con la sua inversa, riotteniamo l'informazione originale. La porta di Fredkin può essere interpretata come uno *switch* controllato di bit. Il bit di controllo è `A`; se questo è acceso i bit `B` e `C` vengono scambiati, altrimenti vengono lasciati identici.

A	B	C	Out1	Out2	Out3
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	1	0	0
1	0	1	1	1	0
1	1	0	1	0	1
1	1	1	1	1	1



1.2 Operazioni Bit-a-Bit

A una stringa di n bit possiamo associare un intero compreso fra 0 e $N-1$ con $N = 2^n$. All'intero x associamo la stringa di bit $x_0x_1x_2 \dots x_n$ con $x_i = 0, 1$ e $i = 0, 1, \dots, n$ tale che $x = \sum_{i=0}^{n} x_i 2^{n-i}$. Possiamo codificare $N = 2^n$ interi ma questi saranno compresi fra 0 e $N - 1 = 2^n - 1$.

1.2.1 Prodotto interno bit-per-bit

$$x \cdot z \equiv (x_1 z_1 + x_2 z_2 + \cdots + x_n z_n) \mod 2$$

E' anche chiamato prodotto `AND` bitwise perchè si ottiene prendendo le operazioni `AND` fra i singoli bit.

1.2.2 Somma bit-per-bit: bitwise `XOR`

Indichiamo con $x \oplus z$ la somma bit-per bit, modulo 2. Il risultato questa volta è una stringa il cui i-esimo bit ha il valore $x_i + z_i \mod 2 = x_i \texttt{ XOR } z_i$.

1.3 Prerequisiti matematici

1.3.1 Numeri complessi

Ogni numero complesso $z \in \mathbb{C}$ può essere scritto come $z = a + ib$, con $a \in \mathbb{R}$ **parte reale** e $b \in \mathbb{R}$ **parte immaginaria**. Se $z = a + ib$ e $w = c + id$, abbiamo

$$z + w = (a + c) + i(b + d)$$

$$z \cdot w = (ac - bd) + i(ad + bc)$$

Per ogni $z \in \mathbb{C}$, $z \cdot z* = a^2 + b^2$ è reale e non negativo. Inoltre, $\sqrt{a^2 + b^2} = |z|$ è detto **modulo** di z.

$$|z|^2 = z \cdot z*$$

Possiamo rappresentare un numero complesso $z = a + ib$ come una coppia (a,b) sul piano complesso. L'asse delle ascisse è utilizzato per la parte reale e l'asse delle ordinate per la parte immaginaria. Si ha $a = |z| \cos\theta$ e $b = |z| \sin\theta$ dove θ è la **fase**. Se $z = 0$ allora θ non è definita. Per $|z| = 1, z = \cos\theta + i \sin\theta$. Più in generale possiamo scrivere $z = pe^{i\theta}$ con $p = |z|$ e $e^{i\theta} = \cos\theta + i \sin\theta$.

1.3.2 Spazi vettoriali in 2D

- **Direzione**: rappresentata dalla retta su cui giace il vettore
- **Verso**: specifica in che direzione punta il vettore

Se abbiamo due vettori u e v possiamo definire la somma che sarà un vettore $w = u+v$ ottenuto mediante la **regola del parallelogramma**.
Dato un numero $\alpha \in \mathbb{R}$, per ogni vettore v, possiamo definire il vettore αv è la freccia ottenuta moltiplicando v per α in modulo e lasciando invariata la direzione se $\alpha > 0$ e invertendo il verso se $\alpha < 0$. Questa operazione è detta **moltiplicazione per scalare**. Se $\alpha = -1$, otteniamo il vettore $-v$ che ha stesso modulo, stessa direzione ma verso opposto a v.
L'insieme di tutti i vettori del piano è allora uno spazio vettoriale reale V chiuso rispetto all'operazione di combinazione lineare:

$$u = \alpha v + \beta w$$

Per ogni vettore u e $v \in V$ e per ogni $\alpha, \beta \in \mathbb{R}$.