70-classic-and-quantum-03

Quantum Computing

Computazioni classiche e computazioni quantum

1

1

ogni quantum circuit può essere *approssimato* usando un numero limitato di gate appartenenti a un piccolo insieme di tipi di gate

2

2

70-classic-and-quantum-03

un ulteriore gate

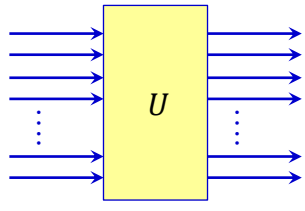
- si chiama T
- la sua matrice è $\mathrm{T} = \begin{pmatrix} e^{i\pi/8} & 0 \\ 0 & e^{-i\pi/8} \end{pmatrix}$
- esegue *rotazioni* di $\frac{\pi}{8}$
- c'è bisogno di un gate con una matrice con numeri complessi ….

3

3

un circuito quantum qualunque

- consideriamo un qualunque circuito quantum U descritto da una matrice $d \times d$ e sia ε una costante

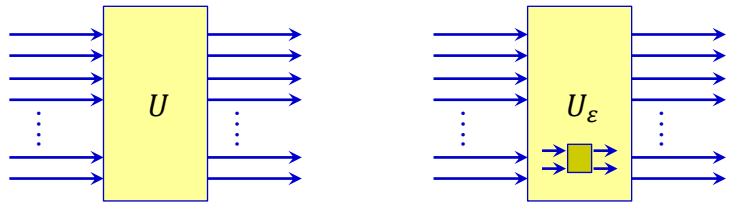


4

4

un circuito quantum qualunque

- è possible realizzare un circuito quantum U_ε composto da soli gate di tipo $CNOT$, H, X, Z e T che *approssima* il comportamento di U
  - visto che la matrice che descrive U contiene numeri complessi qualunque, U_ε può non essere *identico* a U

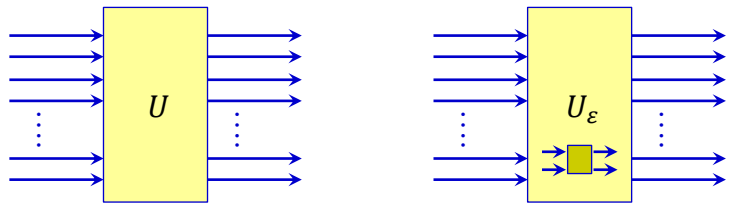


5

5

un circuito quantum qualunque

- il circuito U_ε contiene $O\left(d^2 \log \frac{1}{\varepsilon}^3\right)$ gate e $\|U - U_\varepsilon\| \leq \varepsilon$
  - se diamo lo stesso stato in input ai due circuiti i loro output sono molto vicini in termini di distanza euclidea



6

6

70-classic-and-quantum-03

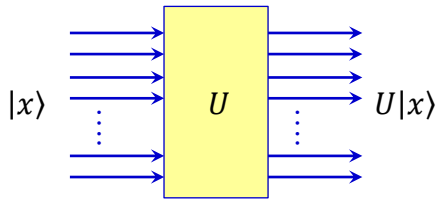
è possibile simulare qualunque circuito classico utilizzando un quantum circuit

7

7

reversibilità del calcolo

- supponiamo che esista un quantum circuit che calcoli una certa funzione U
- immaginiamo che abbia i seguenti input e output:



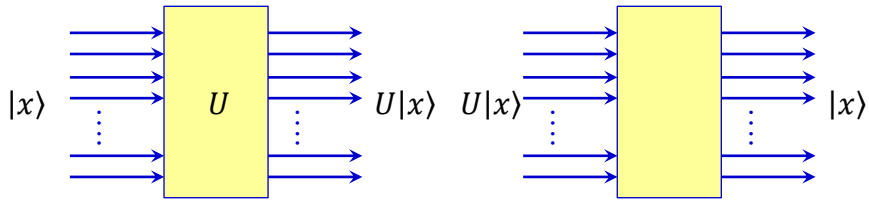
8

8

70-classic-and-quantum-03

reversibilità del calcolo

- allora esiste un quantum circuit che calcola la funzione inversa

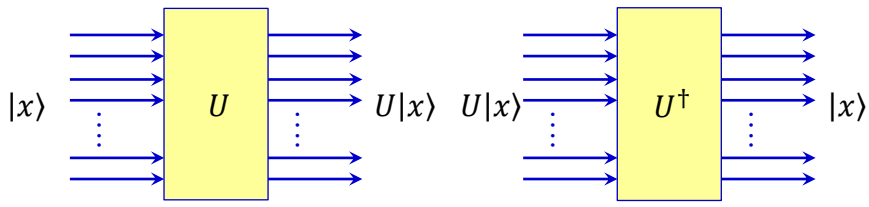


9

9

reversibilità del calcolo

- ed è il circuito che calcola la funzione $U^\dagger$
- infatti abbiamo che $UU^\dagger = U^\dagger U = I$



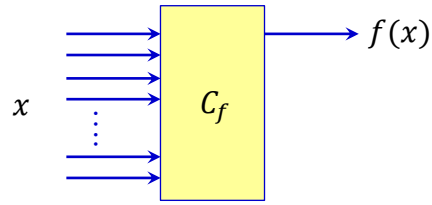
10

10

70-classic-and-quantum-03

circuiti booleani classici

- i circuiti booleani *classici* (non-quantum) che conosciamo, tipicamente non sono invertibili
- pensiamo ad esempio a un circuito tradizionale C_f che calcoli $f: \{0,1\}^n \to \{0,1\}$

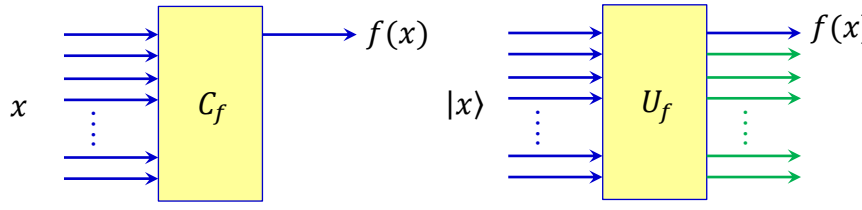


11

11

versione quantum?

- come realizzare C_f con un quantum circuit? una versione quantum U_f di C_f ?
- un bit di output sarà $f(x)$, ma abbiamo bisogno almeno di altri $n-1$ bit come condizione necessaria di invertibilità



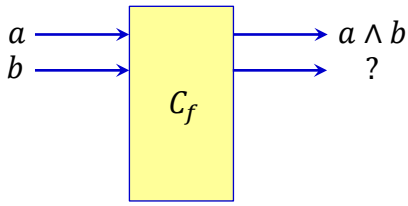
12

12

70-classic-and-quantum-03

versione quantum di un and

- pensiamo ad esempio ad un *and*, dove $a \wedge b = 1 \iff a = b = 1$, ovviamente dato il valore di $a \wedge b$ non è possibile ricostruire i singoli valori di a e di b
- ci riusciamo se in output aggiungiamo un bit?

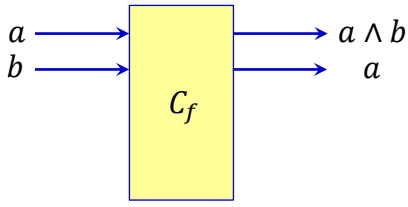


13

13

un bit in più in output non basta

- ci riusciamo se in output aggiungiamo un bit?
- proviamo ad aggiungere a in output
  - purtroppo non è sufficiente: se $a \wedge b = 0$ e $a = 0$ non sappiamo se b vale 0 o 1



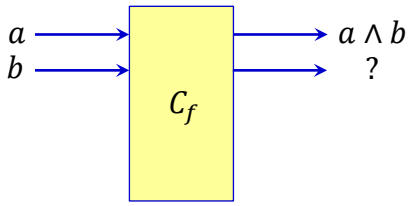
14

14

70-classic-and-quantum-03

un bit in più in output non basta

- ci riusciamo se in output aggiungiamo un bit?
- possiamo provare anche altre scelte (es. aggiungiamo in output $a \vee b$), ma non c'è modo
  - un bit in più non porta sufficiente informazione

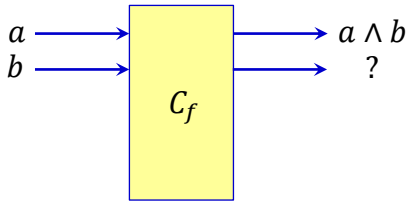


15

15

un bit in più in output non basta

- un bit in più non porta sufficiente informazione
- infatti ci sono tre possibili combinazioni di a e b che hanno come risultato $a \wedge b = 0$ ma con un bit in più posso distinguere solo due casi



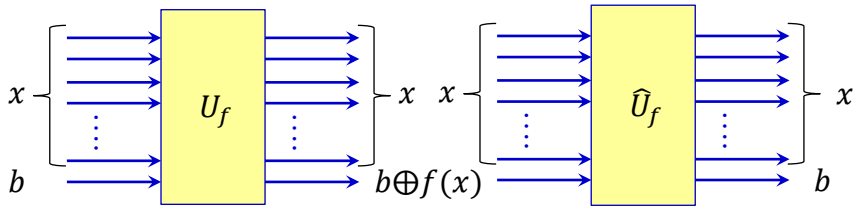
16

16

70-classic-and-quantum-03

una scelta semplice su cui ragionare

- inserire x in input e in output
- aggiungere in input un *answer bit* b (per memorizzare il risultato), che possiamo assumere 0
- mettere in uscita $f(x)$ in xor con b



17

17

alcune porte semplici – not e cnot

- abbiamo già visto l'esempio dell'operatore X che è l'analogo quantum del *not* classico ed abbiamo visto che è invertibile
- abbiamo anche visto l'operatore invertibile $CNOT$

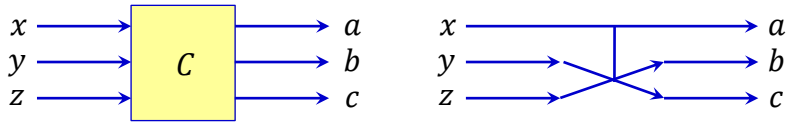
18

18

70-classic-and-quantum-03

come realizzare un and

- consideriamo l'operatore $C - SWAP$
  - $x = 0 \Longrightarrow b = y, c = z$,
  - $x = 1 \Longrightarrow b = z, c = y$ e
  - $a = x$
- il valore di x stabilisce se y e z devono essere invertiti

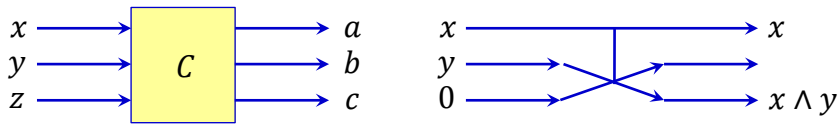


19

19

come realizzare un and

- il valore di x stabilisce se y e z devono essere invertiti
- usiamo $C - SWAP$ per realizzare un *and* tra x e y fissando $z = 0$
  - se $x = 0, y = 0 \Longrightarrow c = 0$, se $x = 0, y = 1 \Longrightarrow c = 0$, se $x = 1, y = 0 \Longrightarrow c = 0$, se $x = 1, y = 1 \Longrightarrow c = 1$



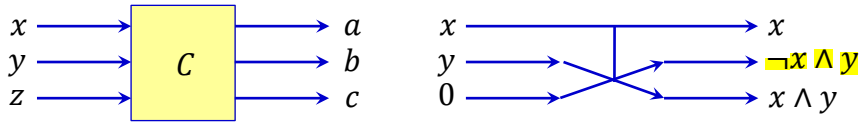
20

20

70-classic-and-quantum-03

come realizzare un and

- diamo uno sguardo a b
- abbiamo (fissando $z = 0$)
  - se $x = 0, y = 0 \Longrightarrow b = 0$, se $x = 0, y = 1 \Longrightarrow b = 1$, se $x = 1, y = 0 \Longrightarrow b = 0$, se $x = 1, y = 1 \Longrightarrow b = 0$

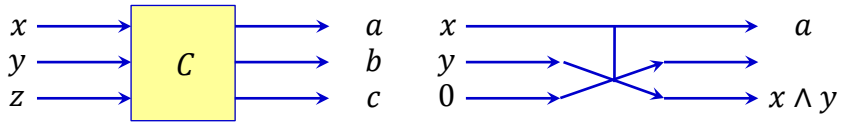


21

21

come realizzare un and

- è facile dimostrare che $C - SWAP$ è invertibile



22

22

70-classic-and-quantum-03

and e not bastano

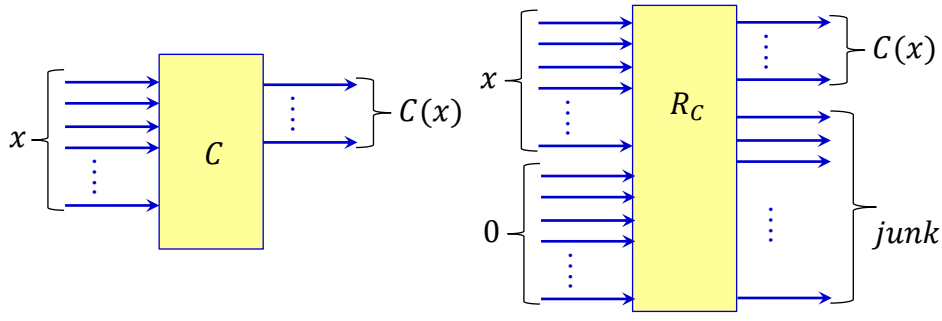
- ricordiamo che per realizzare qualunque circuito classico bastano *and* e *not*
- ora che sappiamo fare un *and* e un *not* possiamo simulare un circuito classico con un quantum-circuit; dobbiamo:
  - rimpiazzare ogni *and* con un $C - SWAP$,
  - aggiungere un bit di input a 0 per ogni $C - SWAP$
  - ricordarci che per ogni $C - SWAP$ c'è un bit di output *inutile*

23

23

uno schema generale

- per realizzare qualunque circuito classico C con un quantum-circuit R_C possiamo fare:



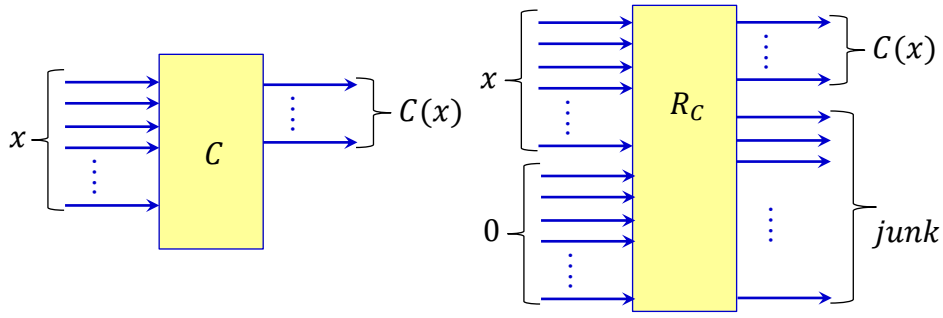
24

24

70-classic-and-quantum-03

uno schema generale

- dato che tutte le parti che costituiscono R_C sono unitary transformation anche R_C è una unitary transformation

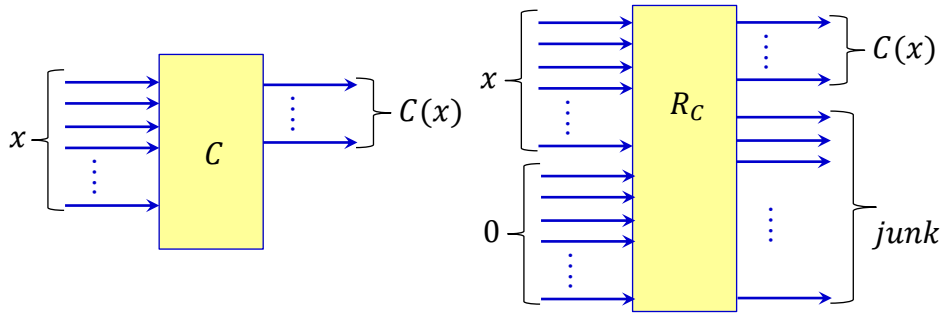


25

25

uno schema generale

- l'input non è più una sequenza di bit e di zero tradizionali ma piuttosto è una superposition $\sum_x \alpha_x |x\rangle |0\rangle$



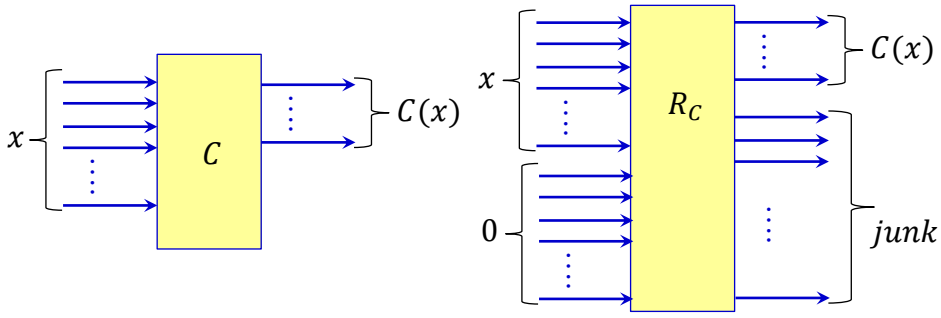
26

26

70-classic-and-quantum-03

uno schema generale

- l'output è una superposition $\sum_x \alpha_x |C(x)\rangle \ |junk\rangle$
- ovviamente ci piacerebbe eliminare la componente spazzatura

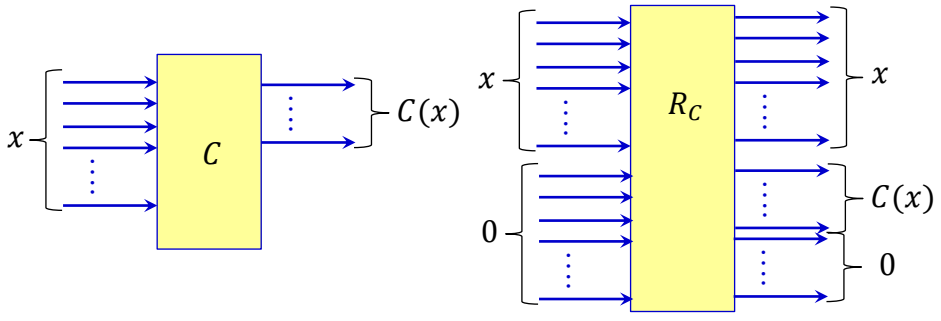


27

27

uno schema generale

- ci piacerebbe un circuito del tipo:



28

28

70-classic-and-quantum-03

perché eliminare junk?

- soprattutto perché comunque *interferisce* con la computazione

29

29

come eliminare junk?

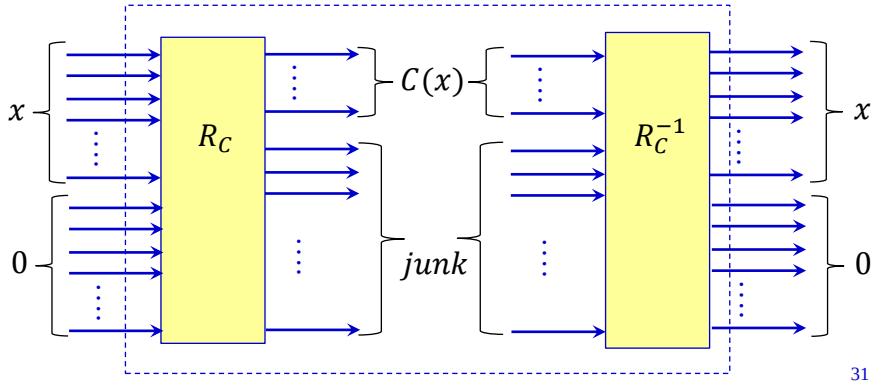
- possiamo semplicemente cancellarlo? no, non avremmo più unitary transformation
- possiamo buttarlo via? no, ovunque lo mettiamo può essere entangled con il resto
  - anche se lo inviamo nello spazio

30

30

una soluzione elegante, ma poco furba

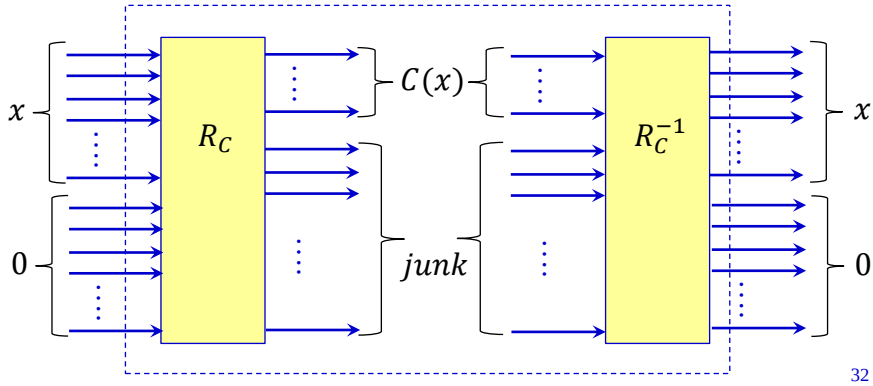
- possiamo prendere il circuito e, dato che la trasformazione è reversibile possiamo farne una copia e applicarlo al contrario



31

una soluzione elegante, ma poco furba

- abbiamo eliminato il $junk$ in output, ma ci siamo persi il risultato

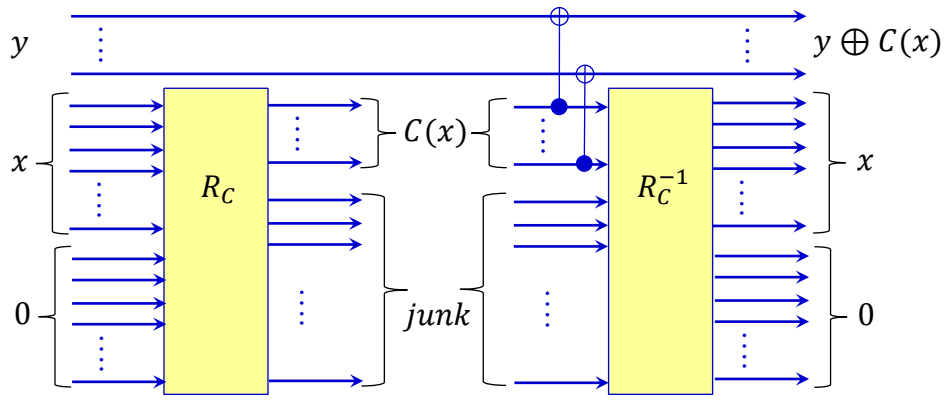


32

70-classic-and-quantum-03

patch per salvare il risultato

- usiamo dei $CNOT$ per copiare il risultato

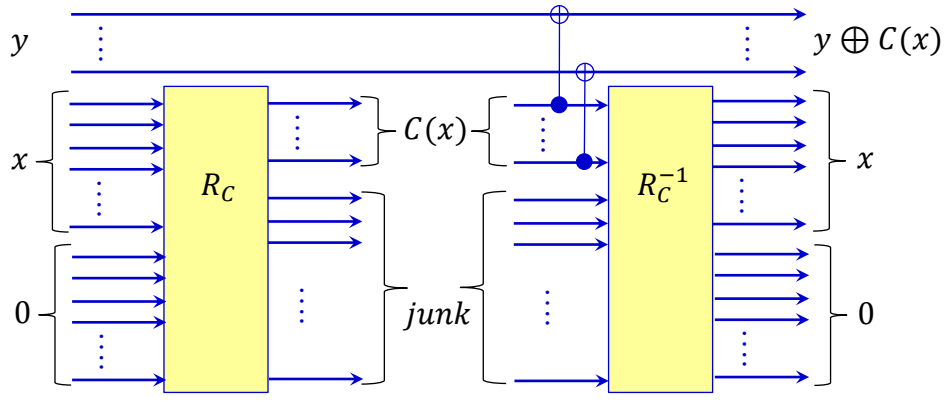


33

33

in sintesi

- tutto ciò che facciamo con un circuito classico possiamo farlo con un quantum circuit

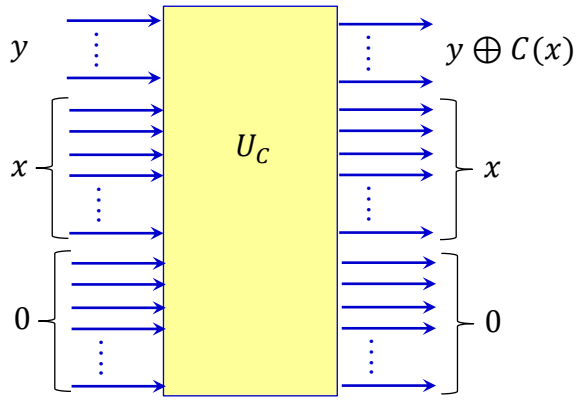


34

34

in sintesi

- abbiamo ottenuto uno schema generale di circuito

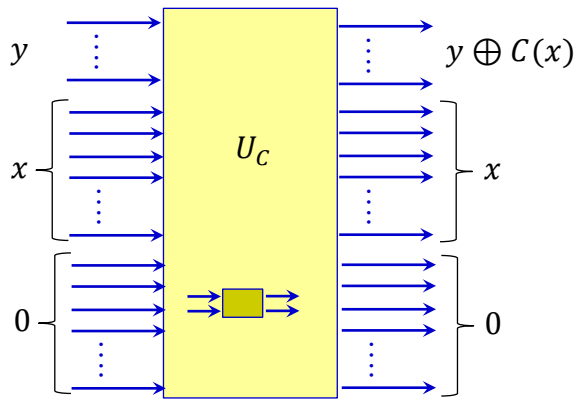


35

35

cosa c'è dentro?

- tanti circuiti collegati tra loro, ciascuno con un numero costante di input e di output



36

36