

Architettura degli Elaboratori

7 - Reti Sequenziali

Zeynep KIZILTAN

Dipartimento di Scienze dell'Informazione
Università degli Studi di Bologna

Anno Accademico 2007/2008

Notizie

- ▶ Il parziale si svolgerà nelle aule "Cremona", "Pincherle" e "VII Piano" (Dipartimento di Matematica) alle ore 9.00.
- ▶ Sul sito web del corso, è possibile determinare in quale delle due aule recarsi.
- ▶ Alcune regole importanti riguardanti la prova scritta:
- ▶ Occorre scrivere in bella calligrafia.
- ▶ Occorre riportare in modo chiaro tutti i passi che portano alla determinazione del risultato.
- ▶ Durante la prova scritta:
 - ▶ non è concesso chiedere alcunchè ai docenti e agli altri studenti.
 - ▶ non è consentito consultare libri, appunti o altro materiale didattico.
 - ▶ non è consentito utilizzare calcolatrici, telefoni cellulari o altri dispositivi elettronici.

Clock

- ▶ In molti circuiti digitali l'ordine degli eventi è cruciale.
 - ▶ In alcuni casi, un evento deve precederne un altro, in altre situazioni due eventi devono avvenire allo stesso tempo.
- ▶ Quindi, molti circuiti digitali utilizzano un **clock** per sincronizzarsi e per ottenere le informazioni temporali desiderate.
- ▶ Un clock è un circuito che emette una serie di impulsi di larghezza definita.
- ▶ L'intervallo temporale compreso tra le estremità di due impulsi consecutivi è detto **ciclo di clock**.
- ▶ La frequenza degli impulsi è in genere compresa tra 1 e 500 Mhz, corrispondenti a cicli di clock compresi tra 1000 e 2 nanosecondi.

Diagramma di Temporizzazione

- ▶ In un calcolatore, più eventi possono verificarsi durante uno stesso ciclo di clock.
- ▶ Se è necessario che si verifichino in un specifico ordine, occorre dividere il ciclo di clock in sottocicli.
- ▶ Un tecnica comune consiste nello:
 - ▶ intercettare la linea di clock principale;
 - ▶ inserire un circuito con un ritardo.
- ▶ In questo modo, viene generato un secondo segnale di clock la cui fase è traslata rispetto a quello originale.

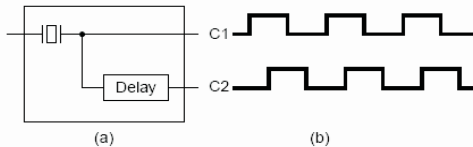
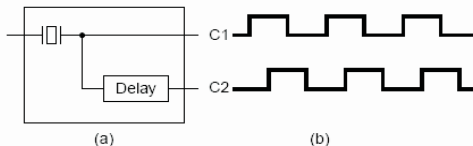


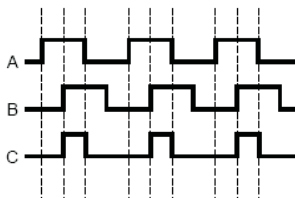
Diagramma di Temporizzazione

- ▶ Le digramma di temporizzazione nella figura fornisce quattro diversi riferimenti temporali utilizzabili per sincronizzare eventi:
 - ▶ fronte di salita di C1;
 - ▶ fronte di discesa di C1;
 - ▶ fronte di salita di C2;
 - ▶ fronte di discesa di C2.
- ▶ Associando diversi eventi ai quattro fronti è possibile stabilire un ordine tra loro.
- ▶ Più di quattro riferimenti temporali possono essere ottenuti collegando al clock principale altre linee secondarie.



Clock Asimmetrico

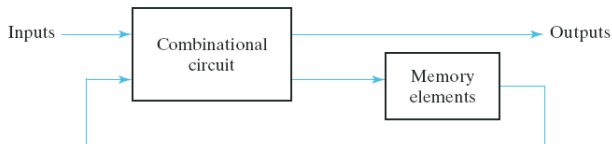
- ▶ I clock sono simmetrici:
 - ▶ il tempo speso nello stato alto è uguale al tempo speso nello stato basso.
- ▶ Un clock asimmetrico può essere ottenuto calcolando l'AND tra il clock originale e quello la cui fase è traslata.



(c)

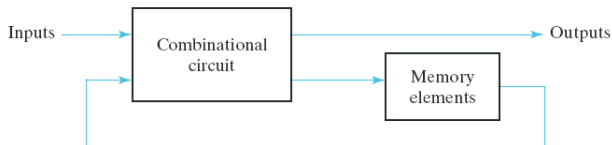
Reti Sequenziali

- ▶ Le reti finora considerate sono le reti combinatorie.
- ▶ Però, molti sistemi digitali includono:
 - ▶ sia parti realizzate con circuiti combinatori;
 - ▶ sia elementi di memoria descritti come circuiti sequenziali.
- ▶ Gli elementi di memoria:
 - ▶ sono in grado di immagazzinare informazioni binarie;
 - ▶ definiscono lo stato del circuito sequenziale.



- ▶ Nella figura, è mostrato il diagramma a blocchi di tale sistema:
 - ▶ il circuito combinatorio e gli elementi di memoria sono interconnessi tra loro ad anello.

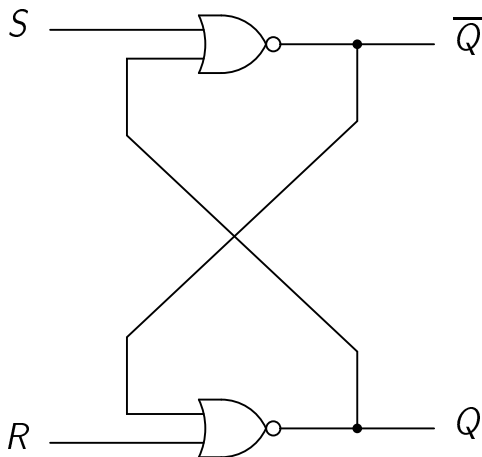
Reti Sequenziali



- ▶ Un circuito sequenziale:
 - ▶ riceve informazioni binarie dall'ambiente esterno, attraverso gli ingressi;
 - ▶ insieme allo stato presente memorizzato negli elementi di memoria, determina il valore delle uscite e lo stato futuro.
- ▶ Quindi, l'uscita di un circuito sequenziale è funzione:
 - ▶ sia degli ingressi;
 - ▶ sia dello stato presente.
- ▶ Di conseguenza, l'analisi e la sintesi dei circuiti sono molto più complesse nel caso sequenziale che nel caso combinatorio.

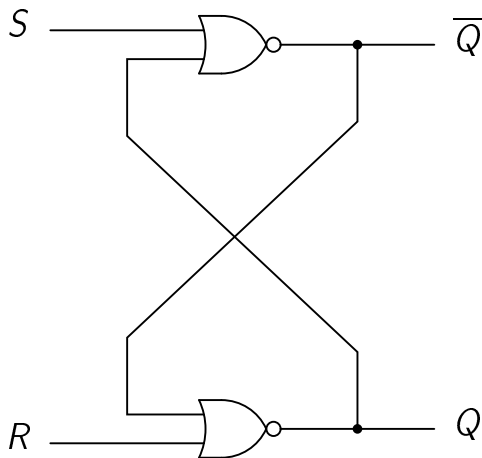
Latch SR

- ▶ Per creare una memoria a 1 bit è necessario disporre di un circuito che può ricordare i precedenti valori di input.
- ▶ Si può costruire tale circuito usando le porte NOR, come illustrato nella figura.



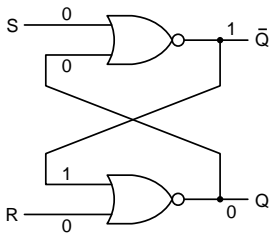
Latch SR

- ▶ Questo circuito è chiamato **latch SR** e ha:
 - ▶ due ingressi, S per impostare (setting) il valore del latch e R per azzerarlo (resetting);
 - ▶ due uscite, Q e \overline{Q} che sono complementari l'una rispetto all'altra.
- ▶ Diversamente dalle reti combinatorie, l'output di un latch non è determinato unicamente dai valori degli ingressi.
- ▶ Ora vediamo come ciò avviene...

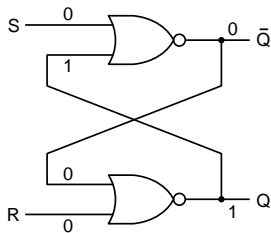


Latch SR - Stati Stabili

- ▶ Assumiamo che $S = R = 0$ e $Q = 0$ (Figura (a)).
- ▶ Dato che Q viene reinserito come input della porta NOR di sopra, l'uscita $\bar{Q} = 1$.
- ▶ Questo 1 è riutilizzato come input della porta NOR di sotto, producendo come output $Q = 0$.
- ▶ Quindi, questo stato è **stabile** (nel senso che, i valori di verità nei fili sono in accordo con il comportamento delle porte).



(a)



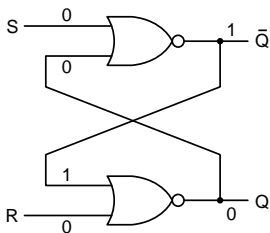
(b)

A	B	NOR
0	0	1
0	1	0
1	0	0
1	1	0

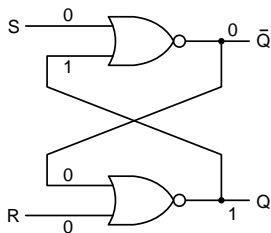
(c)

Latch SR - Stati Stabili

- ▶ Immaginiamo ora che $S = R = 0$ e $Q = 1$ (Figura (b)).
- ▶ La porta XOR di sopra ha come ingressi 0 e 1.
- ▶ La sua uscita \overline{Q} vale 0.
- ▶ Questo 0 è riutilizzato come input della porta NOR di sotto, producendo come output $Q = 1$.
- ▶ Quindi, anche questo stato è stabile.



(a)



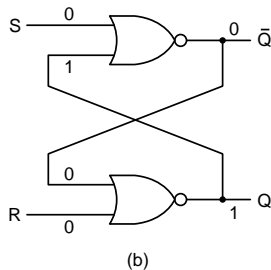
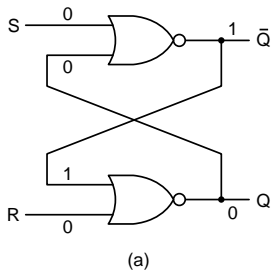
(b)

A	B	NOR
0	0	1
0	1	0
1	0	0
1	1	0

(c)

Latch SR - Stati Stabili

- ▶ Un stato in cui $Q = \overline{Q} = 0$ è invece instabile:
 - ▶ in tal caso, le porte NOR produrrebbero come risultato 1, non 0.
- ▶ Analogamente, è impossibile che $Q = \overline{Q} = 1$:
 - ▶ ciò forzerebbe 0 alle uscite delle porte NOR.
- ▶ Concludiamo che quando $R = S = 0$, il latch ha 2 stati stabili, 0 e 1, in base al valore di Q .

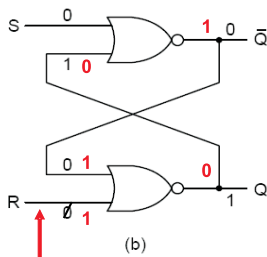
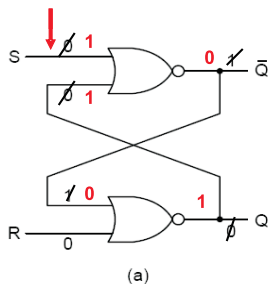


A	B	NOR
0	0	1
0	1	0
1	0	0
1	1	0

(c)

Latch SR - Stati Stabili

- ▶ Esaminiamo ora gli effetti dei valori dell'input sullo stato.
- ▶ Supponiamo che S diventi 1, mentre $R = 0$ e $Q = 0$ (Figura (a)).
- ▶ L'uscita della porta NOR di sopra diventa 0 che forza $\overline{Q} = 0$.
- ▶ Questo cambiamento è propagato alla porta NOR di sotto, producendo come output $Q = 1$.
- ▶ Quindi, settando S (cioè impostandone il valore 1), si può far passare lo stato da 0 a 1.

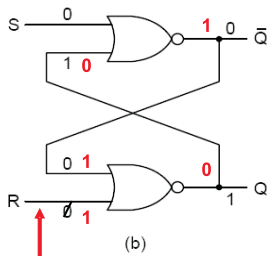
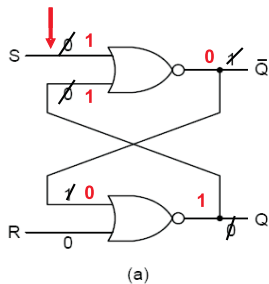


A	B	NOR
0	0	1
0	1	0
1	0	0
1	1	0

(c)

Latch SR - Stati Stabili

- ▶ Supponiamo che R diventi 1, mentre $S = 0$ e $Q = 0$.
- ▶ Settando R non si produce alcun effetto:
 - ▶ in tal caso, l'uscita della porta NOR di sotto non cambia.

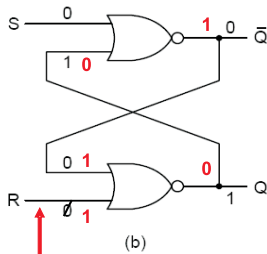
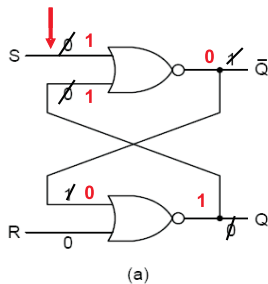


A	B	NOR
0	0	1
0	1	0
1	0	0
1	1	0

(c)

Latch SR - Stati Stabili

- ▶ Applicando un ragionamento simile, si può facilmente verificare che:
 - ▶ impostare $R = 1$ quando $Q = 1$ modifica lo stato del latch, portandolo nello stato $Q = 0$ (Figura (b));
 - ▶ impostare $S = 1$ quando $Q = 1$ non produce alcun effetto.



A	B	NOR
0	0	1
0	1	0
1	0	0
1	1	0

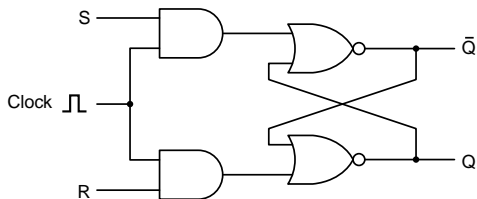
(c)

Latch SR

- ▶ Riassumendo:
 - ▶ lo **stato interno** del latch è rappresentato dal singolo valore binario Q .
 - ▶ quando S è impostato a 1, lo stato del latch diventa $Q = 1$, indipendentemente dallo stato precedente;
 - ▶ quando R è impostato a 1, lo stato del latch diventa $Q = 0$, indipendentemente dallo stato precedente.
- ▶ In altre parole, il circuito ricorda quale tra S e R è stato settato per ultimo.
- ▶ Utilizzando questa proprietà, è possibile costruire le memorie dei calcolatori.

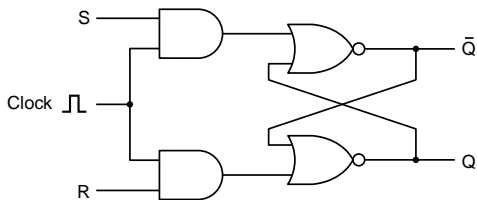
Latch SR Temporizzato

- ▶ Spesso è utile impedire che un latch cambi stato, se non in specifici momenti.
- ▶ Un circuito che gode questa caratteristica è detto **latch SR temporizzato**.



- ▶ Questo circuito ha un ulteriore ingresso C per controllare il suo funzionamento.
- ▶ C è in genere collegato ad un clock, ma questo non è obbligatorio:
 - ▶ i termini **enable** e **strobe** sono largamente utilizzati per indicare che l'input clock vale 1.

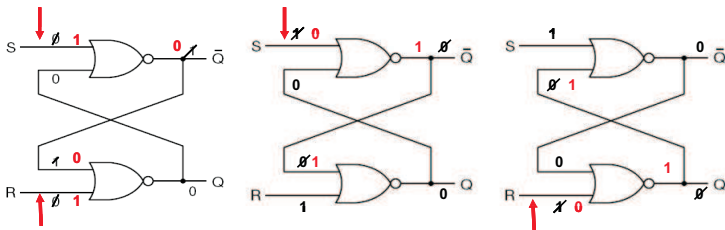
Latch SR Temporizzato



- ▶ Quando $C = 0$, entrambe le porte AND generano in output il valore 0. Quindi:
 - ▶ lo stato interno del latch non cambia;
 - ▶ in particolare, il latch non è sensibile alle variazioni degli ingressi S e R .
- ▶ Quando $C = 1$, le porte AND non bloccano più i segnali S e R . Quindi:
 - ▶ lo stato interno del latch SR può cambiare liberamente, seguendo le regole che conosciamo già.

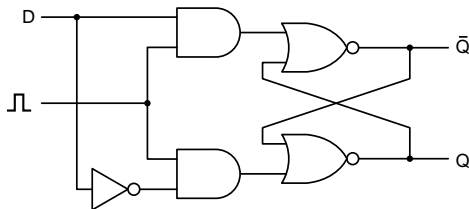
Latch SR Temporizzato

- ▶ Non abbiamo ancora parlato degli stati in cui $S = R = 1$.
- ▶ L'unico stato consistente per $S = R = 1$ è quando $Q = \bar{Q} = 0$.
- ▶ Però, il circuito diventa non deterministico finché sia S sia R non tornino assumere il valore 0:
 - ▶ lo stato stabile è determinato da quale tra S e R torna a 0 per primo.
 - ▶ se entrambi ritornano a 0 nello stesso istante, lo stato stabile è determinato in modo del tutto casuale.

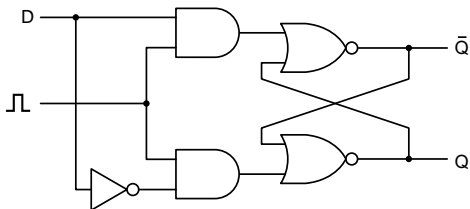


Latch D Temporizzato

- ▶ Un modo per risolvere questa ambiguità è evitare che si verifichi.
- ▶ La figura mostra un circuito che ha solo un input D .
- ▶ Siccome un ingresso è sempre il complemento dell'altro, non accade mai che entrambi gli ingressi valgano 1.
- ▶ Tale circuito sequenziale è chiamato **latch D temporizzato**.



Latch D Temporizzato



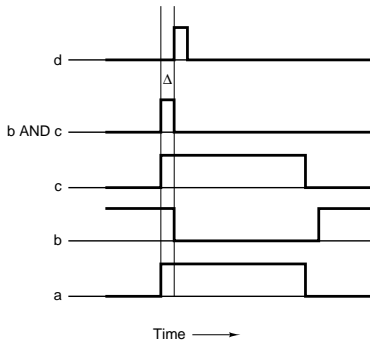
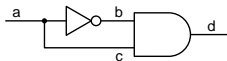
- ▶ Quando $D = 1$ e $C = 1$, il latch viene portato allo stato $Q = 1$.
- ▶ Quando $D = 0$ e $C = 1$, il latch viene portato allo stato $Q = 0$.
- ▶ In altre parole, quando il circuito è abilitato, il valore corrente di D è memorizzato nel latch.
- ▶ Quindi, il latch D temporizzato è una memoria a 1 bit in cui il valore memorizzato è sempre disponibile sulla linea Q .

Flip-Flop

- ▶ In molti circuiti è necessario memorizzare il valore binario in input in un particolare istante, piuttosto che in un intervallo.
- ▶ In tali circuiti, chiamati **flip-flop**, la transizione di stato non si verifica quando il clock vale 1:
 - ▶ ma invece quando il clock cambia il suo valore da 0 a 1 oppure da 1 a 0.
- ▶ Nel primo caso parliamo di **fronte di salita** o **positive edge** e di flip-flop **positive-edge triggered**.
- ▶ Nel secondo caso parliamo invece di **fronte di discesa** o **negative edge** e di flip-flop **negative-edge triggered**.
- ▶ In questo senso, si dice che:
 - ▶ i latch temporizzati sono circuiti sequenziali a **commutazione di livello** (level triggered);
 - ▶ i flip-flop sono circuiti sequenziali a **commutazione di fronte** (edge triggered).

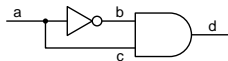
Flip-Flop

- ▶ Un modo semplice per progettare un flip-flop è quello di:
 - ▶ generare un impulso molto breve in corrispondenza del fronte di salita di un clock;
 - ▶ immettere tale impulso in un latch D.
- ▶ La figura mostra il circuito che implementa questa soluzione.

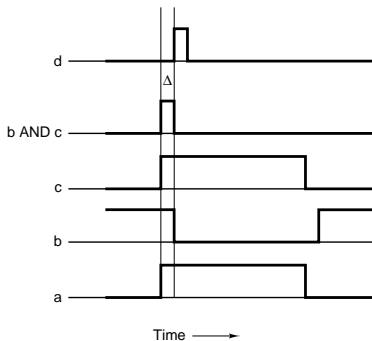


Flip-Flop

- ▶ A prima vista potrebbe sembrare che l'uscita della porta AND debba essere sempre 0, dato che gli ingressi sono 0 e 1.
- ▶ In realtà, l'invertitore induce un piccolo, ma non nullo, ritardo di propagazione che permette la realizzazione dell'impulso.
- ▶ Supponiamo di misurare la tensione in quattro punti *a*, *b*, *c* e *d*.



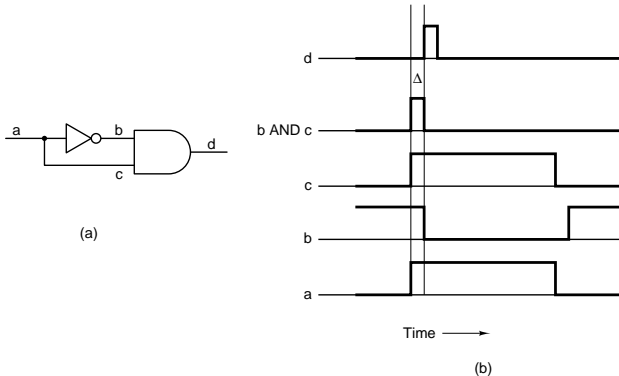
(a)



(b)

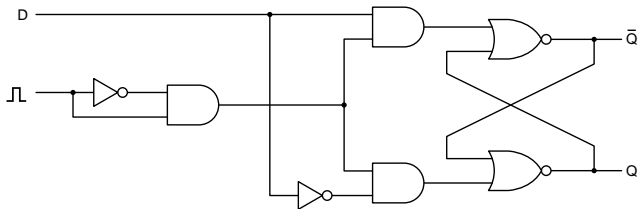
Flip-Flop

- ▶ Il segnale d'ingresso, misurato in a , è un impulso lungo.
- ▶ Il segnale b è invertito e sfasato di un ritardo di pochi nanosecondi.
- ▶ Il segnale c è considerato identico al segnale a .
- ▶ Gli segnali b e c nella porta AND producono un impulso di breve durata.
- ▶ La largehezza Δ dell'impulso è uguale al ritardo dell'invertitore.
- ▶ L'uscita della porta AND (d) corrisponde all'impulso sfasato del ritardo interno alla porta.



Flip-Flop D Positive-edge Triggered

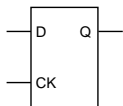
- ▶ Quindi, il latch D verrà attivato con un ritardo fisso rispetto al fronte di salita del clock.
- ▶ Tuttavia ciò non ha alcun effetto sulla durata dell'impulso.
- ▶ Il circuito finale del flip-flop D è mostrato nella figura.



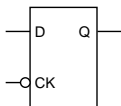
- ▶ Il flip-flop D può essere realizzato anche in un altro modo, utilizzando un latch D e un latch SR.
- ▶ In generale, circuiti con lo stesso comportamento possono essere realizzati in modo diverso.

Latch D e Flip-Flop D: Simboli Grafici

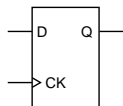
- ▶ La figura mostra i simboli usati per i latch e i flip-flop D:
 - ▶ La prima indica un latch il cui stato viene caricato quando il clock vale 1.
 - ▶ La seconda indica un latch il cui stato viene caricato quando il clock vale 0.
 - ▶ La terza indica un flip-flop positive-edge triggered.
 - ▶ La quarta indica un flip-flop negative-edge triggered.
- ▶ Molti latch e flip-flop hanno anche l'uscita \overline{Q} .
- ▶ Alcuni sono dotati di due ingressi aggiuntivi *Set* e *Reset* che forzano lo stato a $Q = 1$ e $Q = 0$, rispettivamente.



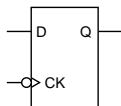
(a)



(b)



(c)



(d)

Flip-Flop JK

- ▶ Esiste poi un altro flip-flop, chiamato **flip-flop JK** che, oltre all'ingresso per il clock, ha due ingressi J e K .
- ▶ Il fronte di salita (o di discesa) del clock modifica lo stato interno come segue:
 - ▶ Quando $J = K = 0$, lo stato interno **non cambia**.
 - ▶ Quando $J = K = 1$, lo stato interno **cambia**.
 - ▶ Quando $J = 0$ e $K = 1$, lo stato interno diventa 0.
 - ▶ Quando $J = 1$ e $K = 0$, lo stato interno diventa 1.
- ▶ La figura mostra il simbolo grafico per il flip-flop JK.

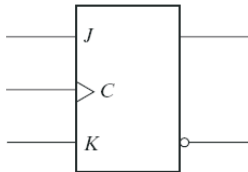


Tabella Caratteristiche

- ▶ Il comportamento di un flip-flop viene descritto dalla sua **tabella caratteristica**.
- ▶ Una **tabella caratteristica** elenca per ogni assegnamento di valori di verità agli ingressi, il valore dello stato futuro $Q(t + 1)$ nei termini dello stato presente $Q(t)$.
 - ▶ t si riferisce all'istante di applicazione dell'impulso di clock.
- ▶ E.g.,

Flip-flop D

D	$Q(t + 1)$
0	0
1	1

Lo stato futuro di un flip-flop D dipende soltanto dall'ingresso D , ed è indipendente dallo stato presente.

Tabella Caratteristiche

- ▶ Il comportamento di un flip-flop viene descritto dalla sua **tabella caratteristica**.
- ▶ Una **tabella caratteristica** elenca per ogni assegnamento di valori di verità agli ingressi, il valore dello stato futuro $Q(t + 1)$ nei termini dello stato presente $Q(t)$.
 - ▶ t si riferisce all'istante di applicazione dell'impulso di clock.
- ▶ E.g.,

Flip-flop JK

J	K	$Q(t + 1)$
0	0	$Q(t)$
0	1	0
1	0	1
1	1	$\overline{Q(t)}$

Lo stato futuro di un flip-flop JK dipende dagli ingressi J e K .

Equazioni Caratteristiche

- ▶ Le proprietà logiche di un flip-flop possono essere espresse algebricamente tramite le equazioni caratteristiche.
- ▶ Tale equazioni sono derivate facilmente dalle tabelle caratteristiche.
- ▶ Per il flip-flop D, $Q(t + 1) = D$
- ▶ Per il flip-flop JK:

$$\begin{aligned}Q(t + 1) &= \overline{J} \overline{K} Q(t) + J \overline{K} + JK \overline{Q(t)} \\ &= \overline{JQ(t)} + \overline{K} Q(t) \\ &= J\overline{Q} + \overline{K}Q\end{aligned}$$

Flip-flop JK

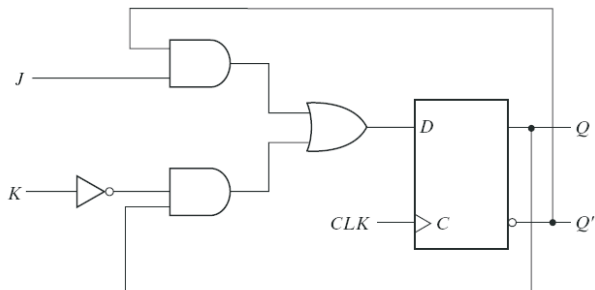
J	K	$Q(t + 1)$
0	0	$Q(t)$
0	1	0
1	0	1
1	1	$\overline{Q(t)}$

Flip-flop D

D	$Q(t + 1)$
0	0
1	1

L'implementazione del flip-flop JK

- ▶ Possiamo realizzare un flip-flop JK usando un flip-flop D e delle porte esterne, come mostrato nella figura.
- ▶ L'ingresso al flip-flop D è $J\bar{Q} + \bar{K}Q$.
- ▶ Siccome $Q(t+1) = D$ in un flip-flop D, abbiamo $Q(t+1) = J\bar{Q} + \bar{K}Q$.



Circuiti Sequenziali Sincroni

- ▶ Le reti sequenziali di cui ci occuperemo con maggior dettaglio sono i **circuiti sequenziali sincroni**, che sono costituiti da:
 - ▶ alcuni **flip-flop**, regolati dallo stesso clock;
 - ▶ una **rete combinatoria** che, a partire dal valore degli ingressi e dello stato presente, calcoli il valore delle uscite e dello stato futuro.
- ▶ Graficamente:

