DEI
DIPARTIMENTO DI
INGEGNERIA DELL'INFORMAZIONE

UNIVERSITÀ
DEGLI STUDI
DI PADOVA

Sistemi Digitali

Registri a scorrimento e contatori

Marta Bagatin, marta.bagatin@unipd.it

Corso di Laurea in Ingegneria dell'Informazione

Anno accademico 2022-2023

Scopo della lezione

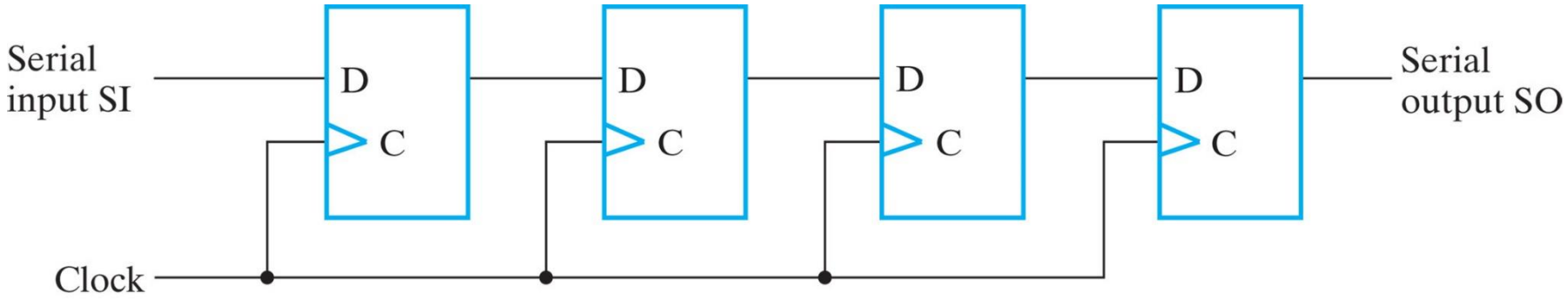
- Studiare il funzionamento di alcuni registri
  - Registri a scorrimento (shift register)
  - Contatori (counter)
    - Contatore ripple
    - Contatore sincrono

Registri a scorrimento
(Shift Register)

Registro a scorrimento (Shift Register)

- E’ un registro che fa **scorrere lateralmente** (verso destra, verso sinistra o in entrambe le direzioni) i **bit** in esso immagazzinati
- Consiste in una **catena di flip-flop**, collegati uno dopo l’altro, con l’uscita di ciascuno collegata all’ingresso del successivo, e controllati tutti dallo stesso **segnale di clock, i cui fronti attivi abilitano lo scorrimento**
- Il registro a scorrimento più semplice è quello costituito da soli flip-flop

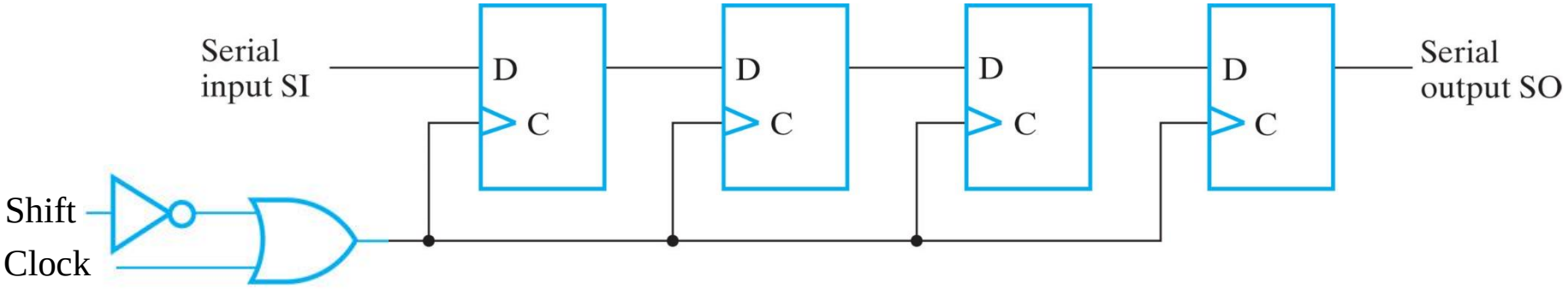
Shift register a 4 bit



- **Input seriale**: ingresso del flip-flop più a sinistra
- **Output seriale**: uscita del flip-flop più a destra
- Il **clock** è connesso direttamente ai FF: lo scorrimento avviene ad ogni fronte di salita del clock (FF di tipo Positive Edge Triggered)

Simbolo

Shift register a 4 bit con shift enable



- Il clock è connesso ai FF tramite un **circuito di abilitazione con un segnale di shift** (analogo al segnale di Load visto in precedenza per abilitare il caricamento di dati in un registro)
- In questo modo si può abilitare lo **scorrimento solo in corrispondenza a determinati fronti del clock**
- Con questa soluzione ci possono essere problematiche di **clock skew,** per questo è preferibile adottare un controllo sull'ingresso del flip-flop piuttosto che sul clock

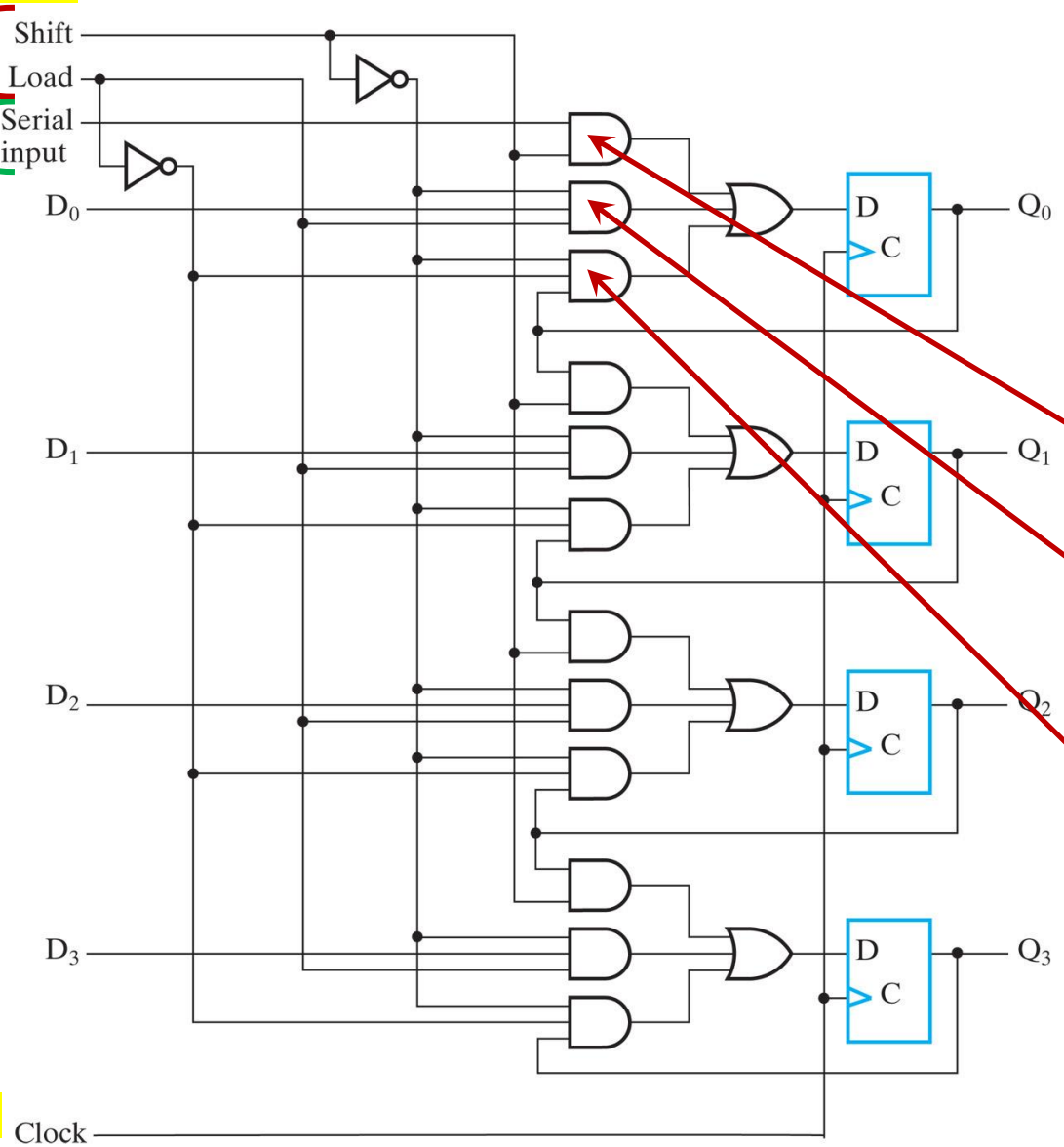
Shift register con caricamento parallelo

- Se abbiamo **accesso all'ingresso di ciascun flip-flop** nello shift register (non solo a quello del primo FF nella catena), possiamo **immettere in parallelo i dati nel registro**
- Analogamente, avendo accesso all'uscita di ogni FF nella catena, possiamo avere in parallelo all'uscita dello shift register i bit immessi in modo seriale
- Uno shift register con **accesso agli ingressi e uscite di tutti i flip-flop** può essere usato per **convertire dati immessi in parallelo in dati uscenti in serie e viceversa**

Shift register con caricamento parallelo

2 segnali di controllo (scorrimento, caricamento)

1 segnale di ingresso



Clock collegato direttamente ai FF

Ciascuno stadio consiste in

- 3 porte AND
- 1 porta OR
- 1 D-FF

Abilita lo scorrimento

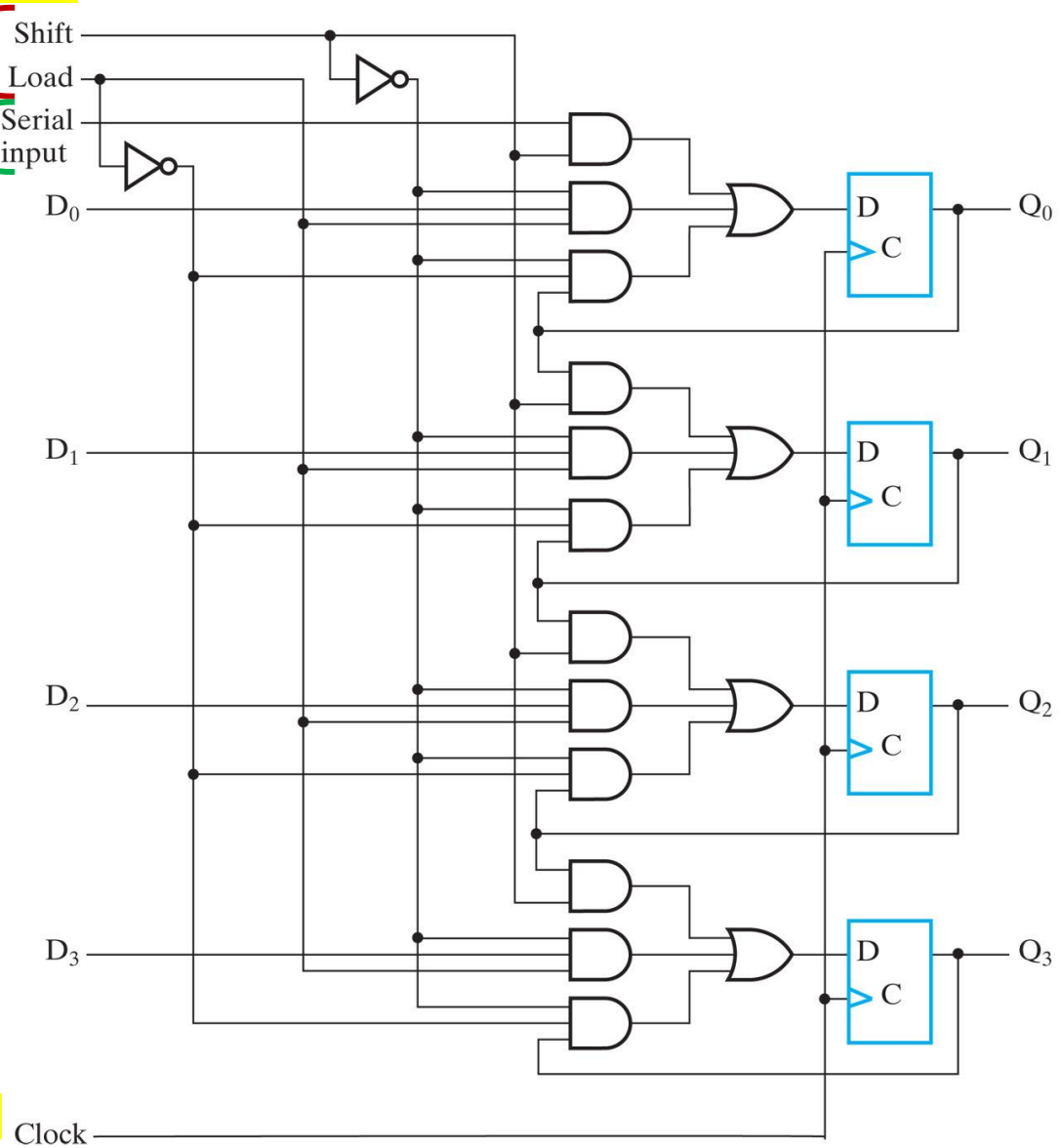
Abilita il caricamento del dato D_i

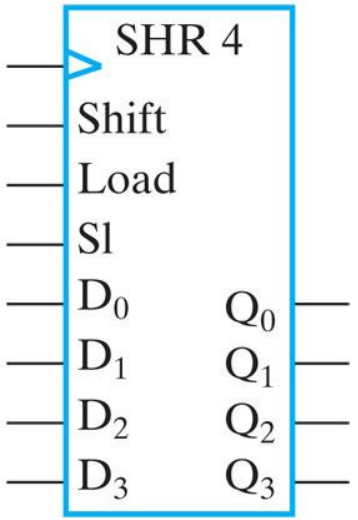
Ripristina il valore corrente (carica il valore dell'uscita del FF nel suo ingresso) se non è richiesto alcun cambiamento

Shift register con caricamento parallelo

2 segnali di controllo (scorrimento, caricamento)

1 segnale di ingresso

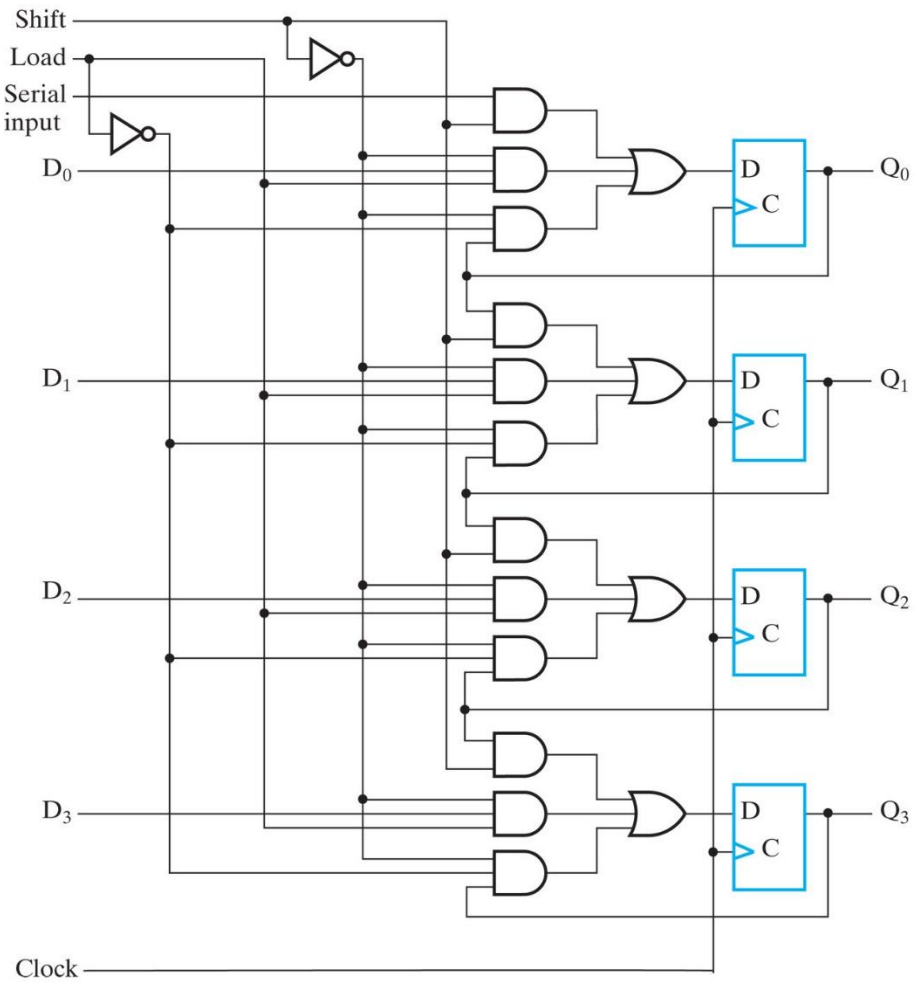




Clock collegato direttamente ai FF

Shift register con caricamento parallelo

Shift	Load	Operation
0	0	No change (Hold)
0	1	Load parallel data
1	×	Shift left (down) from Q_0 to Q_3



Shift : Q ← sl Q

$\overline{\textbf{Shift}} \cdot \textbf{Load}$ **: Q ← D**

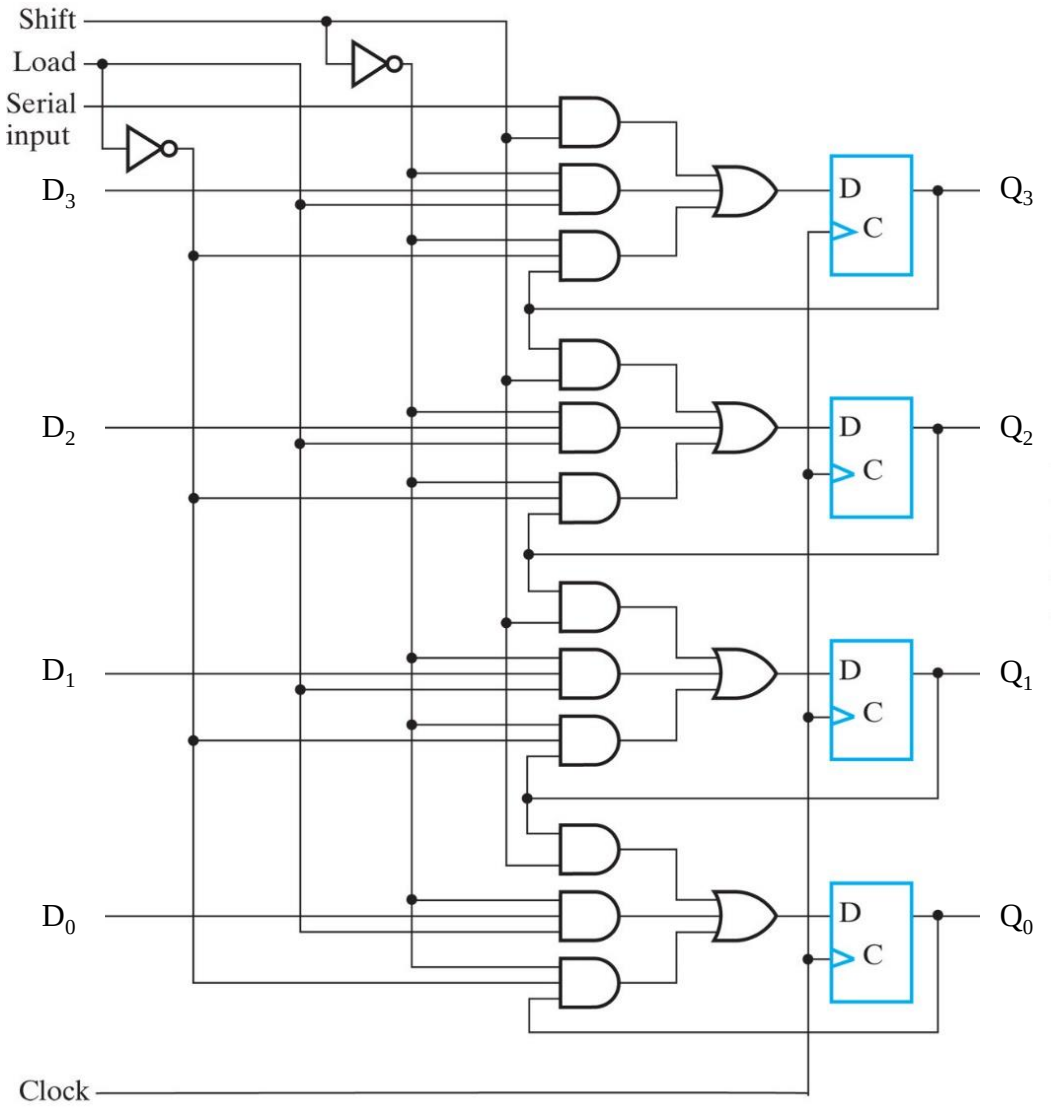
- se Shift = 1 (indipendentemente da Load): al successivo ciclo di clk, SI viene trasferita a Q_0, Q_0 a Q_1, Q_1 a Q_2, Q_2 a Q_3 cioè i bit **scorrono verso sinistra** (dal LSB Q_0 verso il MSB Q_3)
- se Shift = 0 e Load = 1: caricamento parallelo dei dati D_i all'ingresso di ogni FF, al successivo ciclo sono trasferiti alle uscite Q_i
- se Shift = 0 e Load = 0: mantiene i valori precedenti (hold - no change), al ciclo successivo le uscite Q_i vengono ripristinate allo stesso valore. Questa operazione è implicita se nessuna delle due precedenti condizioni è soddisfatta

Shift register con caricamento parallelo

- Per lo **scorrimento in direzione opposta** (verso destra, cioè dal MSB al LSB) il circuito è identico, ma i bit D_i e Q_i sono in ordine opposto
- In linguaggio RTL:

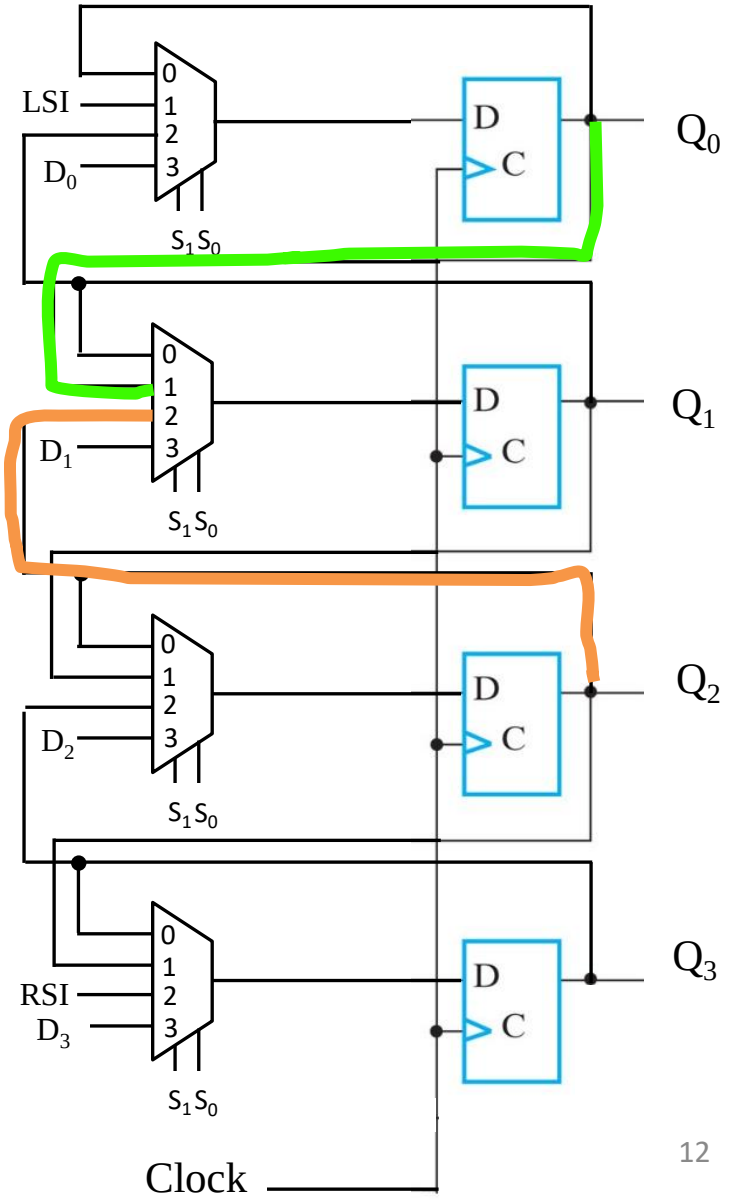
$$\mathbf{Shift} : \mathbf{Q} \leftarrow \mathbf{sr\ Q}$$

$$\overline{\mathbf{Shift}} \cdot \mathbf{Load} : \mathbf{Q} \leftarrow \mathbf{D}$$



Shift register bidirezionale

- Finora abbiamo visto registri di scorrimento unidirezionali. Per realizzare uno **shift register bidirezionale**, si aggiunge una porta AND (per abilitare lo scorrimento nell'altra direzione) a ciascuno stadio, nello schema per lo scorrimento unidirezionale
- Ciò equivale ad **interporre un mux 4-to-1 ad ogni stadio**, i cui ingressi di selezione S_0, S_1 controllano il tipo di operazione del registro
  - **$S_1 = 0$, $S_0 = 1$: shift left**
  - **$S_1 = 1$, $S_0 = 0$: shift right**



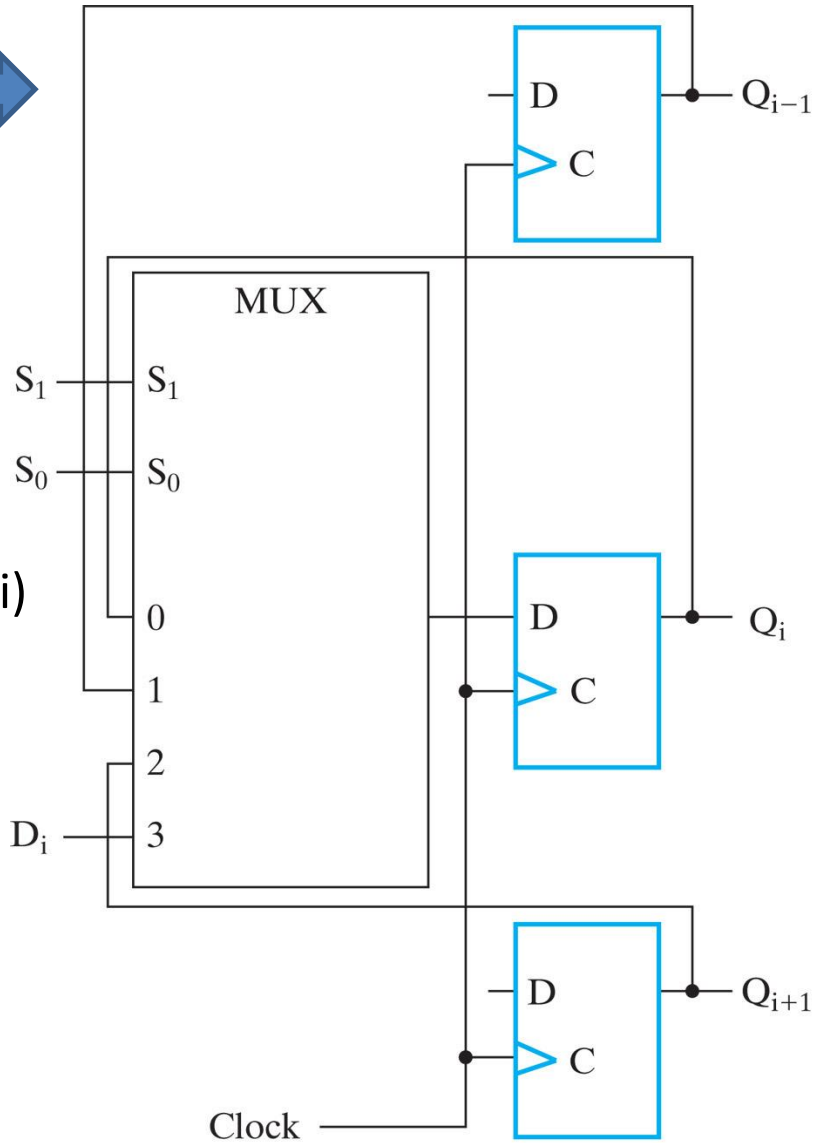
Shift register bidirezionale

Schema logico di uno stadio

- In linguaggio RTL:

$$\mathbf{S_1 \cdot \overline{S_0}:\ Q \leftarrow sr\ Q}$$
$$\mathbf{\overline{S_1} \cdot S_0:\ Q \leftarrow sl\ Q}$$
$$\mathbf{S_1 \cdot S_0:\ Q \leftarrow D}$$

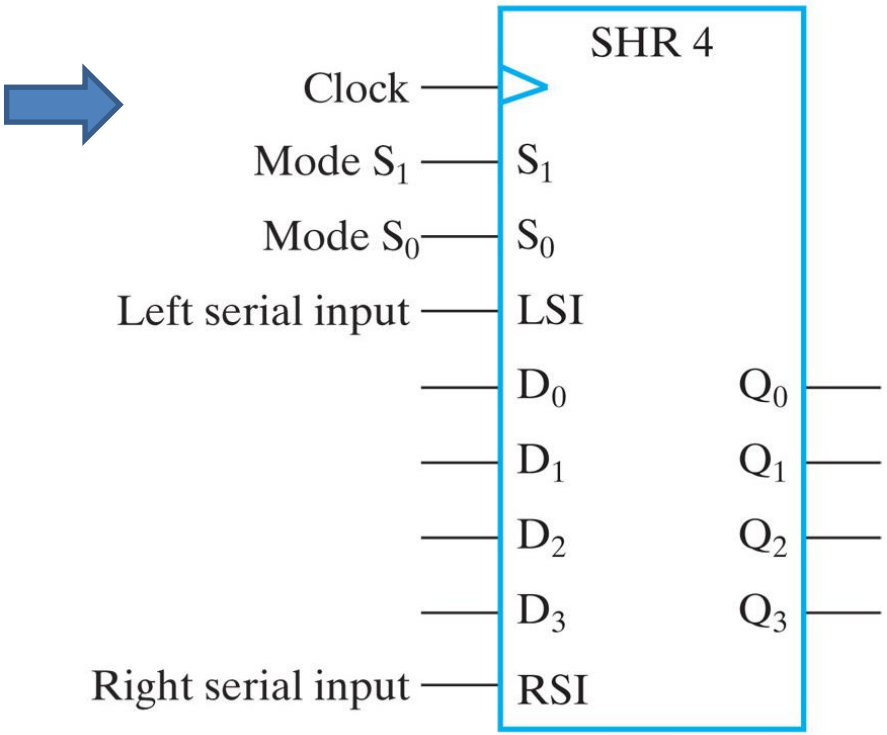
(operazione di hold è implicita quando non è soddisfatta nessuna delle precedenti condizioni)



Mode Control		Register Operation
S_1	S_0	
0	0	No change (Hold)
0	1	Shift left
1	0	Shift right
1	1	Parallel load

Shift register bidirezionale

Simbolo dello shift register bidirezionale (chiamato anche universale)



LSI: Left Serial Input

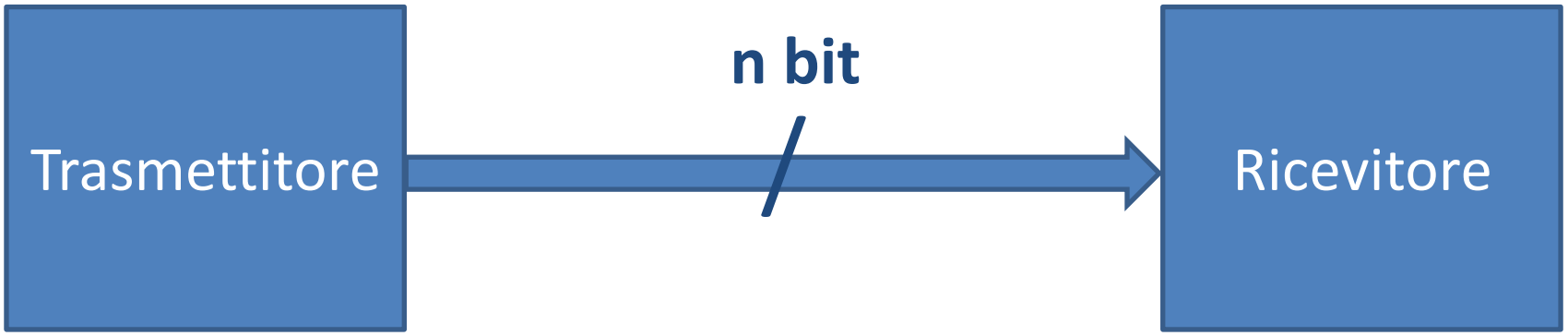
RSI: Right Serial Input

Q_0: output seriale per Right Shift

Q_3: output seriale per Left Shift

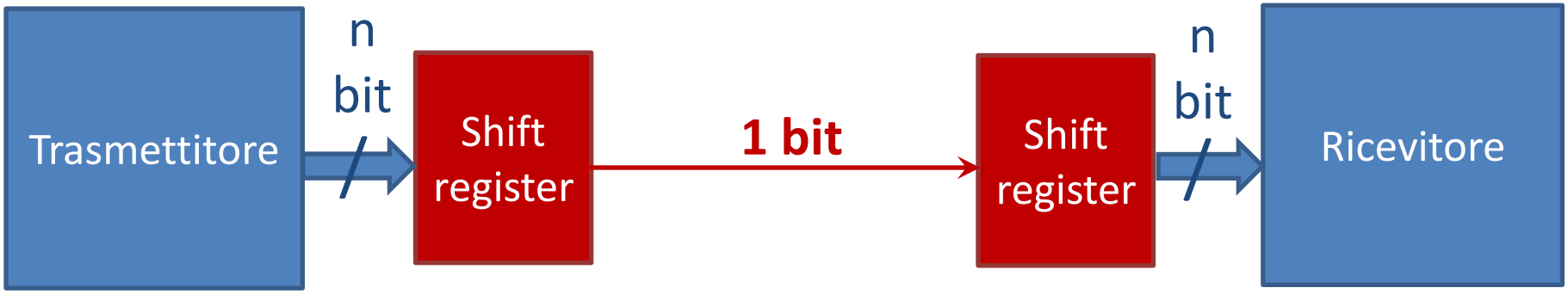
Dove si usa lo shift register?

- Uno degli usi più comuni dei registri a scorrimento è la realizzazione di **interfacce tra parti diverse e distanti tra loro di un sistema digitale**
- Supponiamo di dover trasferire dei dati (n bit) in parallelo da una parte all'altra di un circuito: usare n linee di dati può essere costoso, se la distanza tra le due parti è grande



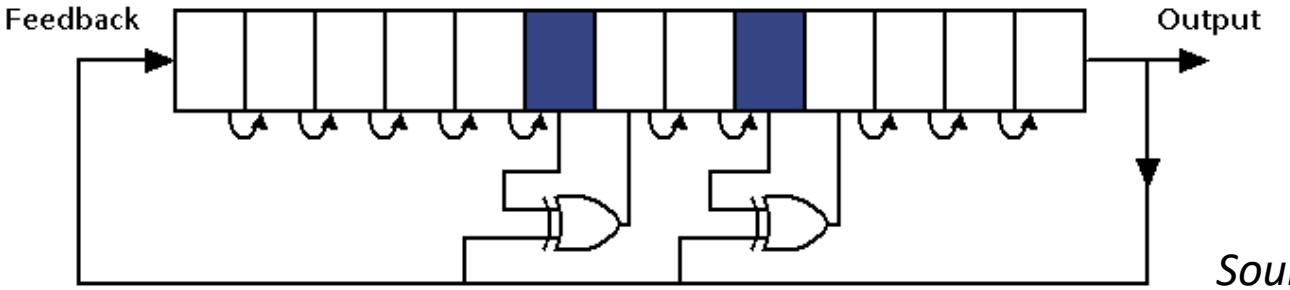
Dove si usa lo shift register?

- Con uno shift register usiamo **una sola linea di dati** (anziché n) trasmettendoli in **modo seriale**
  - Il trasmettitore carica i dati in parallelo in uno shift register
  - I dati vengono trasmessi in serie lungo la linea comune
  - Il ricevitore riceve serialmente i dati in uno shift register e quando sono arrivati tutti può prenderli in parallelo dall'uscita del registro



Dove si usa lo shift register?

- Sono impiegati anche in altri tipi di operazioni, per esempio:
  - **Moltiplicazione**: è la somma di più prodotti parziali, opportunamente shiftati di un certo numero di posizioni
  - **Generatore di numeri pseudo-casuali**: dando in ingresso ad uno shift register lo XOR tra alcuni dei bit memorizzati all'interno del registro, si ottiene in uscita una sequenza di numeri pseudo-casuali



Source: Wikipedia

  - **Linea di ritardo**: trasmette un segnale ad un elemento del sistema con un certo ritardo rispetto al momento in cui è stato generato (si può variare il ritardo cambiando la frequenza di clock o prelevando il segnale su una diversa uscita del registro)

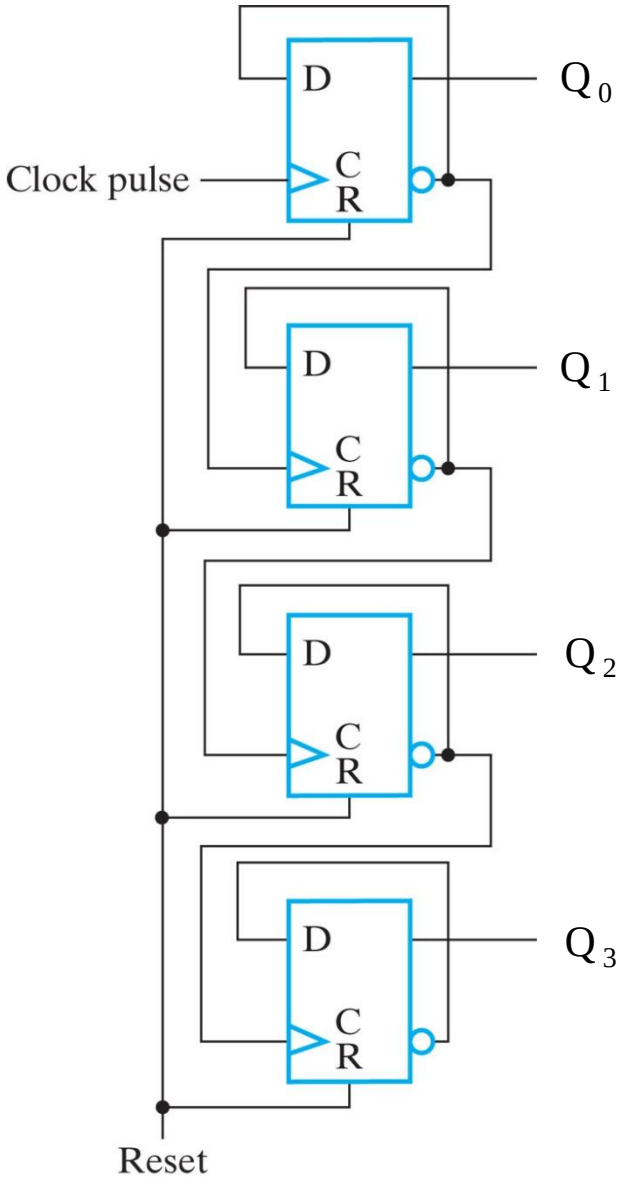
Contatori

Definizione e categorie di contatori

- Il contatore è un registro che, all'applicazione di una serie di ingressi, attraversa una **sequenza predefinita di stati** (numeri binari, sequenza casuale, etc.), **fornendo un conteggio in uscita**
- Un **contatore binario a n bit** è costituito da n flip-flop e fornisce in uscita un conteggio binario da 0 a $2^n - 1$
- Un contatore può appartenere a due categorie:
  - In un **contatore ripple** (= a catena), le transizioni in uscita ai flip-flop forniscono il clock agli altri flip-flop nel sistema. Si tratta quindi di un contatore asincrono
  - Un **contatore sincrono** è sincronizzato da un clock in comune, quindi ogni transizione in uscita (cambiamento di stato) avviene in corrispondenza dei fronti del clock

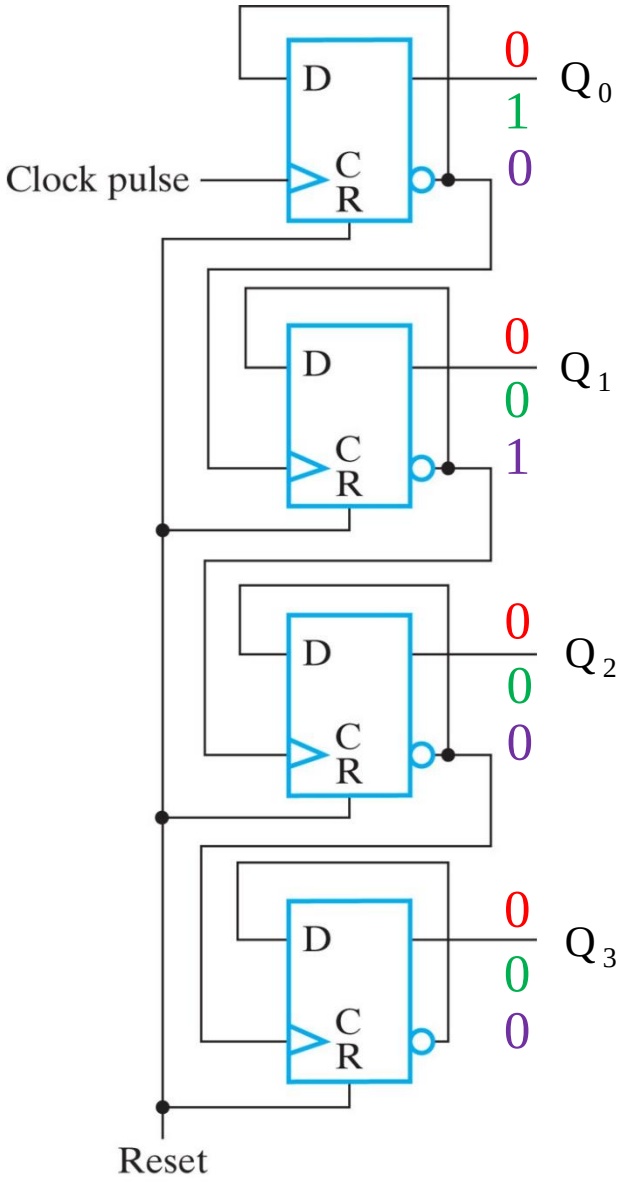
Contatore ripple

Contatore ripple binario a 4 bit (verso l'alto)



- Catena di 4 D flip-flop, le cui uscite **Q_0, Q_1, Q_2, Q_3 rappresentano il conteggio** in uscita
- Il FF corrispondente al LSB Q_0 riceve direttamente il clock del sistema. Ogni FF successivo riceve come **clock l'uscita negata del FF precedente**
- L'**ingresso di ogni FF è collegato alla sua uscita negata**: al fronte di salita del corrispondente C, ogni FF inverte il suo stato
- Un segnale di Reset asincrono consente di azzerare il contenuto del registro

Contatore ripple binario a 4 bit (verso l'alto)

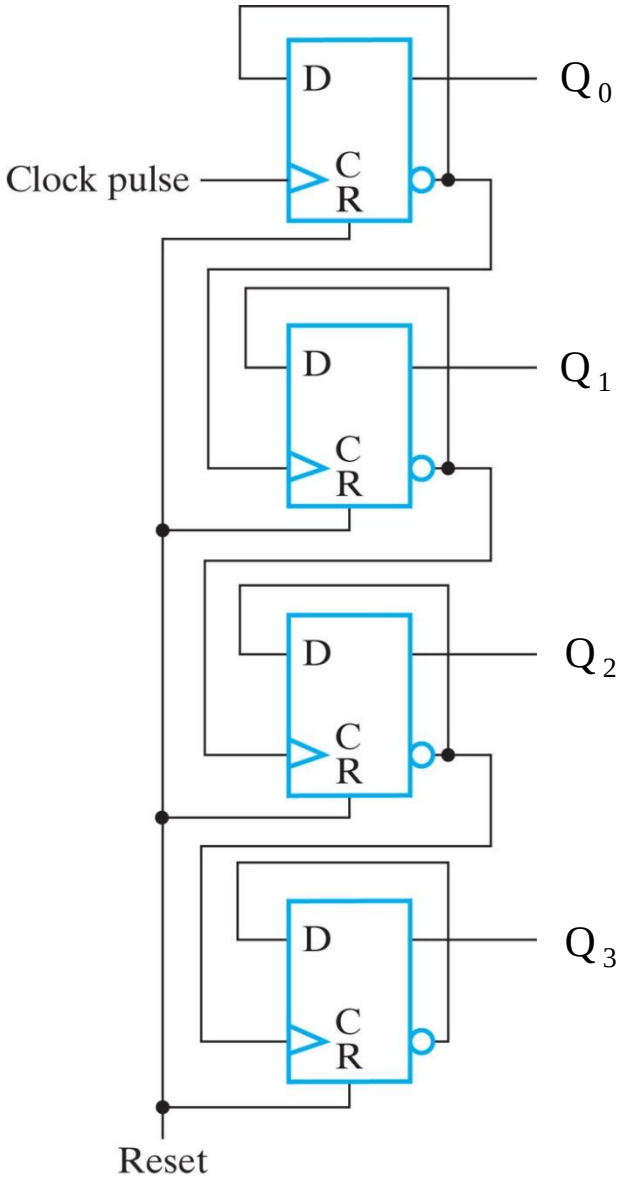


- Reset (Q_i posti a '0'): l'uscita (conteggio) è «0000»
- Al primo fronte di salita del clock:
  - FF_0 viene triggerato: Q_0 va da 0 a 1 e $\overline{Q_0}$ va da 1 a 0
  - I rimanenti FF non sono triggerati (Q_1,Q_2,Q_3 invariati)

  => l'uscita (conteggio) passa da «0000» a «0001»
- Al secondo fronte di salita del clock:
  - FF_0 viene triggerato: Q_0 va da 1 a 0 e $\overline{Q_0}$ va da 0 a 1
  - FF_1 viene triggerato: Q_1 passa da 0 a 1
  - FF_2-FF_3 non sono triggerati (Q_2 e Q_3 invariati)

  => l'uscita (conteggio) passa da «0001» a «0010»
- Quando Q_i cambia da 0 a 1 non è indotto alcun cambiamento in Q_{i+1}. Al contrario, ad ogni cambio di Q_i da 1 a 0, Q_{i+1} viene complementato
- Il tempo di transizione da un conteggio all'altro dipende dal numero di FF nella catena (cfr ripple carry adder)

Contatore ripple binario a 4 bit (verso l'alto)



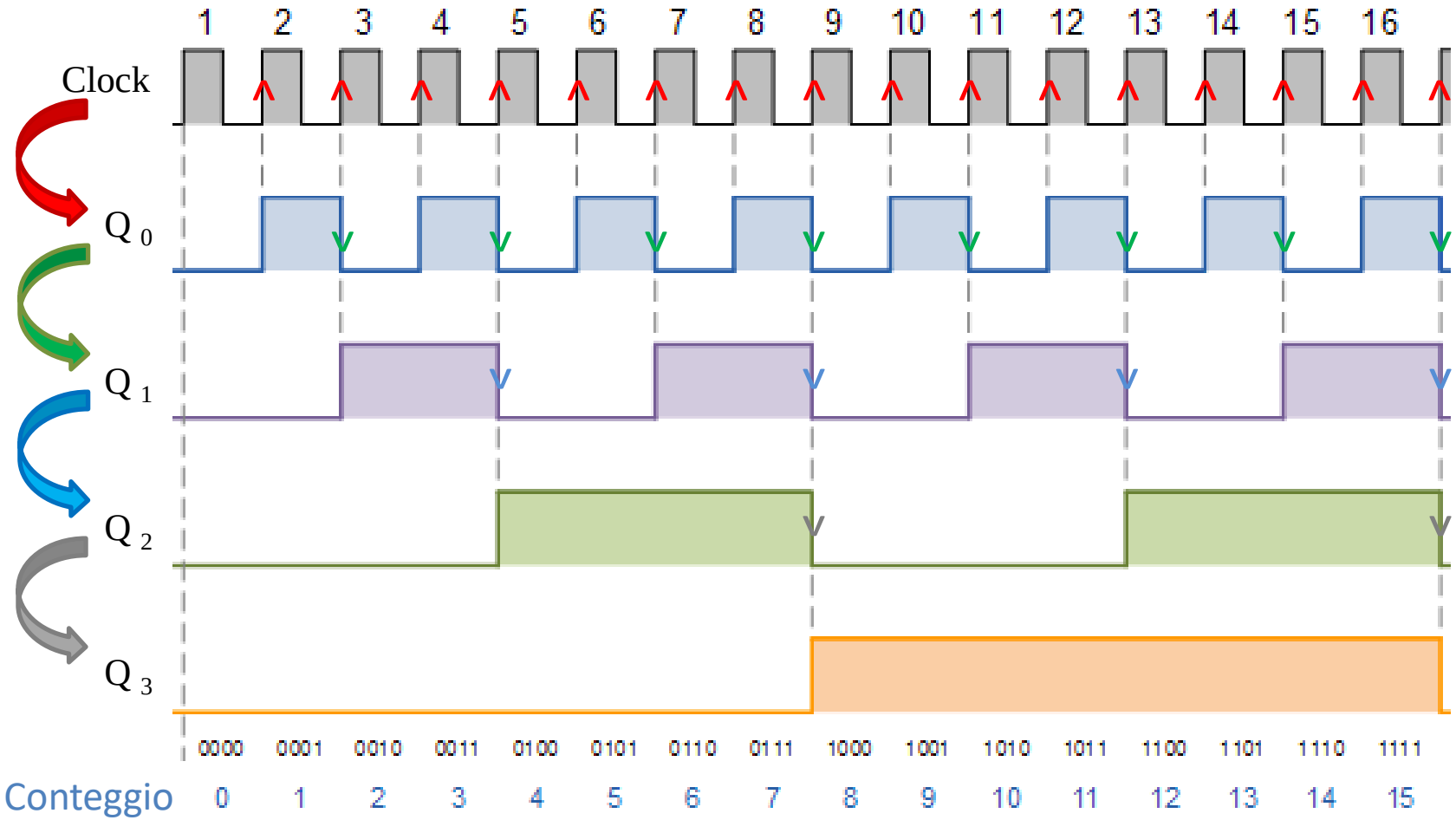
Upward Counting Sequence

Q_3	Q_2	Q_1	Q_0
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1
1	0	1	0
1	0	1	1
1	1	0	0
1	1	0	1
1	1	1	0
1	1	1	1

Conteggio verso l'alto

Q_0 commuta ad ogni fronte del clock del sistema, Q_1 commuta ogni volta che Q_0 va da 1 a 0, Q_2 commuta ogni volta che Q_1 va da 1 a 0, e così via…

Contatore ripple binario a 4 bit: Diagramma temporale



Contatore ripple binario a 4 bit (verso il basso)

- Se l'ingresso di **clock di ciascun FF è collegato all'uscita diretta** (anziché negata) del FF precedente, otteniamo invece un **conteggio verso il basso**
- Q_0 commuta ad ogni fronte del clock del sistema, Q_1 commuta ogni volta che Q_0 va da 0 a 1, Q_2 commuta ogni volta che Q_1 va da 0 a 1, e così via…

Downward Counting Sequence

Q_3	Q_2	Q_1	Q_0
1	1	1	1
1	1	1	0
1	1	0	1
1	1	0	0
1	0	1	1
1	0	1	0
1	0	0	1
1	0	0	0
0	1	1	1
0	1	1	0
0	1	0	1
0	1	0	0
0	0	1	1
0	0	1	0
0	0	0	1
0	0	0	0

Conteggio verso il basso

Contatore ripple: pro e contro

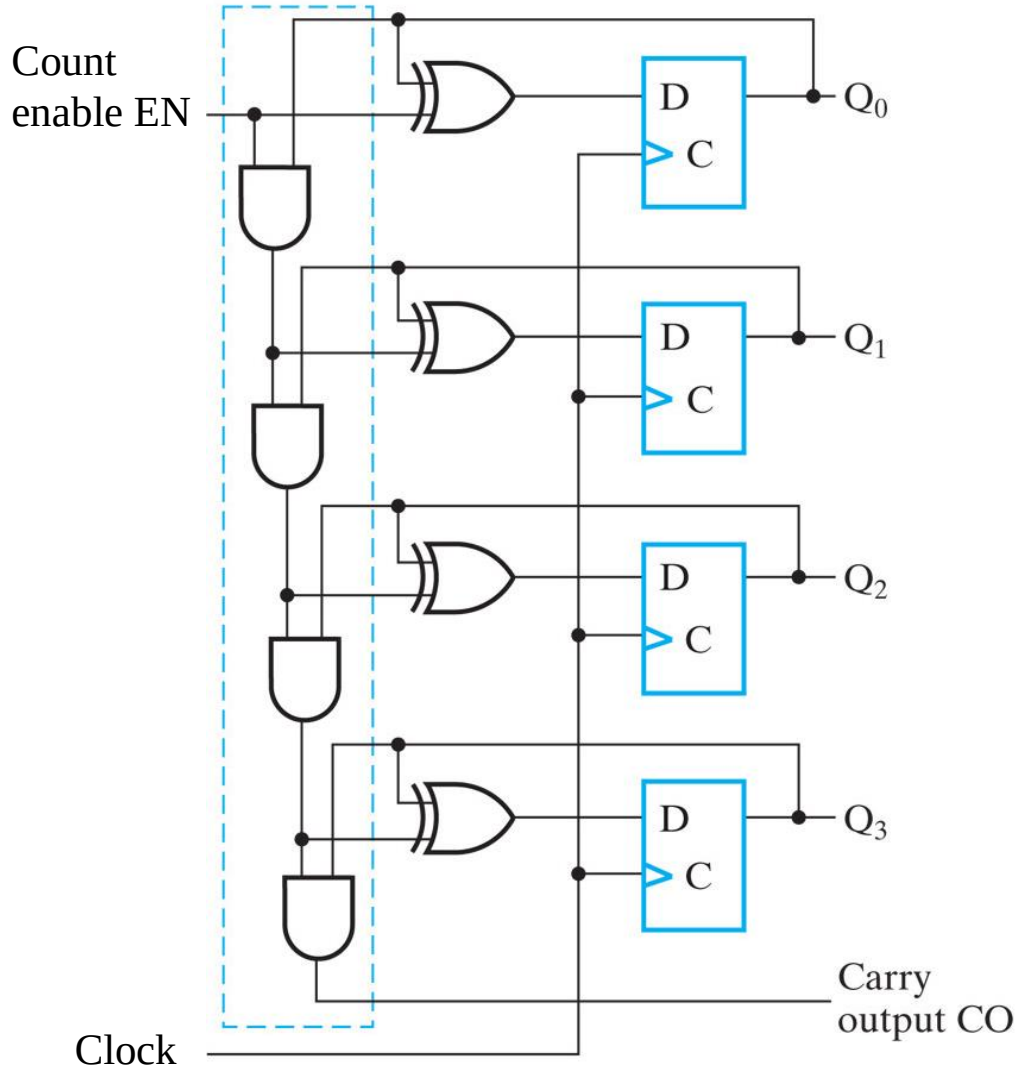
- I contatori ripple hanno il vantaggio di avere una **struttura semplice**
- Sono però dei circuiti asincroni, quindi i bit di uscita hanno **ritardi diversi** (specie se contengono un numero elevato di bit) dato che i flip-flop cambiano di stato uno alla volta in successione, essendo triggerati ognuno dal precedente, il che può essere un problema
- I contatori ripple possono essere impiegati in circuiti che non hanno vincoli stringenti sulla velocità di funzionamento (frequenze di clock sufficientemente basse)

Contatori sincroni binari

- A differenza del contatore ripple, in un contatore sincrono tutti i **flip-flop sono regolati dallo stesso segnale di clock**, quindi vengono triggerati tutti allo stesso tempo anziché uno dopo l'altro (come avveniva nel ripple counter)

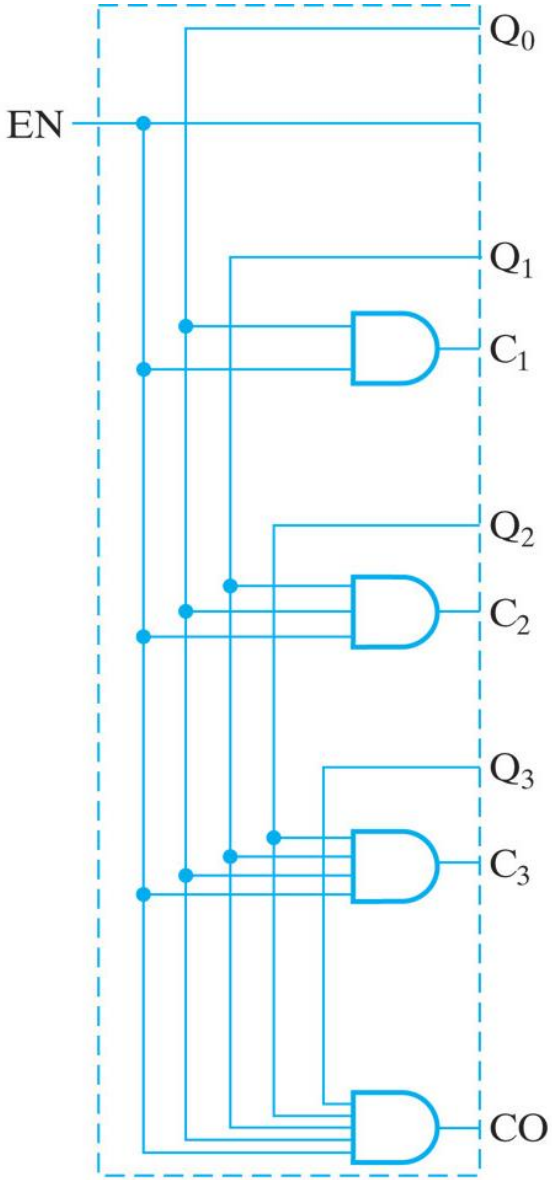
Contatori binari sincroni

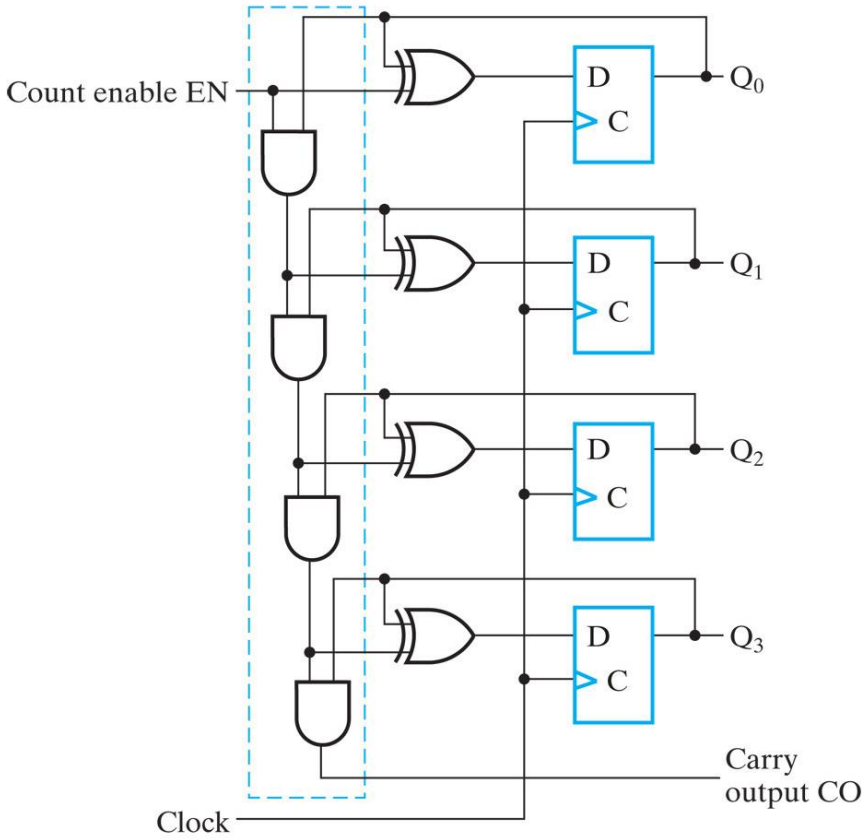
Contatore sincrono binario seriale



- EN (input): **abilita il conteggio**
- CO (carry output): permette di realizzare contatori a più stadi con struttura gerarchica
- Se **EN = 1**, ad ogni fronte di clk il valore immagazzinato nel registro è incrementato di 1
- Porte XOR eseguono la somma (somma 1 al valore immagazzinato nel registro), le porte AND propagano il carry da uno stadio al successivo
- C'è un ritardo di propagazione di 4 porte AND per generare CO, 3 AND + 1 XOR per Q_3

Contatore sincrono binario parallelo

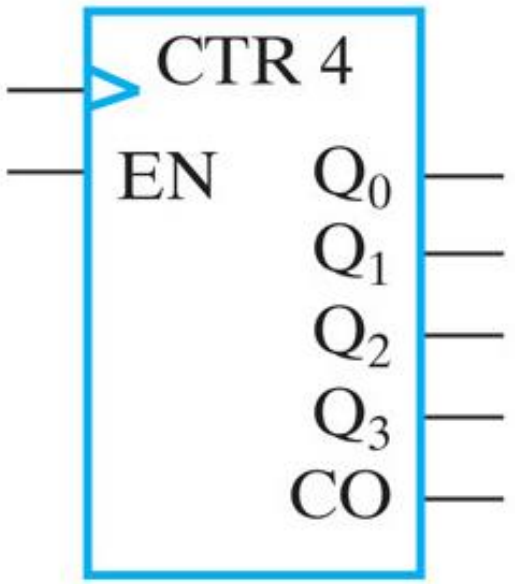




Sostituendo la catena di porte AND con altrettante AND in parallelo a più ingressi otteniamo un contatore più veloce

Contatore sincrono binario

Il simbolo del contatore sincrono binario a 4 bit è



Connettendo insieme due contatori sincroni binari a 4 bit (con l'uscita di carry del primo attaccata all'ingresso EN del secondo) otteniamo un contatore a 8 bit

Riepilogo

- Abbiamo visto possibili realizzazioni di due tra i blocchi più usati nei sistemi digitali: **shift register** e **contatori**
- Uno shift register è un registro costituito da una catena di flip-flop e fa scorrere lateralmente i bit verso destra, verso sinistra (shift register unidirezionale) o in entrambe le direzioni (bidirezionale o universale)
- Un contatore è costituito da una catena di flip-flop, le cui uscite forniscono un determinato conteggio (verso l'alto, basso, etc.)
- Un contatore ripple è un contatore asincrono, dove il clock di ogni flip-flop è fornito dal flip-flop precedente
- Un contatore sincrono è costituito da flip-flop connessi in catena e controllati tutti dallo stesso segnale di clock
  - Questo tipo di contatore ha una struttura leggermente più complessa rispetto al contatore ripple, ma è più veloce e meno soggetto a problematiche di temporizzazione

Disclaimer

Figures from *Logic and Computer Design Fundamentals*, Fifth Edition, GE Mano | Kime | Martin