Gli elementi di memoria: i bistabili

Circuiti sequenziali

- Nei circuiti sequenziali il valore delle uscite in un determinato istante dipende sia dal valore degli ingressi in quello stesso istante sia dal tempo
  - Una stessa configurazione in ingresso applicata in due istanti di tempo successivi può produrre due valori d'uscita differenti

- Un circuito sequenziale ha memoria degli eventi passati e, quindi, richiede degli elementi in grado di conservare informazioni
  - L'informazione contenuta da questa memoria in un generico istante *t* rappresenta il concetto di "stato" corrente

Circuiti sequenziali e bistabili

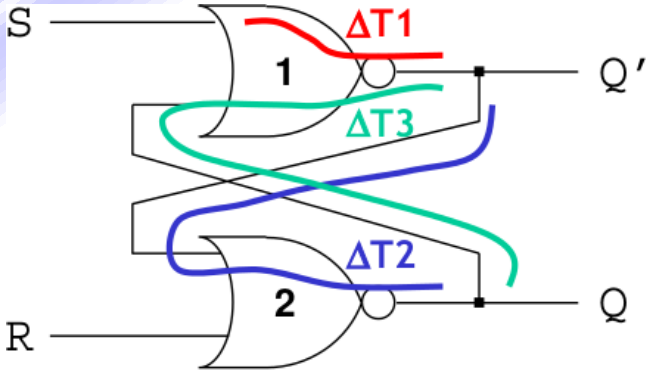
- Gli elementi in grado di conservare informazioni sono detti bistabili
  - Il termine bistabile deriva dal fatto che tale elemento è stabile in due stati (0 e 1)
- NOTA: I bistabili sono *volatili*, cioè rispettano quanto indicato solo se sono alimentati
- La differenza principale tra i vari tipi di elementi di memoria è costituita da:
  - Numero di ingressi dell'elemento di memoria
  - Modo in cui gli ingressi ne determinano lo stato

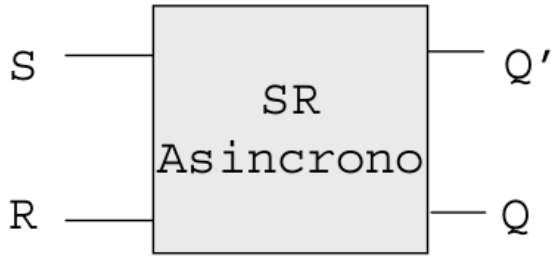
Bistabili: classificazione

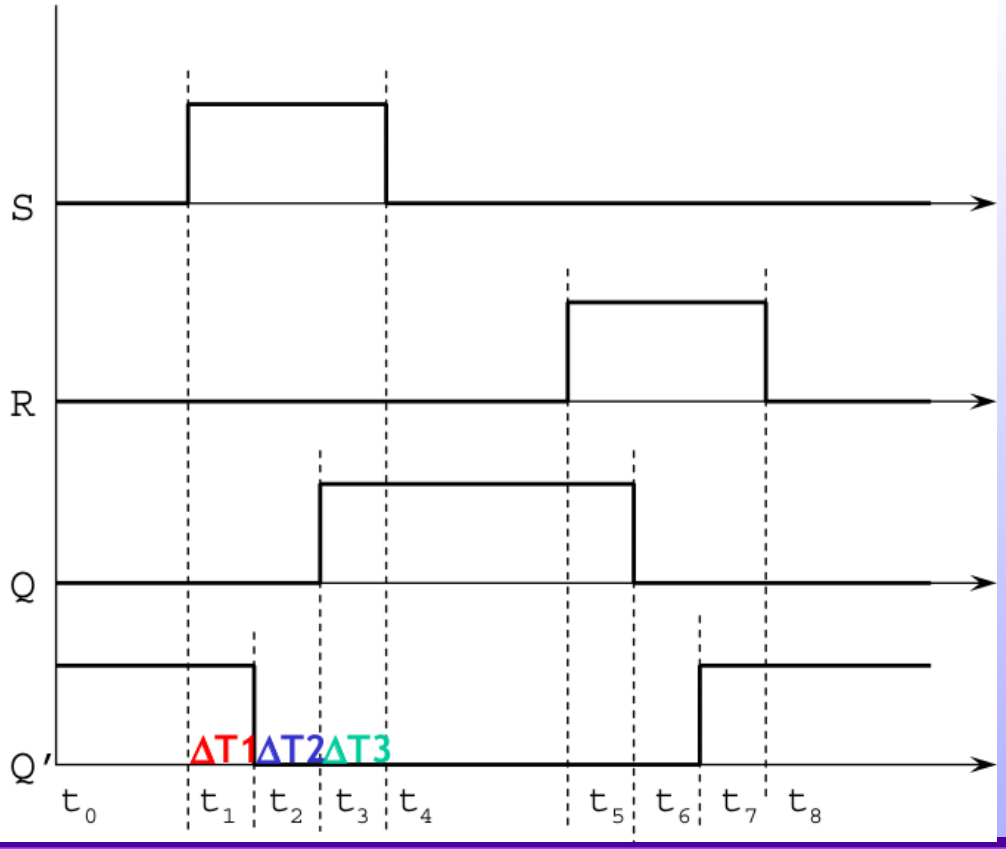
- Asincroni
  - Sono privi di un segnale di sincronizzazione
  - Modificano il loro stato rispondendo direttamente ad eventi sui segnali di ingresso
- Sincroni
  - Sono sensibili ad un segnale di controllo (spesso il clock)
  - La transizione da uno stato all'altro avviene solo in corrispondenza di un impulso del segnale di controllo
  - Ulteriore classificazione:
    - Bistabili sincroni controllati (gated latch)
    - Flip-flop
      - Master-slave (a livello o pulse-triggered)
      - Edge triggered (a fronte)

Bistabili asincroni: SR

- Il bistabile asincrono più semplice è il bistabile SR (Set-Reset)
  - Viene utilizzato come blocco base per realizzare bistabili più complessi







Asincroni SR – Analisi di funzionamento

- Tempo $t=t_0=0$
  - Condizione iniziale: S=0, R=0, Q=0, Q'=1
- Tempo $t=t_1$: evento S=1 (set)
  - La porta 1 ha in ingresso 1,0 e in uscita, al tempo t_2, Q'=0
- Tempo $t=t_2$
  - La porta 2 ha in ingresso 0,0 e in uscita, al tempo t_3, Q=1
- Tempo $t=t_3$
  - La porta 1 ha in ingresso 1,1 e mantiene l'uscita a Q'=0
  - La porta 2 ha in ingresso 0,0 e mantiene l'uscita a Q=1
- Tempo $t=t_4$: evento S=0
  - La porta 1 ha in ingresso 0,1 e quindi mantiene l'uscita Q'=0
  - La porta 2 ha in ingresso 0,0 e quindi mantiene l'uscita Q=1
  - Il circuito è stabile nello stato Q=1, Q'=0

Asincroni SR – Analisi di funzionamento (cont.)

- Tempo $t=t_5$: Evento R=1 (reset)
  - La porta 2 ha in ingresso 1,0 e in uscita, al tempo t_6, Q=0
- Tempo $t=t_6$
  - La porta 1 ha in ingresso 0,0 e in uscita, al tempo t_7, Q'=1
- Tempo $t=t_7$
  - La porta 2 ha in ingresso 1,1 e mantiene l'uscita a Q=0
  - La porta 1 ha in ingresso 0,0 e quindi mantiene l'uscita a Q'=1
- Tempo $t=t_8$: Evento R=0
  - La porta 2 ha in ingresso 0,1 e quindi mantiene l'uscita a Q=0
  - La porta 1 ha in ingresso 0,0 e quindi mantiene l'uscita a Q'=1
  - Il circuito è stabile nello stato Q=0, Q'=1

Asincroni SR - Set e Reset

- I segnali S e R prendono il nome di Set e Reset
  - Un 1 su Set porta Q ad 1, mentre un 1 su reset porta Q a 0
- Riassumendo
  - Un valore 0 sugli ingressi S e R non modifica lo stato
  - Un valore 1 su S quando R=0 porta le uscite allo stato stabile Q=1, Q'=0
  - Un valore 1 su R quando S=0 porta le uscite allo stato stabile Q=0, Q'=1
  - La configurazione S=1 e R=1 è una configurazione non ammissibile
- Osservazione
  - Nelle configurazioni valide le uscite Q e Q' sono complementari per costruzione

Asincroni SR – ingressi non ammissibili

- Applicando contemporaneamente su S e su R un valore 1 il circuito si porta in uno stato instabile con Q=0, Q'=0

- Tale configurazione non è ammissibile, infatti:
  - Nel passaggio degli ingressi da 11 a 00, non è possibile identificare chi tra S o R cambia per primo
  - Il bistabile asincrono ritorna quindi in modo imprevedibile allo stato Q=0, Q'=1 oppure allo stato Q=1, Q'=0
  - Questa condizione è chiamata *race condition* (corsa critica) o transizione non deterministica

Bistabili – Descrizione del comportamento

- Tabella delle transizioni (o mappa di Karnaugh)
  - Ingressi: ingressi primari i^t, stato presente Q^t
  - Uscita: prossimo stato Q^{t+1}
- Tabella delle eccitazioni
  - Ingressi: stato presente Q^t, prossimo stato Q^{t+1}
  - Uscita: configurazione degli ingressi primari che realizza la transizione
- Equazione di funzionamento (espressione logica)
  - Ricavata dalla tabella delle transizioni

Esempio: Bistabili asincroni SR

Tabella delle transizioni

Q \ SR	00	01	11	10
0	0	0	-	1
1	1	0	-	1

⟺

S	R	Q*
0	0	Q
0	1	0
1	0	1
1	1	-

Tabella delle eccitazioni

Q	Q*	S	R
0	0	0	-
0	1	1	0
1	0	0	1
1	1	-	0

Espressione logica

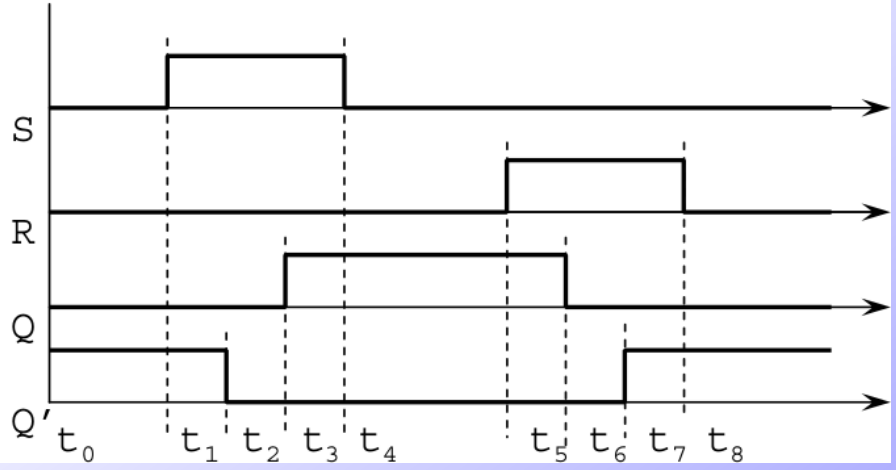
$Q^* = S + R'Q$

Con vincolo $S=R\neq 1$

Q*: stato prossimo

Q : stato presente

Diagramma Temporale

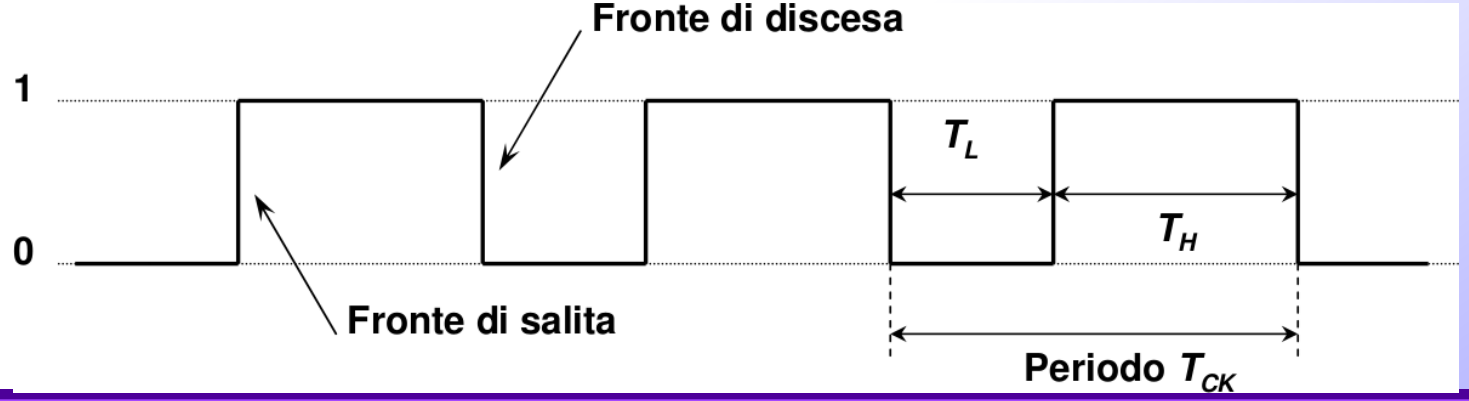


Sincronismo

- Un bistabile asincrono modifica il proprio stato solo in relazione ad eventi sugli ingressi
- Il progetto di circuiti digitali può richiedere che la modifica dello stato avvenga in modo controllato
  - Ad esempio, solamente ad istanti di tempo ben precisi, in modo che eventi transitori non siano significativi
- Questa esigenza impone l'aggiunta di un ingresso di controllo al bistabile
- Il segnale applicato all'ingresso di controllo può essere
  - Aperiodico
  - Periodico (denominato *Clock*) – nella maggior parte dei casi

Segnale di clock

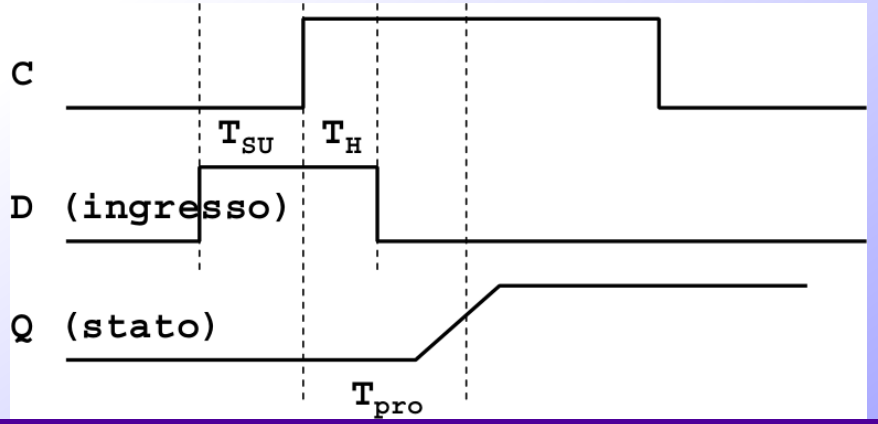
- Il clock è un segnale indipendente caratterizzato da un periodo di clock (o ciclo di clock) T_{CK}
  - Frequenza del clock: $f_{CK} = 1 / T_{CK}$
- Nel periodo T_{CK} il segnale assume il valore logico 1 per un tempo T_H e il valore logico 0 per un tempo T_L
  - Il rapporto T_H / T_{CK} è detto *duty-cycle*
- Il passaggio dal valore 0 al valore 1 è detto *fronte di salita*
- Il passaggio dal valore 1 al valore 0 è detto *fronte di discesa*



Tempi di Hold e Set-Up

- Per essere riconosciuto correttamente, un ingresso primario di un bistabile deve rimanere stabile all'interno di una finestra di tempo nell'intorno di un fronte del clock
- Tempo di Set-Up (T_{su})
  - Intervallo minimo che precede l'evento di clock durante il quale l'ingresso deve essere mantenuto stabile
- Tempo di Hold (T_H)
  - Intervallo minimo che segue l'evento di clock durante il quale l'ingresso deve essere mantenuto stabile

T_{su}: tempo di set-up
T_H: tempo di hold
T_{pro}: tempo di propagazione



Bistabili sincroni: Relazione ingresso-stato

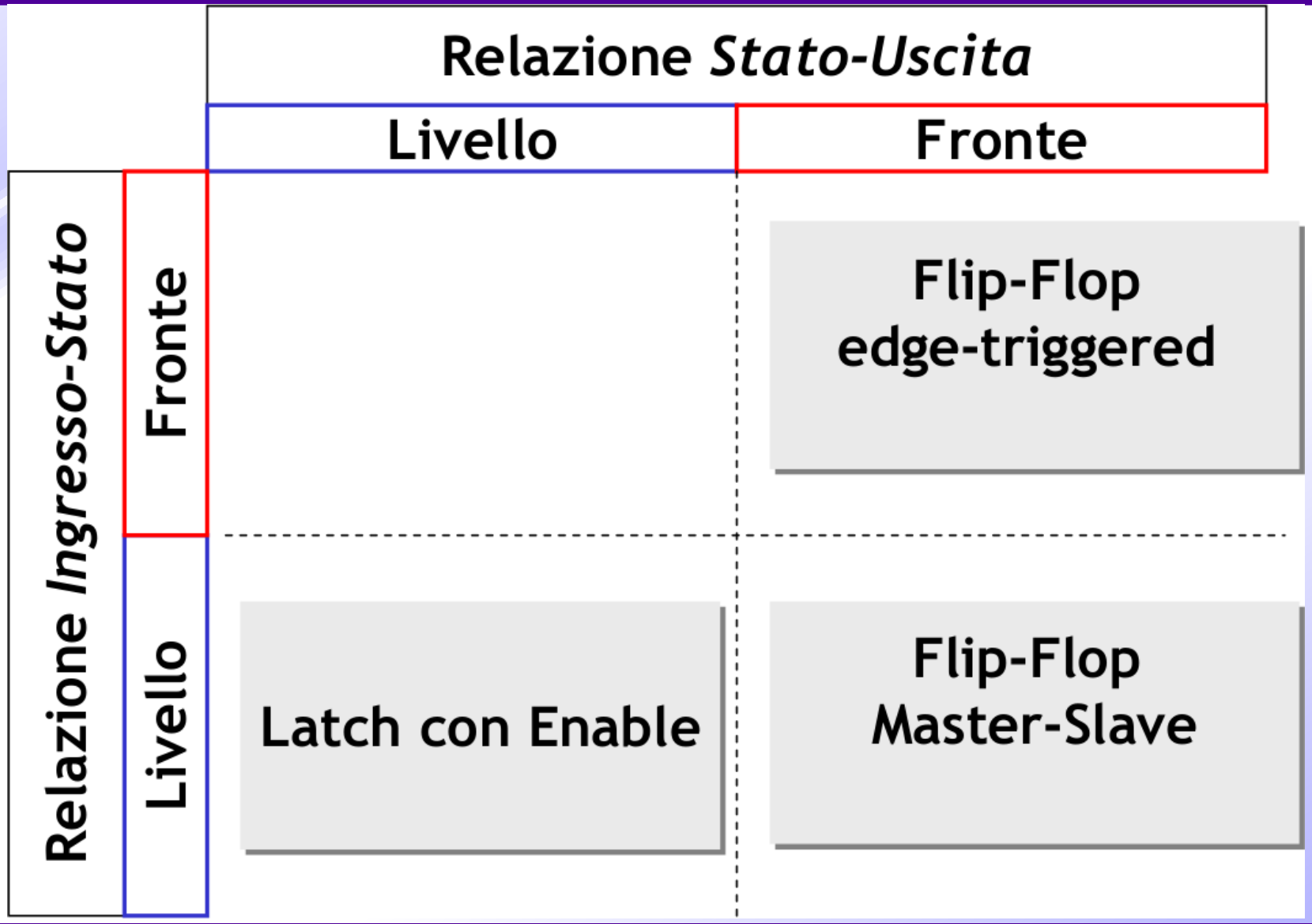
- I fattori che differenziano i bistabili sincroni riguardano
  - La relazione ingresso-stato (quando gli ingressi sono efficaci)
  - La relazione stato-uscita (quando vengono modificate le uscite)

- La relazione ingresso-stato (tipo di temporizzazione) definisce quando gli ingressi modificano lo stato (interno) del bistabile
  - Basato sul livello del segnale di controllo (*master-slave*)
    - Durante tutto l'intervallo di tempo in cui il segnale di controllo è attivo, qualsiasi variazione sui segnali di ingresso influenza il valore dello stato interno del bistabile
  - Basato sul fronte del segnale di controllo (*edge-triggered*)
    - Il valore dello stato interno del bistabile viene aggiornato solamente in corrispondenza di un fronte del segnale di controllo

Bistabili sincroni: Relazione stato-uscita

- La relazione stato-uscita definisce quando lo stato aggiorna le uscite
  - Basato sul livello del segnale di controllo (*latch*)
    - Durante tutto l'intervallo in cui il segnale di controllo è attivo un cambiamento dei segnali di ingresso modifica oltre allo stato interno anche le uscite
    - Le uscite cambiano quando cambiano gli ingressi
    - Il segnale di controllo è solitamente chiamato *enable*
  - Basato sul fronte del segnale di controllo (*flip-flop*)
    - Le uscite vengono aggiornate su di un fronte del segnale di sincronismo
    - Le uscite cambiano in corrispondenza di un evento del clock

Bistabili sincroni – Tipologie



Latch: SR

- Il latch SR è ottenuto aggiungendo al bistabile asincrono SR un circuito di controllo
- Sul livelo alto di C una variazione sugli ingressi modifica lo stato interno e lo stato interno modifica le uscite Q e Q'
  - C=1 modalità *trasparente*
  - C=0 modalità *opaca*

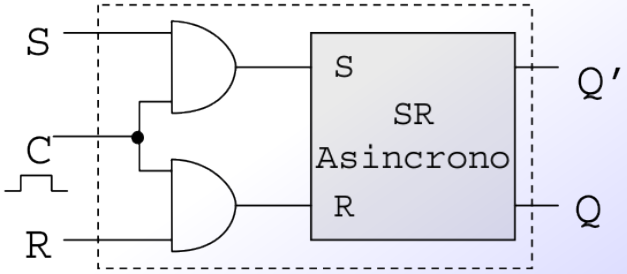


Tabella delle transizioni

C	S	R	Q*	
0	-	-	Q	
1	0	0	Q	hold
1	0	1	0	reset
1	1	0	1	set
1	1	1	-	not allowed

Tabella delle eccitazioni

Q	Q*	C	S	R
0	0	0	-	-
1	1	0	-	-
0	0	1	0	-
0	1	1	1	0
1	0	1	0	1
1	1	1	-	0

Espressione logica

$$Q^* = C'Q + C(S + R'Q)$$

Latch: D

- Il latch D è ottenuto a partire da un latch SR imponendo che S=R'
  - C=1 modalità *trasparente*
    - Q segue l'ingresso
  - C=0 modalità *opaca*
    - Q mantiene l'ultimo ingresso letto
- "D": Delay o Data

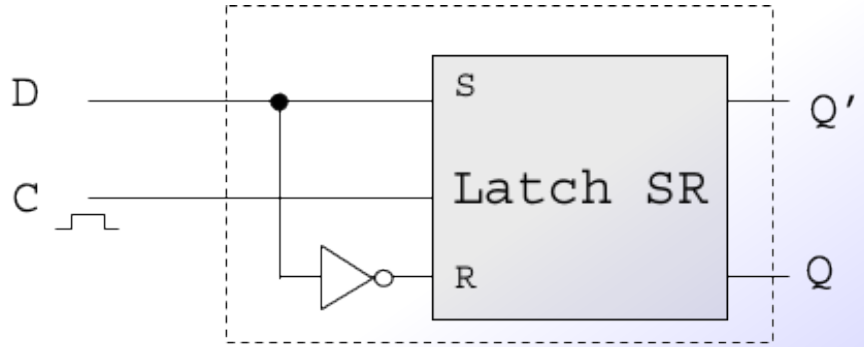


Tabella delle transizioni

C	D	Q*
0	-	Q
1	0	0
1	1	1

Tabella delle eccitazioni

Q	Q*	C	D
0	0	0	-
1	1	0	-
0	0	1	0
0	1	1	1
1	0	1	0
1	1	1	1

Espressione logica

$Q^* = C'Q + CD$

Latch: JK - qui

- Il latch JK è simile ad un SR, ma con la configurazione J=K=1 (con C=1) il valore dello stato viene invertito
  - Per J=K=1 si ottiene Q*=Q'

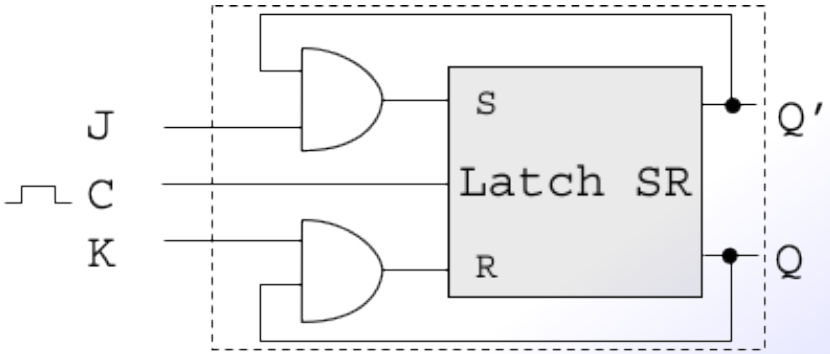


Tabella delle transizioni

C	J	K	Q*	
0	-	-	Q	
1	0	0	Q	hold
1	0	1	0	reset
1	1	0	1	set
1	1	1	Q'	toggle

Tabella delle eccitazioni

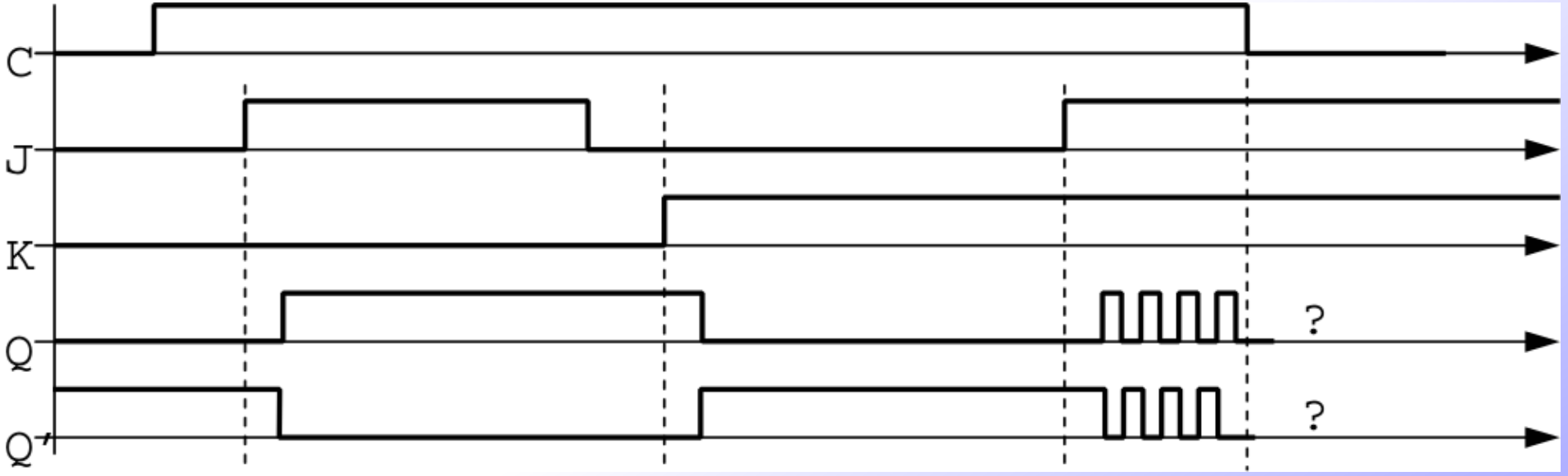
Q	Q*	C	J	K
0	0	0	-	-
1	1	0	-	-
0	0	1	0	-
0	1	1	1	-
1	0	1	-	1
1	1	1	-	0

Espressione logica

$$Q^* = C'Q + C(K'Q + JQ')$$

Flip-Flop -1

- I latch, spesso, non consentono di garantire un comportamento affidabile nella realizzazione di una data funzionalità
  - Esempio 1: analisi del comportamento del latch JK



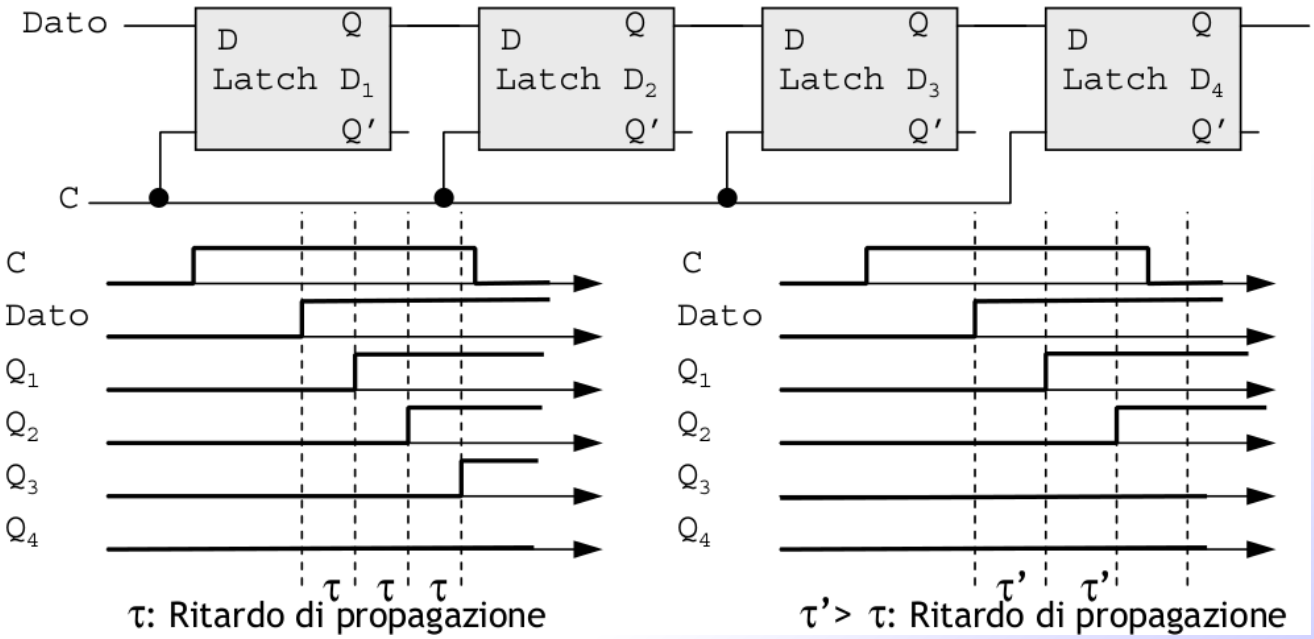
Flip-Flop-2

- Per J=K=1 il bistabile ha un comportamento instabile
  - Le uscite Q e Q' hanno un comportamento oscillatorio ed il valore risultante quando J, K o C cambiano non è noto a priori (race condition)

- Per un funzionamento corretto con gli ingressi J=K=1
  - Un solo cambiamento di stato per ciclo di clock per evitare l'effetto di propagazione indesiderato tra uscite ed ingresso

Flip-Flop

- Esempio 2: *shift register*



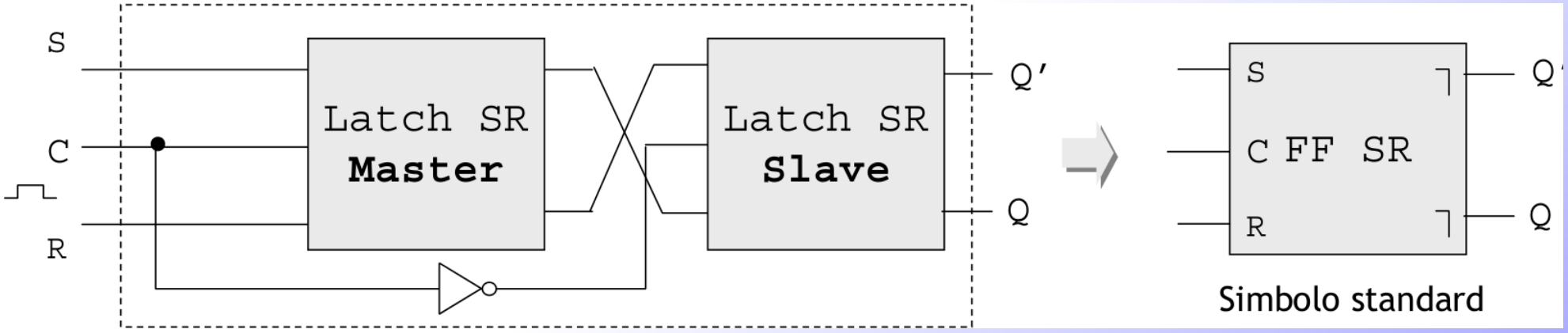
- Due problemi
  - Non produce una singola traslazione di un bit (non rispetta le specifiche)
  - Il risultato dipende
    - Sia dal ritardo di propagazione dei latch
    - Sia dalla durata del valore alto su C

Flip-Flop

- Per evitare l'effetto di propagazione indesiderata, i bistabili sincroni vengono modificati in modo che lo stato possa modificare le uscite solo in corrispondenza di un evento (fronte) del segnale di controllo
- Flip-Flop:
  - Relazione stato-uscita (aggiornamento dell'uscita)
    - Sul fronte
  - Relazione ingresso-stato (aggiornamento dello stato)
    - A livello (Flip-Flop master-slave)
    - A fronte (Flip-Flop edge triggered)

Flip-Flop: SR Master-Slave

- I flip-flop master-slave vengono realizzati utilizzando due latch in cascata
  - Il primo latch sincrono è chiamato latch principale (*master*)
  - Il secondo latch sincrono è chiamato latch ausiliario (*slave*)
  - I due latch lavorano in contrapposizione di fase (il percorso di propagazione ingresso-uscita non è continuo)
- Flip-flop master-slave SR (fronte di discesa)



Flip-Flop: Master-Slave

- Funzionamento
  - Segnale di sincronismo sul livello alto (fronte di salita)
    - Il latch *master* è trasparente e modifica il valore dello stato interno al Flip-Flop in relazione ai valori assunti dai segnali di ingresso
    - Il latch *slave* è opaco e non consente che le uscite vengano modificate
  - Segnale di sincronismo passa al livello basso (fronte di discesa)
    - Il latch *master* passa da trasparente a opaco, mantenendo stabile il valore dello stato interno
    - Il latch *slave* passa da opaco a trasparente e lo stato interno aggiorna le uscite
- Il comportamento complessivo vede dunque due fasi:
  1. Durante il livello attivo alto del segnale di sincronizzazione il valore degli ingressi (ad esempio, S e R) determinano il valore dello stato interno del latch *master*
  2. Sul fronte di discesa del segnale di clock viene aggiornato il valore delle uscite del bistabile che rimane fisso fino al successivo fronte di discesa

Flip-Flop: D Master-Slave

- Flip-flop master-slave D (fronte di discesa)

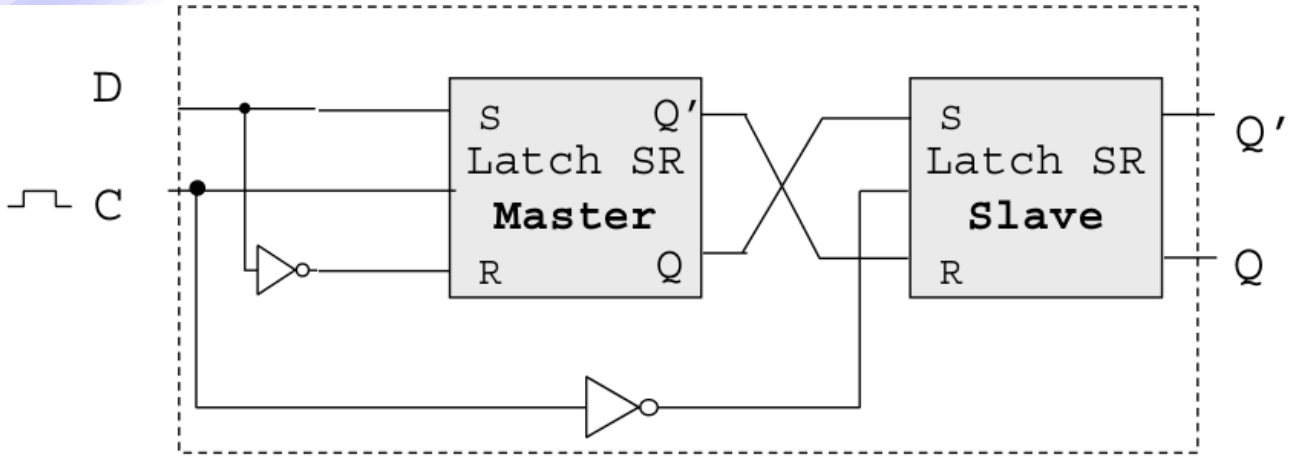


Tabella delle transizioni

D	Q*
0	0
1	1

C=1

Tabella delle eccitazioni

Q	Q*	D
0	0	0
0	1	1
1	0	0
1	1	1

Espressione logica

$Q^* = D$

Flip-Flop: JK Master-Slave

- Flip-flop master-slave JK (fronte di discesa)

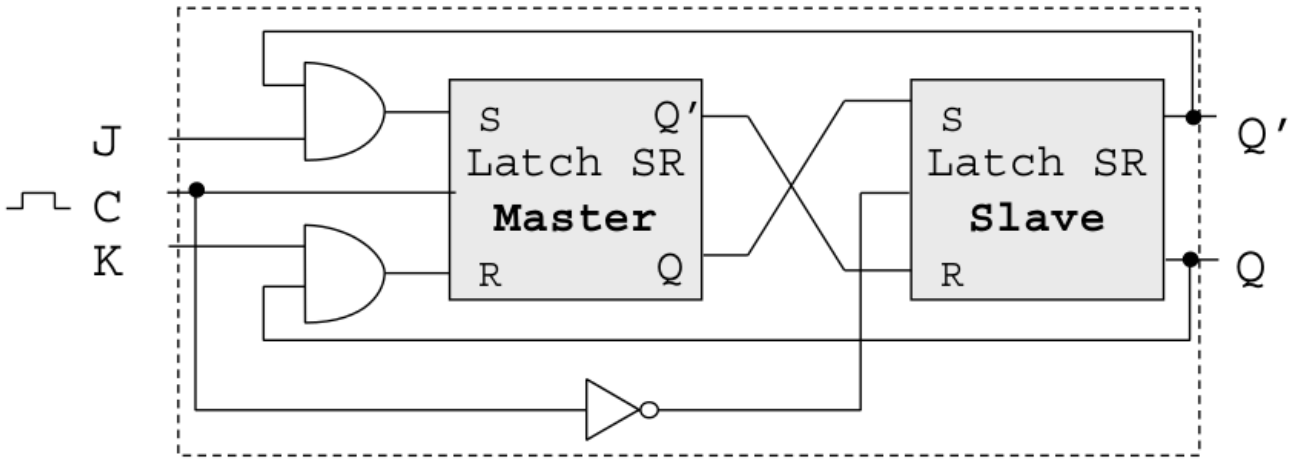


Tabella delle transizioni

J	K	Q*	
0	0	Q	hold
0	1	0	reset
1	0	1	set
1	1	Q'	toggle

C=1

Tabella delle eccitazioni

Q	Q*	J	K
0	0	0	-
0	1	1	-
1	0	-	1
1	1	-	0

Espressione logica

$Q^* = JQ' + K'Q$

Flip-Flop: T(oggle) Master-Slave

- Flip-flop master-slave T (fronte di discesa)

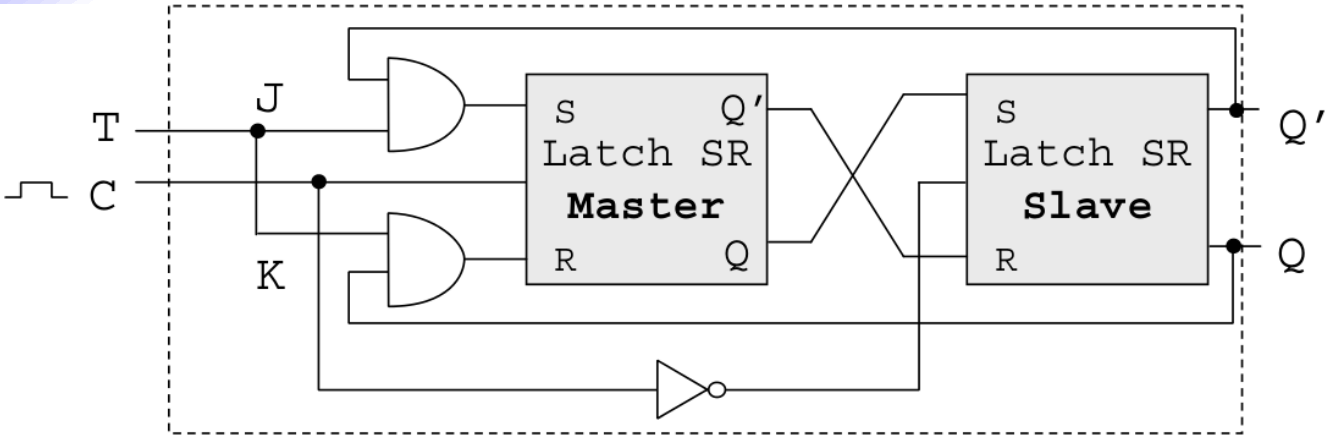


Tabella delle transizioni

T	Q*
0	Q
1	Q'

Tabella delle eccitazioni

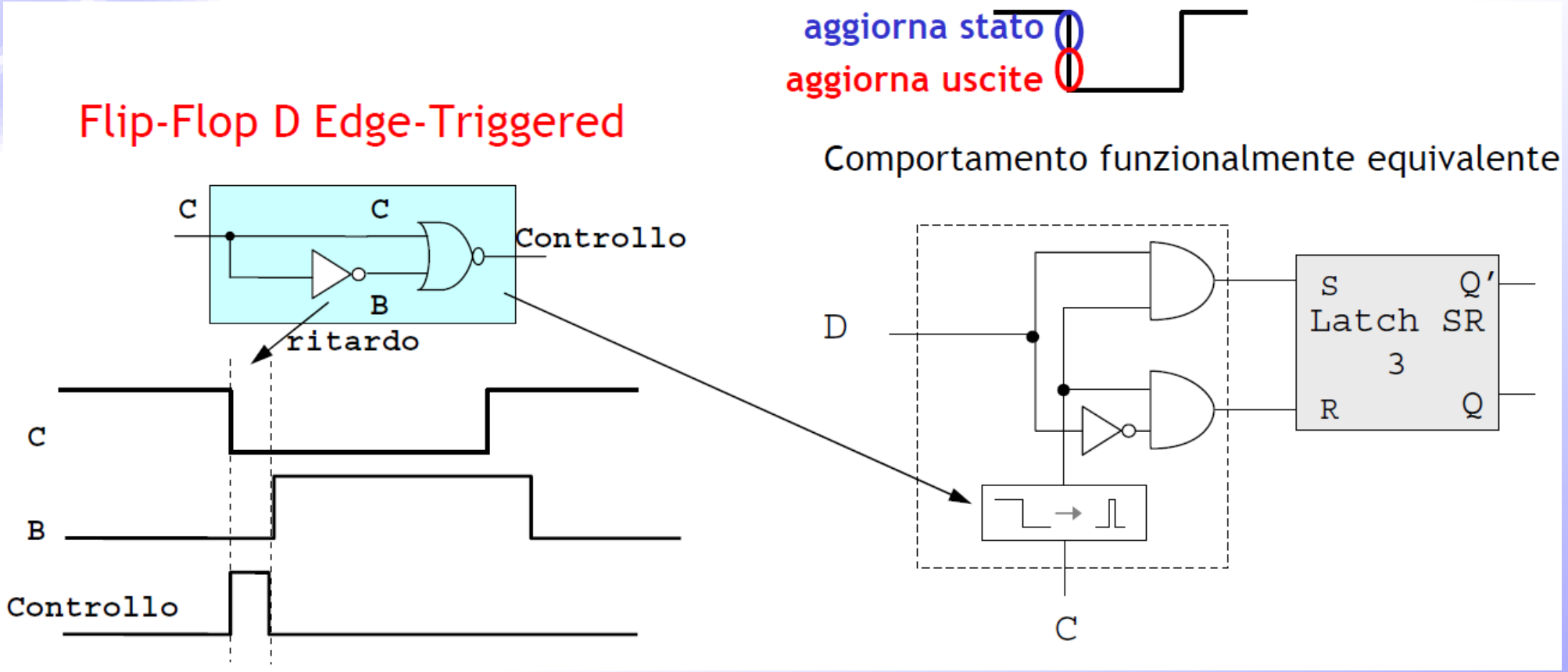
Q	Q*	T
0	0	0
0	1	1
1	0	1
1	1	0

Espressione logica

$Q^* = TQ' + T'Q$

Flip-Flop: Edge-Triggered

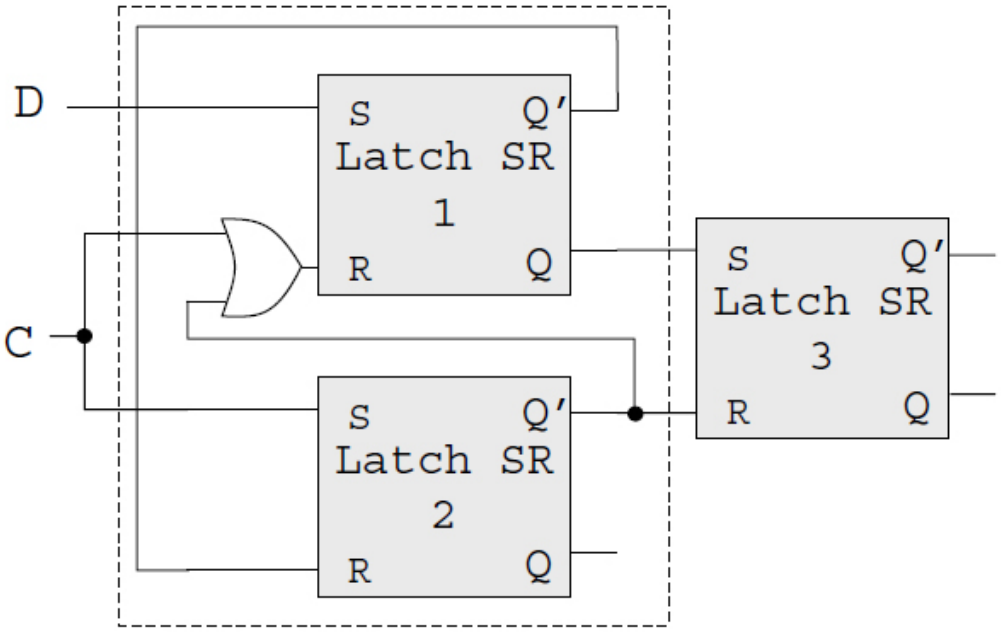
- I flip-flop Edge-Triggered vengono realizzati producendo, o fisicamente o funzionalmente, la derivata del segnale di clock
  - Genera un impulso (fisico o funzionale) in corrispondenza di un fronte



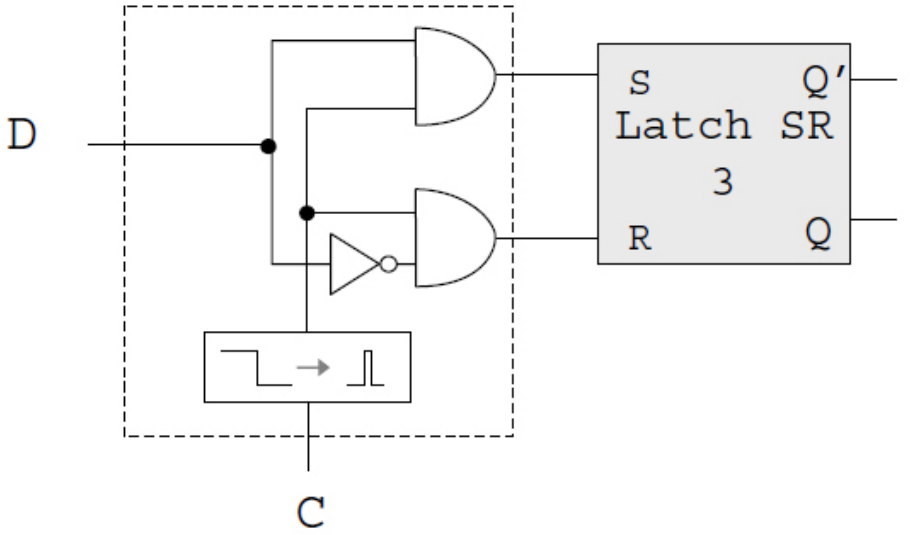
Flip-Flop: Edge-Triggered

- Flip-Flop D Edge-Triggered costituito da 3 latch con comportamento globale equivalente a quello prima visto

Flip-Flop D Edge-Triggered

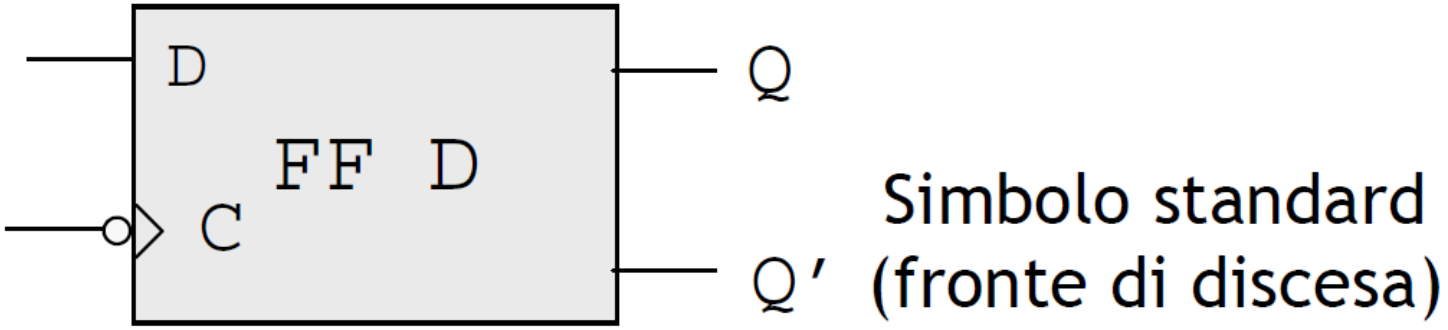


Comportamento funzionalmente equivalente



Flip-Flop: Edge-Triggered

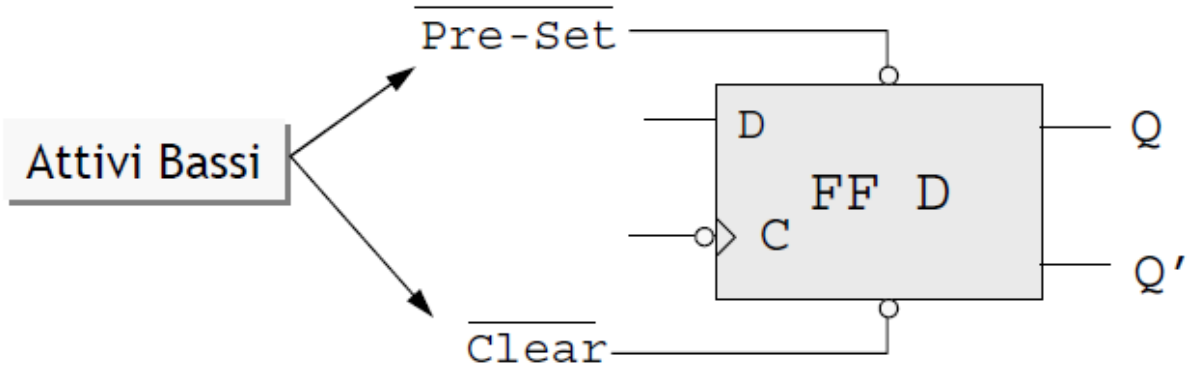
- Funzionamento:
  - Per C=1 gli ingressi di Latch SR 3 sono S=0 e R=0
  - Durante C=1➔0, il valore su D attiva il latch SR 1 e, successivamente, il latch SR 2 viene attivato.
  - Se D=1, il segnale Q del latch SR 1 viene portato a 1; se D=0 il segnale Q del latch SR 1 resta a 0
- Nota:
  - per C=1 il Latch SR 1 può trovarsi nella condizione instabile 11 (a cui consegue Q=Q'=0); tale situazione viene risolta nel passaggio di C da 1 a 0 producendo uno stato stabile e deterministico che dipende solo dal valore assunto da D durante la transizione.
  - I tempi di *Hold* e *Set-Up* devono essere rispettati.



Simbolo standard (fronte di discesa)

Latch & Flip-Flop: Pre-set e Clear

- Spesso, nei Flip Flop e nei Latch sono presenti degli ingressi diretti che sono utilizzati per scavalcare gli ingressi dati
  - Gli ingressi diretti sono asincroni
  - Sono utili per:
    - Stabilire lo stato iniziale del Flip-Flop o del Latch;
    - Mantenere il Flip-Flop o il Latch in uno stato particolare indipendentemente dai dati presenti ai terminali di ingresso.



Esempio di simbolo standard con ingressi diretti di `Pre-Set` e `Clear`. (FF D su fronte di discesa)

Latch & Flip-Flop

Tabella riassuntiva conclusiva:

- Nota: i bistabili Latch e M/S considerati sono attivi a livello alto. Analoghe considerazioni possono essere effettuale per elementi attivi a livello basso.

Tipo	Quando campiona gli ingressi	Quando le uscite sono valide
Latch senza clock	Sempre	Ritardo di propagazione dal cambiamento degli ingressi
Latch sensibile a livello	Clock alto (T_{SU} e T_H attorno al fronte di salita)	Ritardo di propagazione dal cambiamento degli ingressi
Flip-Flop master/slave	Clock alto (T_{SU} e T_H attorno al fronte di discesa)	Ritardo di propagazione dal fronte di discesa del clock
Flip-Flop attivo sul fronte di salita	Transizione 0➔1 del Clock (T_{SU} e T_H attorno al fronte di salita)	Ritardo di propagazione dal fronte di salita del clock
Flip-Flop attivo sul fronte di discesa	Transizione 1➔0 del Clock (T_{SU} e T_H attorno al fronte di discesa)	Ritardo di propagazione dal fronte di discesa del clock

Tabelle delle Transizioni e delle Eccitazioni

Tabelle delle Transizioni:

S	R	Q*
0	0	Q
0	1	0
1	0	1
1	1	-

J	K	Q*
0	0	Q
0	1	0
1	0	1
1	1	Q’

D	Q*
0	0
1	1

T	Q*
0	Q
1	Q’

Tabelle delle Eccitazioni:

Q	Q*	S	R
0	0	0	-
0	1	1	0
1	0	0	1
1	1	-	0

Q	Q*	J	K
0	0	0	-
0	1	1	-
1	0	-	1
1	1	-	0

Q	Q*	D
0	0	0
0	1	1
1	0	0
1	1	1

Q	Q*	T
0	0	0
0	1	1
1	0	1
1	1	0

I Registri

Circuiti sequenziali speciali: Introduzione

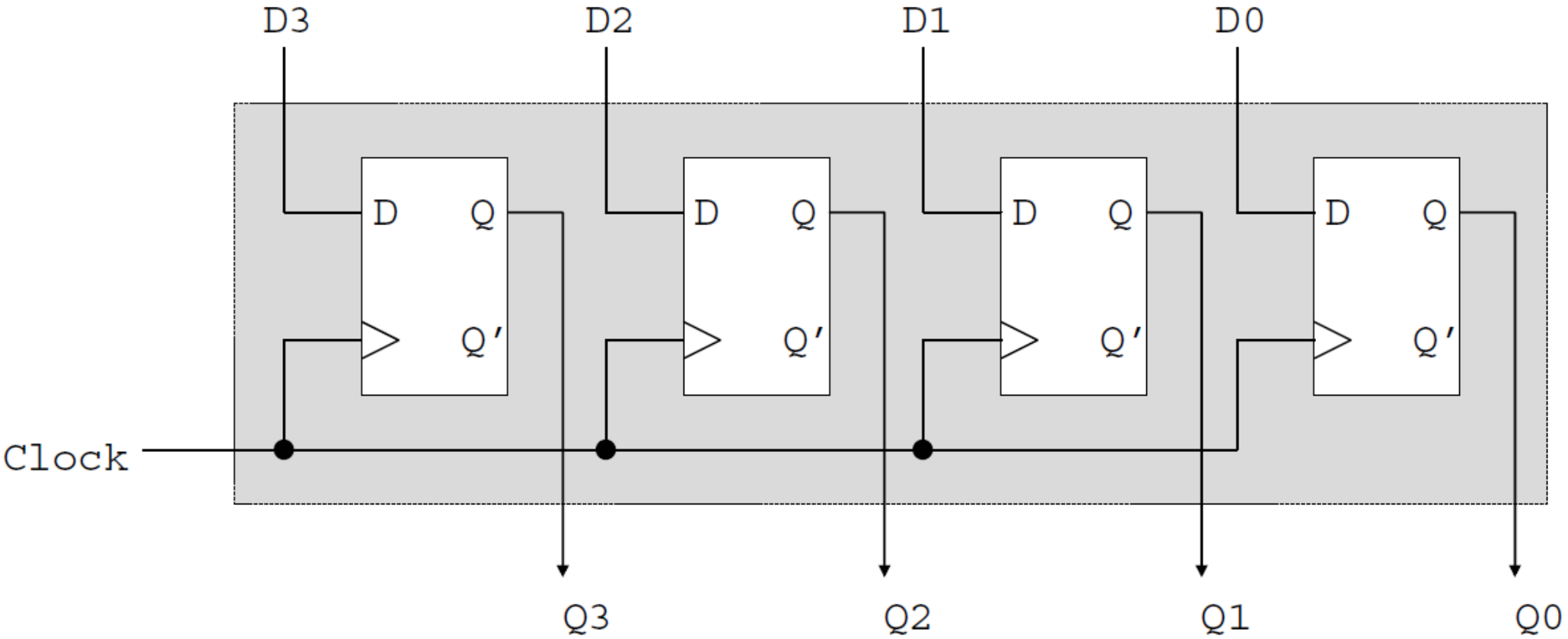
- Esiste una classe di circuiti sequenziali la cui progettazione potrebbe seguire il processo "classico" di sintesi ma che è più conveniente analizzare in altro modo.
- A questa classe appartengono:
  - Registri
    - Memorizzano una definita quantità di informazione
    - Possono operare sul contenuto una o più semplici trasformazioni.
      - **Shift destro/sinistro**
      - **Caricamento parallelo/seriale**
  - Contatori
    - Attraversano ripetutamente un numero definito di stati
      - **Contatori sincroni**
      - **Contatori asincorni**

Registri

- Un *registro* è un elemento di memoria in grado di conservare un insieme di bit, denominato parola, su cui possono eventualmente operare una o più semplici trasformazioni.
  - Benché si possa utilizzare un qualunque tipo di bistabile, per realizzare i registri si preferisce utilizzare FF D (*master-slave o edge-triggered*).
- I registri si distinguono sulla base dei seguenti aspetti:
  - Modalità di caricamento dati
    - Parallelo
    - Seriale
  - Modalità di lettura dati
    - Parallelo
    - Seriale
  - Operazioni di scorrimento sui dati:
    - a destra e/o a sinistra (aritmetico o non aritmetico) e circolare.

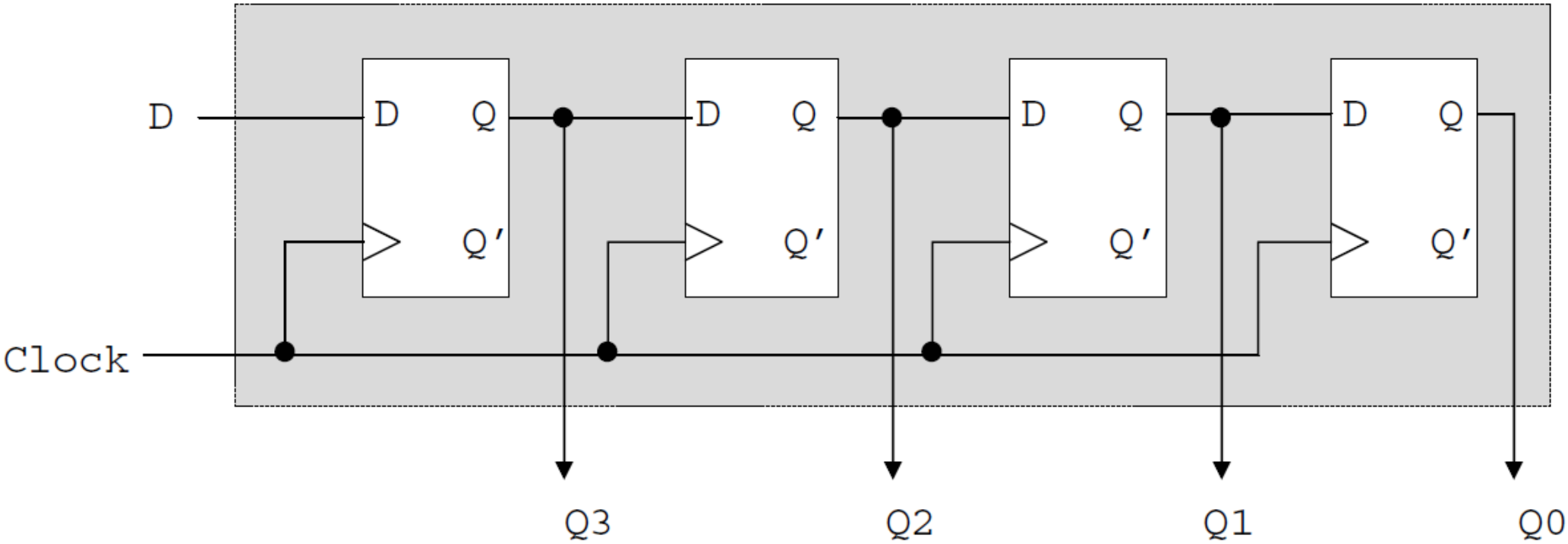
Registro parallelo-parallelo

- Esempio di registro a 4 bit



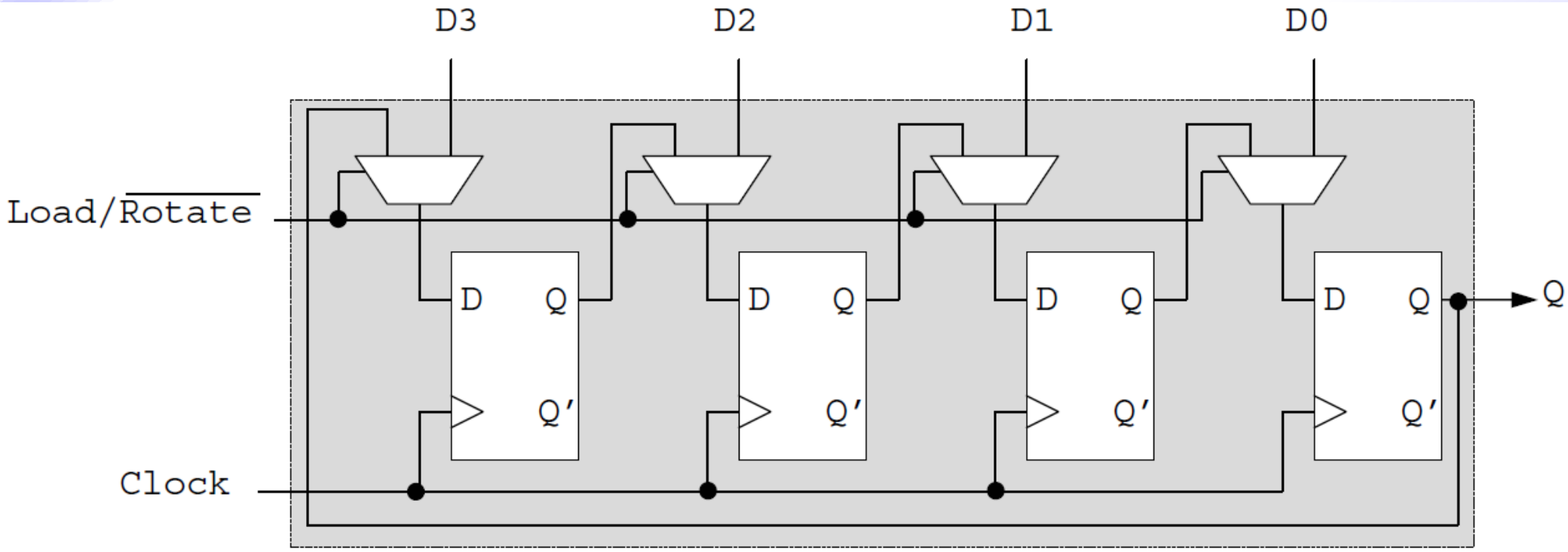
Registro serie-parallelo

- Esempio di registro a 4 bit



Registro circolare a 4 bit

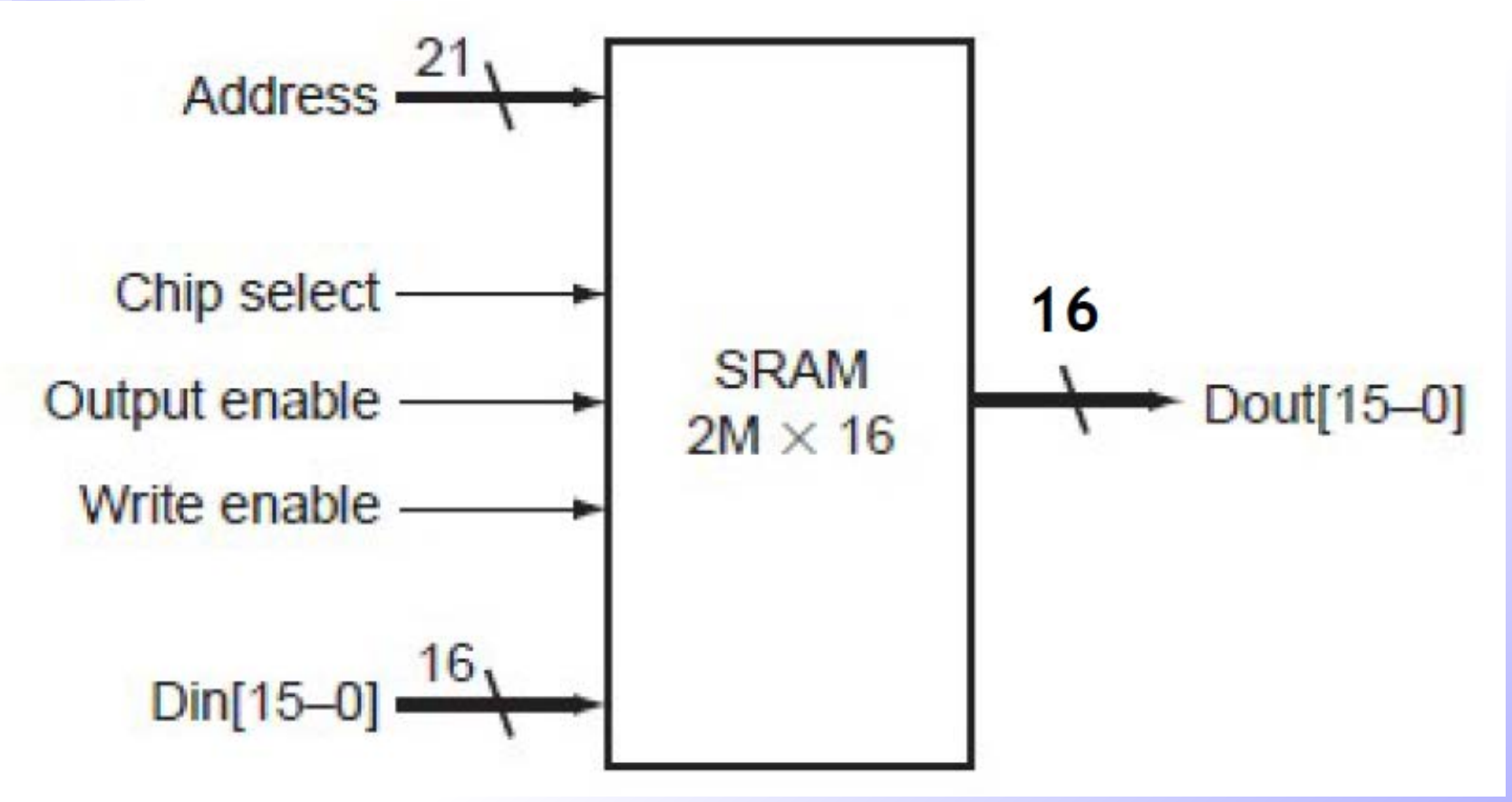
- Esempio a 4 bit con rotazione a destra
  - In fase di traslazione, trasferisce il bit meno significativo al posto di quello più significativo, spostando i rimanenti di una posizione a destra.



Memorie dei calcolatori (cenni)

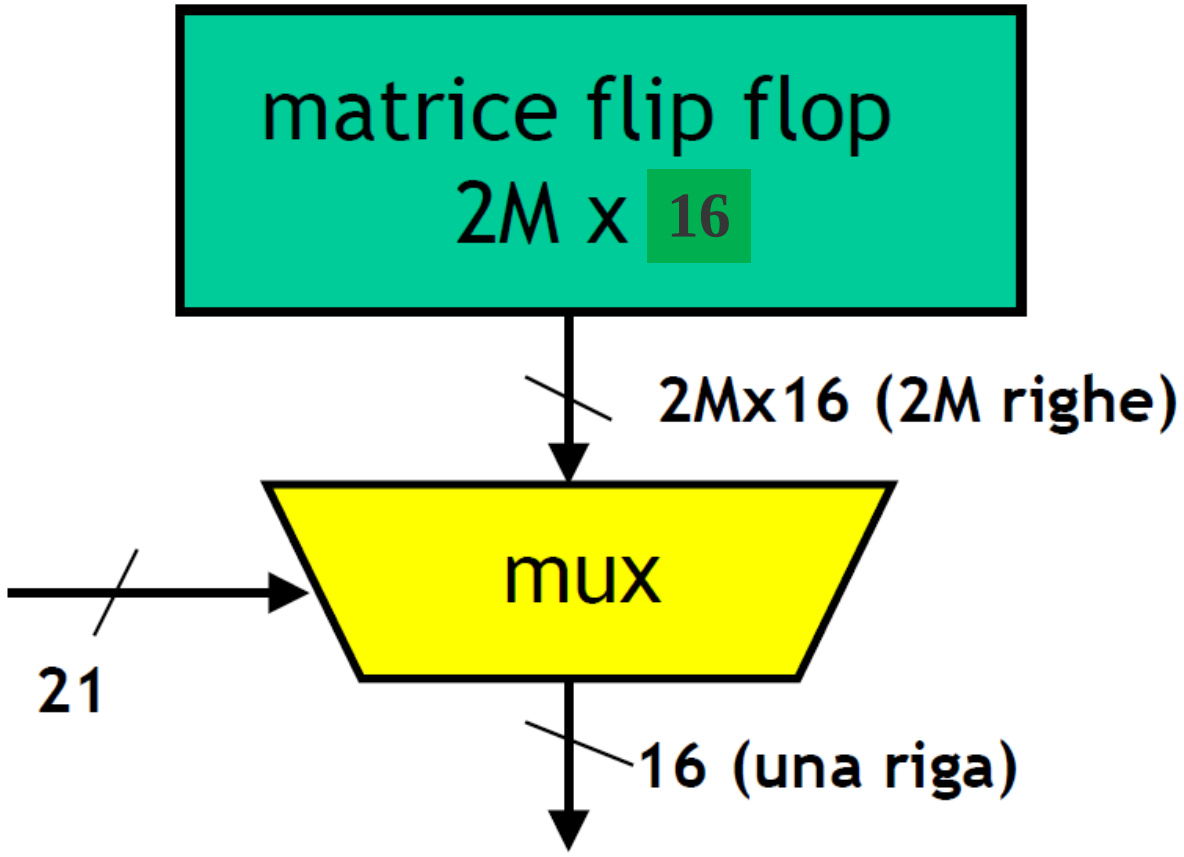
Static RAM (SRAM)

- SRAM = Static Random Access Memory
- Un chip SRAM 2M x 16 (NB: $1M=2^{20}$)



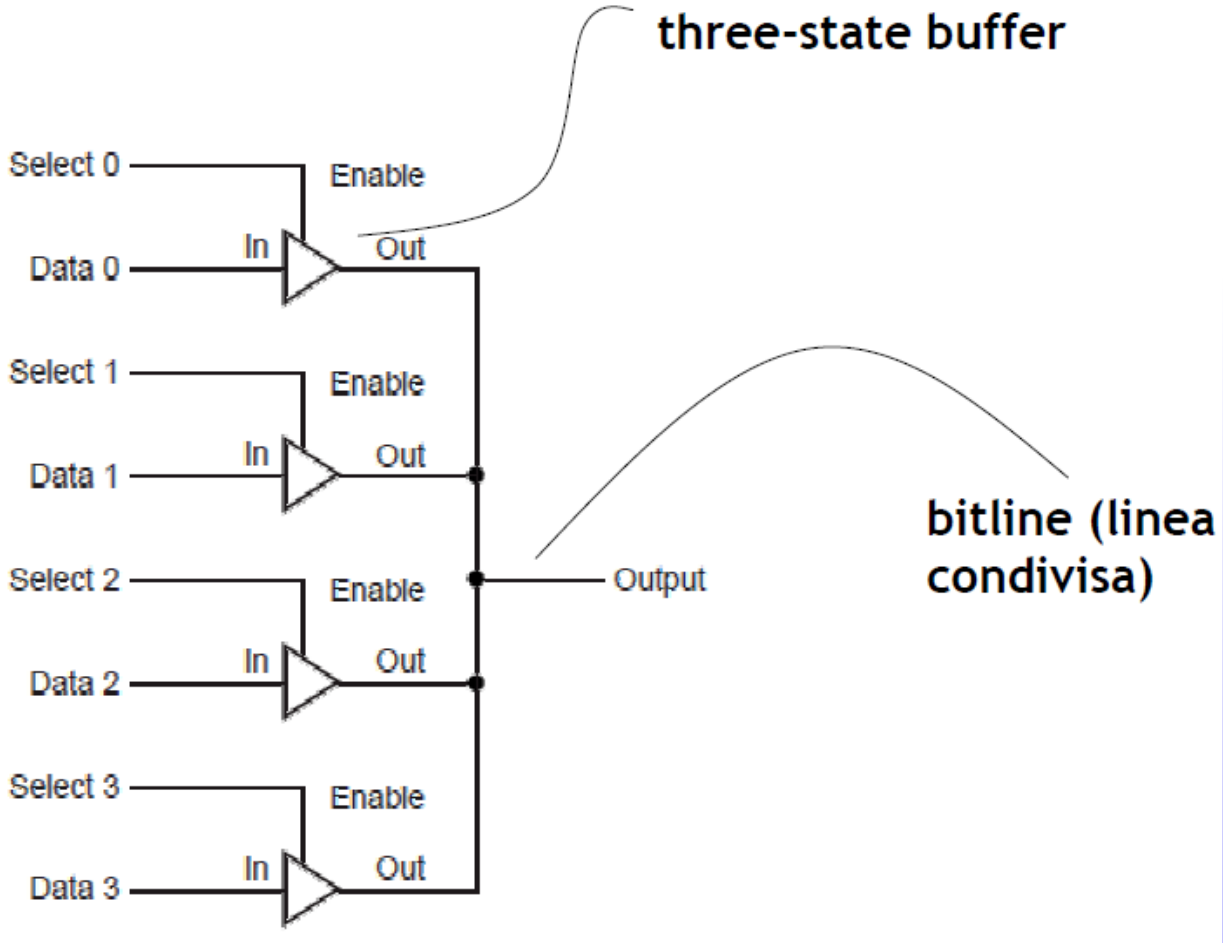
Logica di uscita: soluzione naive

- Usa un Mux con 2Mx16 ingressi, assolutamente *non pratico* dal punto di vista tecnologico:

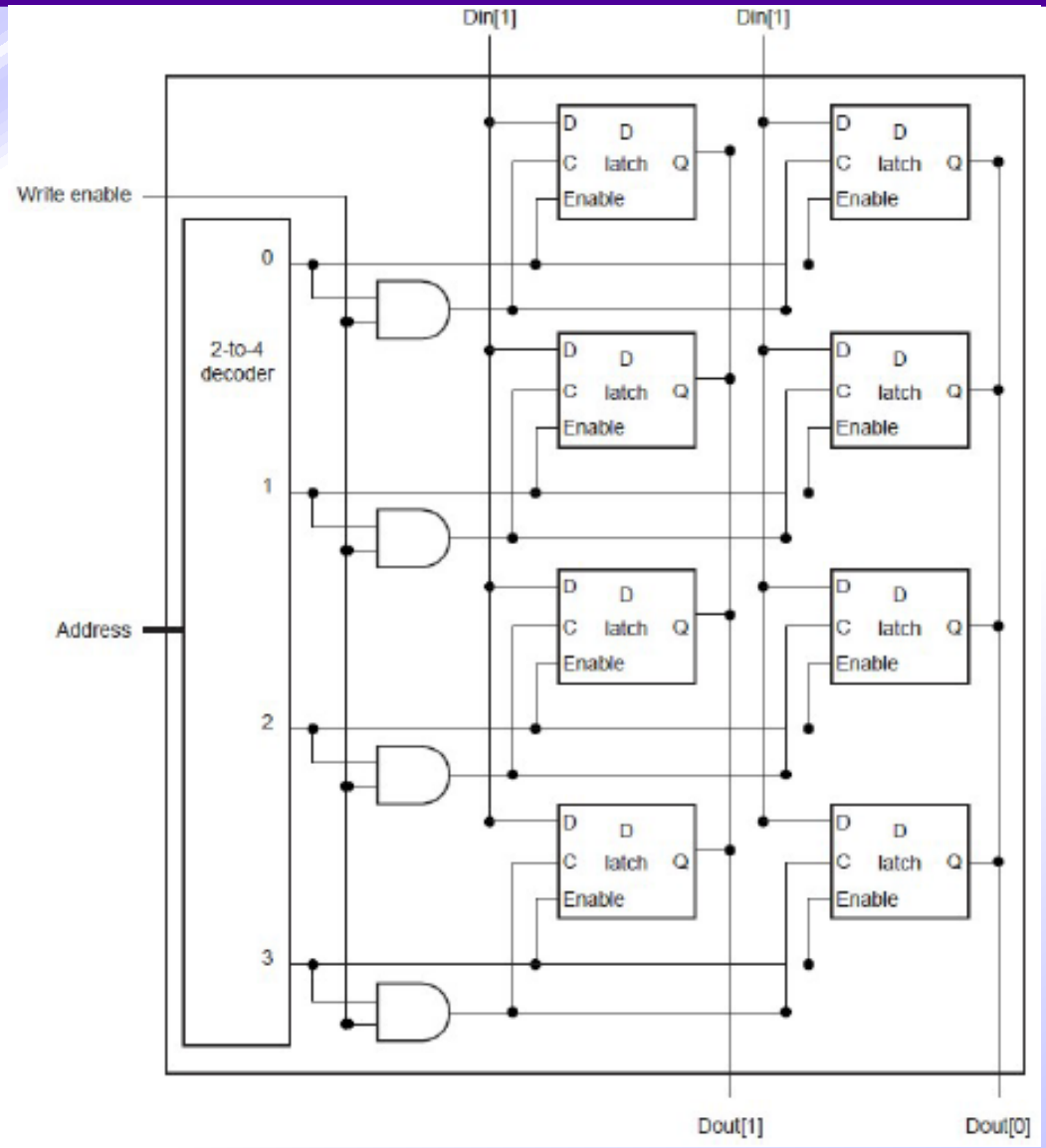


Logica di uscita: linea condivisa

- Esempio: un mux 4-a-1 realizzato con 4 *three-state buffers* e *1 bitline*:



Esempio: SRAM 4 x 2



SRAM: altri aspetti tecnologici

- Tempi accesso: pochi ns (1 ns = 10^{-9} s)
- Capienza tipica: ~ 1 Gbit
- Costo al bit relativamente elevato
- Uso tipico: memoria *cache dei calcolatori*
- Varianti
  - SSRAM: *Synchronous SRAM*. Forniscono accesso veloce ad una serie di bit all'interno di una stessa riga, che vengono trasferiti sequenzialmente alla CPU sotto il controllo di un segnale di clock.

Dynamic RAM (DRAM)

- 1 bit = carica statica all'interno di un *condensatore*
- Necessità di *refresh*: lettura e riscrittura dello stesso valore
- Frequenza refresh: ~ 1 refresh / 1M cicli di clock
- Per fortuna, si può rinfrescare *una riga alla volta* (*v.* decodifica in 2 passi)
- Risultato: 98-99% cicli: normali letture/scritture 2-1 % cicli: refresh

Decodifica in 2 passi

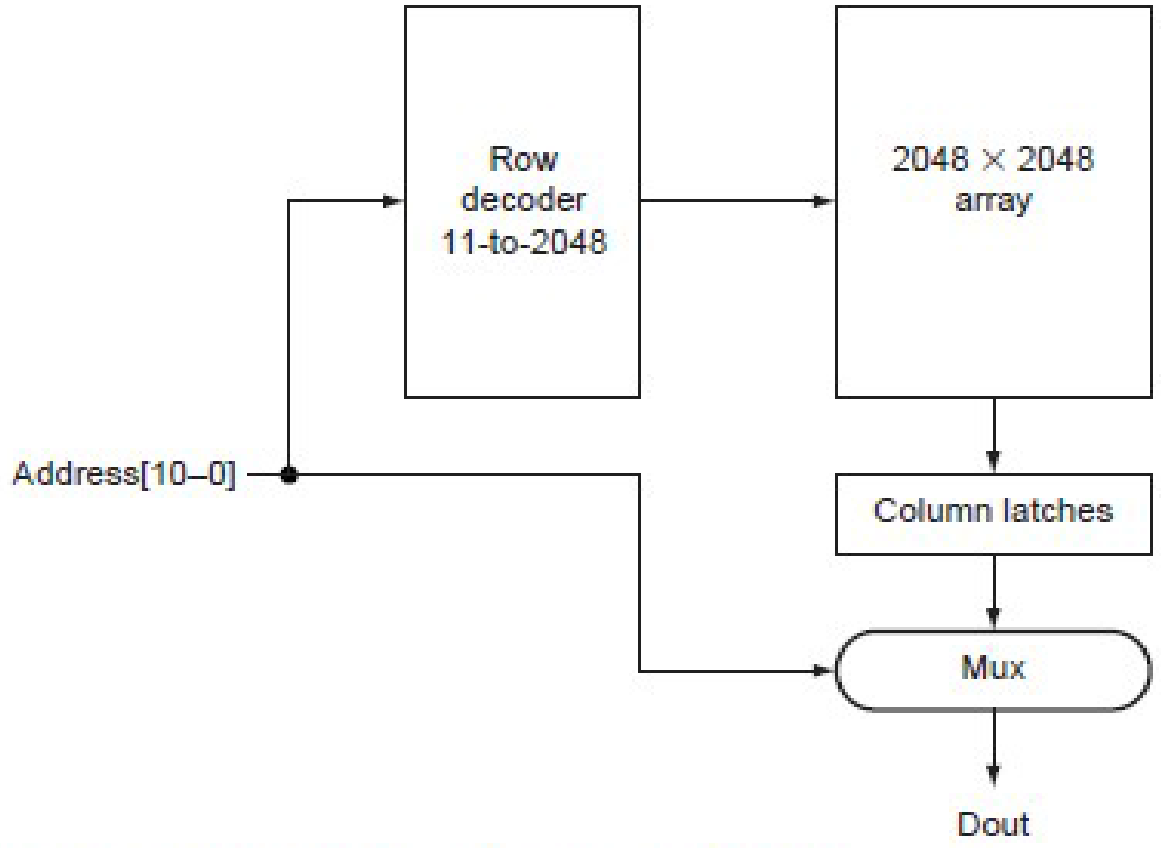


FIGURE B.9.6 A 4M ¥ 1 DRAM is built with a 2048 ¥ 2048 array. The row access uses 11 bits to select a row, which is then latched in 2048 1-bit latches. A multiplexor chooses the output bit from these 2048 latches. The RAS and CAS signals control whether the address lines are sent to the row decoder or column multiplexor.

DRAM: altri aspetti tecnologici

- Tempi accesso: ~ decine ns (10-100 volte più lente di SRAM)
- Costo al bit molto contenuto rispetto a SRAM
- Capienza tipica: ~ diversi Gbit
- Uso tipico: memoria principale dei calcolatori
- Varianti
  - SDRAM
  - DDRAMS (Double DRAM: lettura possibile sia al rising che al falling edge)