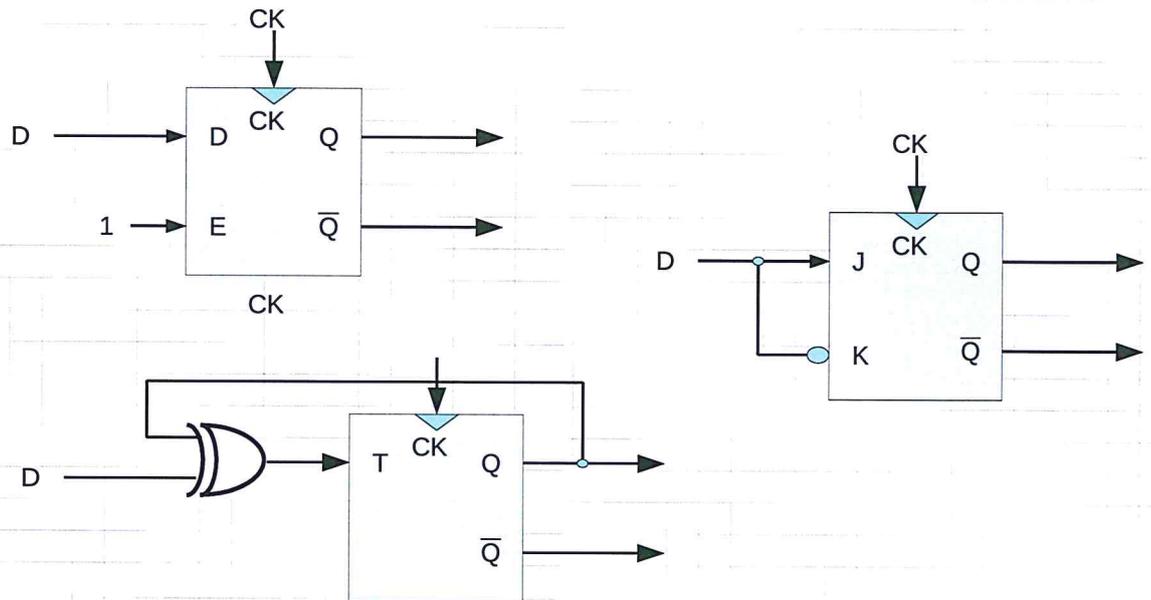


## Da flip-flop DE, JK, T a D



33

17

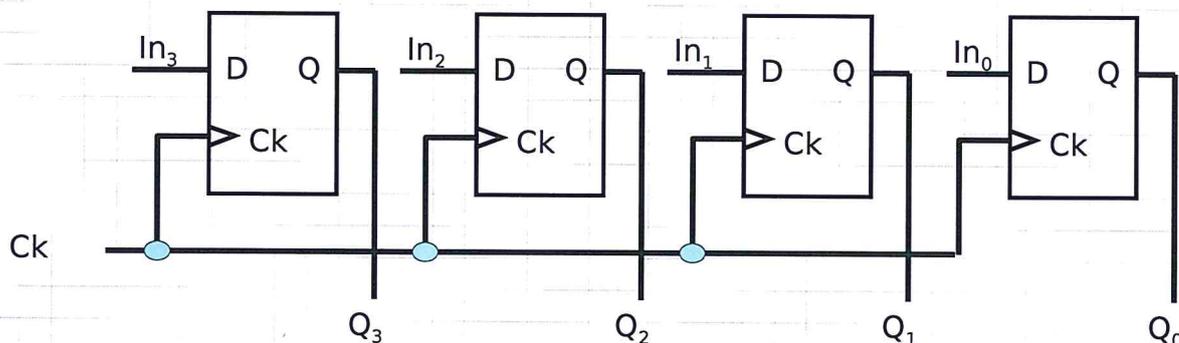
## I registri

- › Array di FF, in grado di memorizzare più bit di informazione
  - › Si caratterizzano per il modo con cui avviene la memorizzazione e la lettura dei dati
    - › **Parallelo** o **seriale** nel tempo
- › Possono eseguire manipolazione dei bit
  - › Scorrimenti (shift) destri o sinistri dei bit
- › Più funzionalità possono essere incorporate in un unico dispositivo
  - › Registro **universale**, usato anche all'interno dei microcontrollori

34

## Registro parallelo

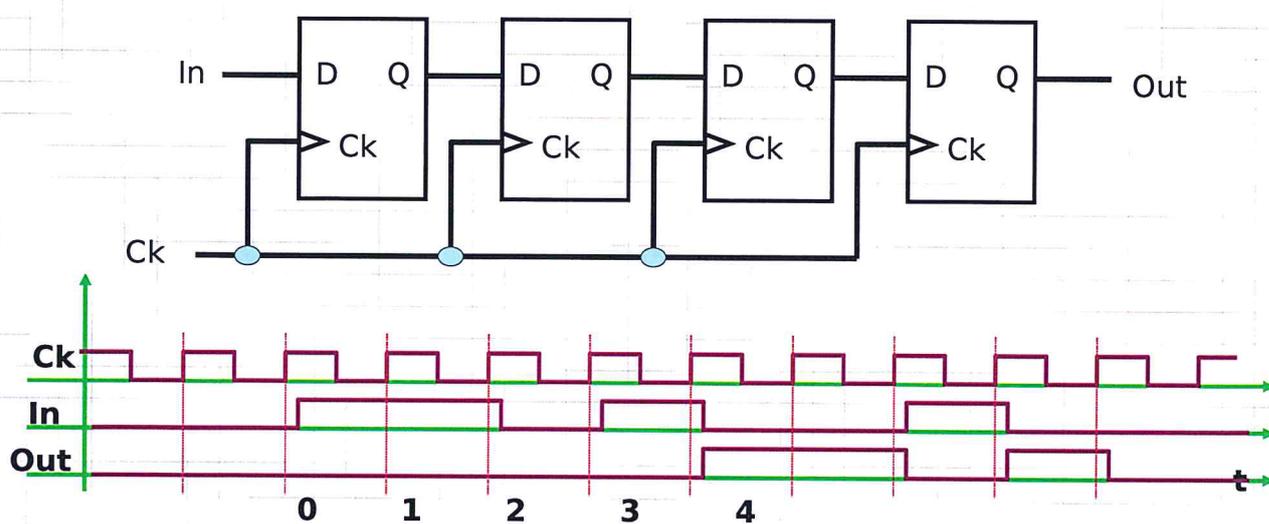
Parallel In - Parallel Out (PIPO)



35

## Registro a scorrimento (1) (shift register)

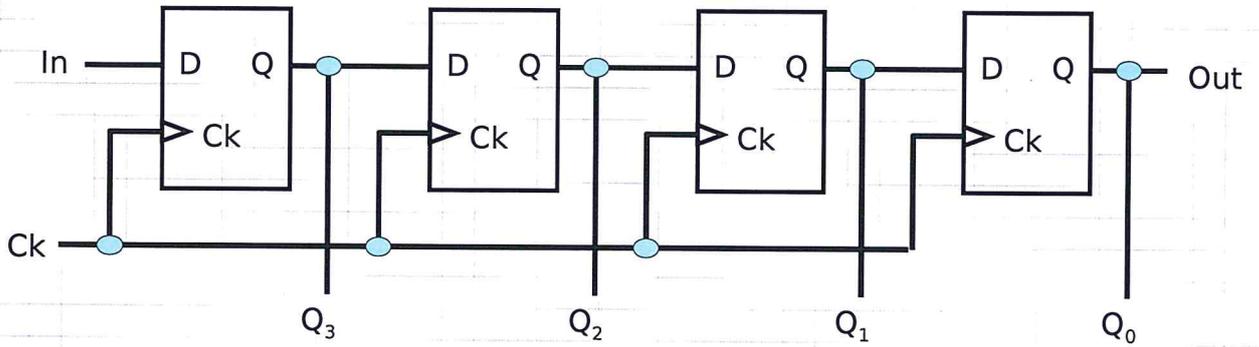
Serial In - Serial Out (SISO)



36

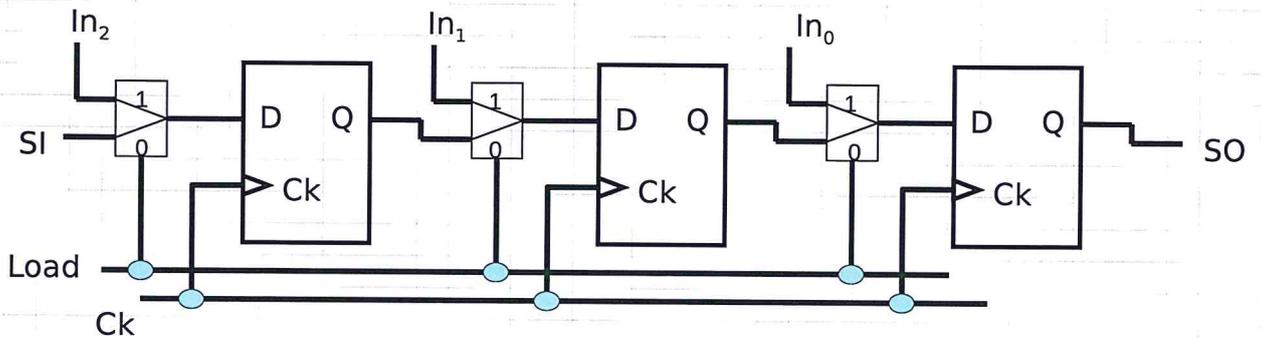
# Registro a scorrimento (2) (shift register)

Serial In - Parallel Out (SIPO)

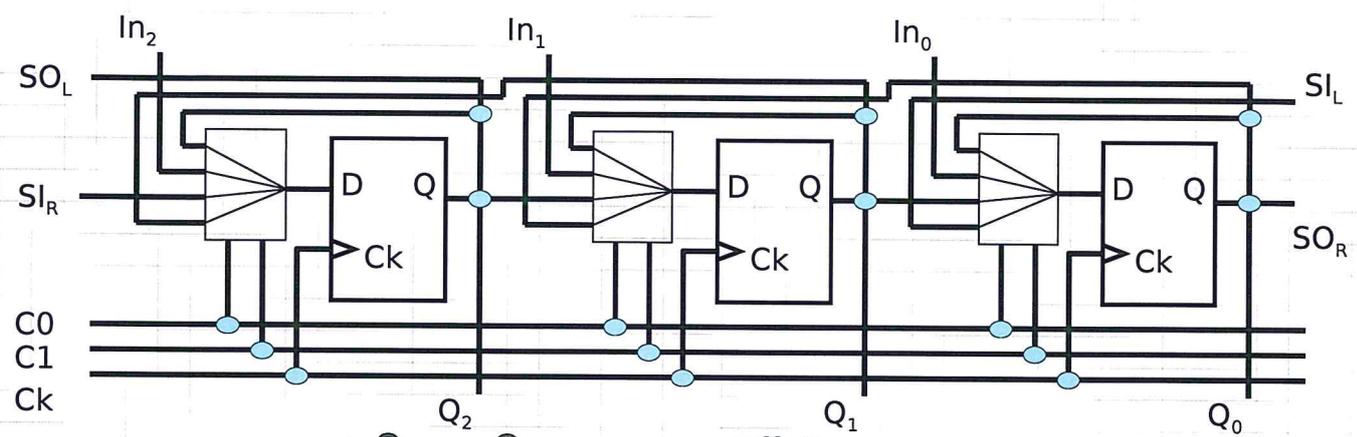


# Registro a scorrimento (3) (shift register)

Parallel In - Serial Out (PISO)



# Registro universale

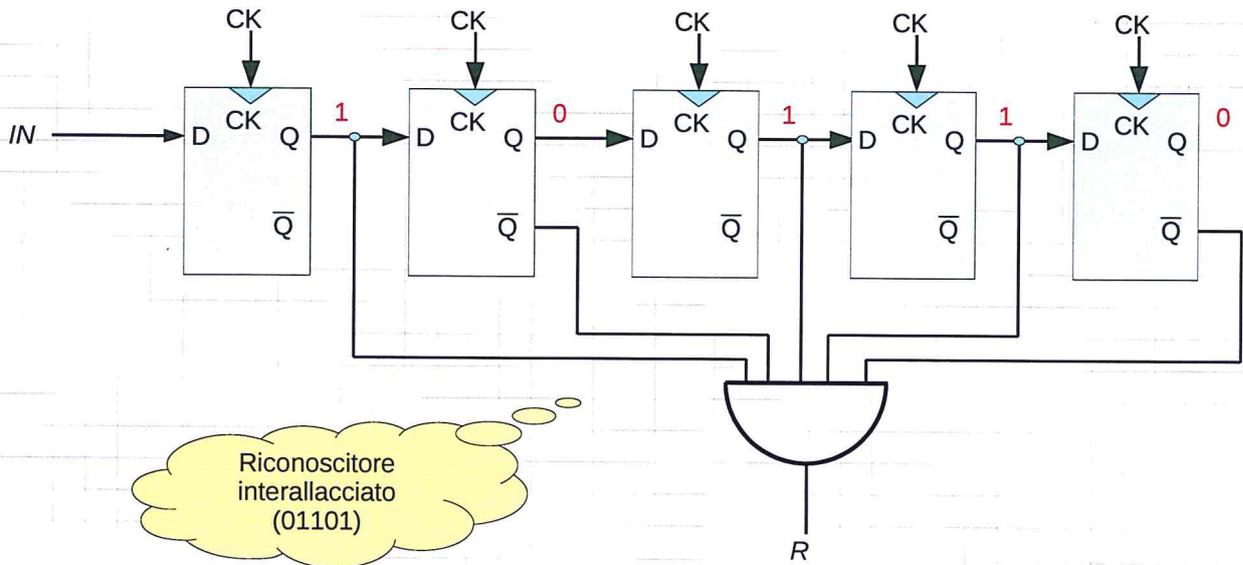


C <sub>1</sub>	C <sub>0</sub>	Effetto
0	0	Conserva
0	1	Ingresso parallelo
1	0	Shift destro
1	1	Shift sinistro

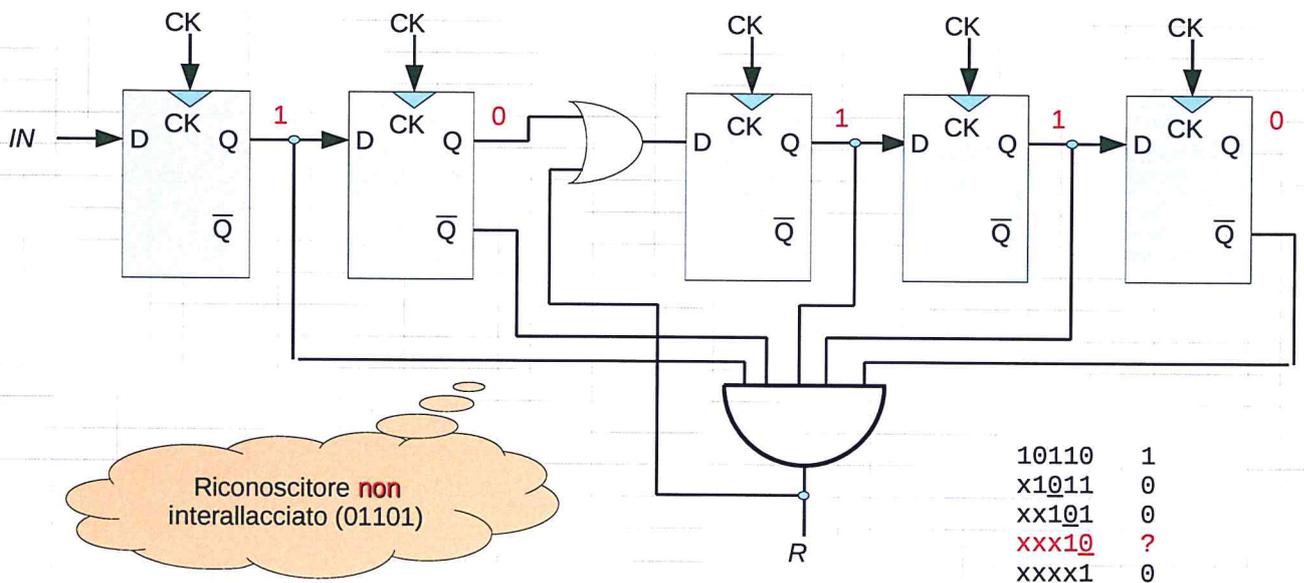
# Riconoscitori di sequenza

- > Macchine sequenziali con un uscita che si attiva quando l'ingresso presenta in sequenza una serie preordinata di valori
  - > Esempio di sequenza: 01101
  - > Riconoscimento di sequenze **interallacciate**
    - > I bit di una sequenza riconosciuta possono essere considerati parte di una nuova sequenza
      - > Nell'esempio, lo ...01 finale può essere considerato nuovamente
  - > Sequenze **non interallacciate**
    - > Una volta riconosciuti, i bit non si usano più
- > Esempio di funzionamento
  - > In: ...0111011011011011010110110101000...
  - > Out\_I: ...0000000010010010010000100100000...
  - > Out\_NI: ...0000000010000010000000100000000...

# Riconoscitori con shift (1)



# Riconoscitori con shift (2)



E' SEMPRE POSSIBILE?

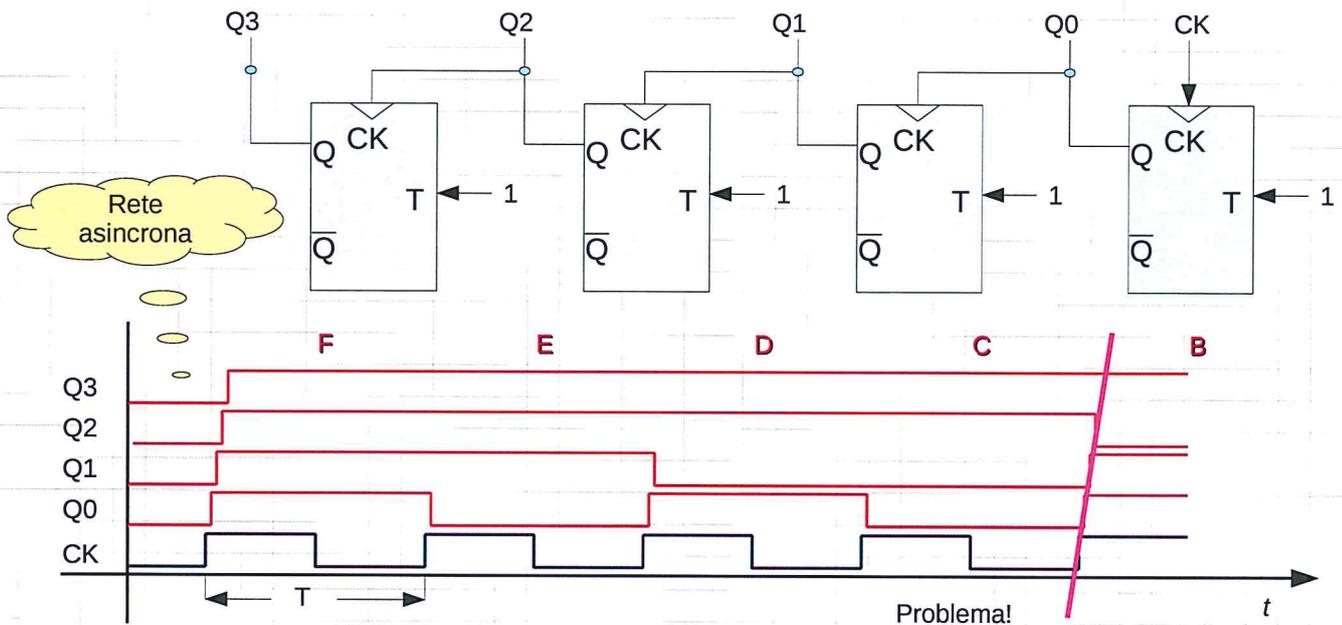
## Contatori (1)

- Macchine sequenziali
  - Sincrone o asincrone
- Interfaccia
  - Hanno in ingresso un'onda quadra
  - Generano una sequenza di valori predefiniti
    - Ogni valore è ricavato dal precedente
      - Sommando o sottraendo un valore costante

## Contatori (2)

- Alla **fine del conteggio** possono comportarsi diversamente
  - Ripartire da 0
  - Modulo  $n$
  - Restare al valore massimo raggiunto (saturazione)
- Possono avere diversi ingressi per **funzioni opzionali**
  - Azzeramento (Reset)
  - Abilitazione (Enable)
  - Caricamento parallelo (Count/ $\overline{\text{Load}}$ )
  - Direzione del conteggio (Up/ $\overline{\text{Down}}$ )

## Divisori di frequenza



45

## Contatori modulo $n$

- Macchine sequenziali che generano, ciclicamente e in ordine, valori tra 0 e  $n - 1$ 
  - Al termine del conteggio ripartono da capo
  - Categoria particolare: contatori modulo  $2^m$ 
    - Richiedono  $m$  variabili di stato
- Possibili variabili di controllo
  - Enable: 1 (abilita il conteggio); 0 (conserva)
  - Reset: 1 (azzerà); 0 (conta normalmente)
  - Preset: 1 (tutti i bit accesi); 0 (conta normalmente)
  - Up/Down: 1 (conteggio crescente); 0 (decrescente)
  - Load: 1 (carica il valore); 0 (conta normalmente)

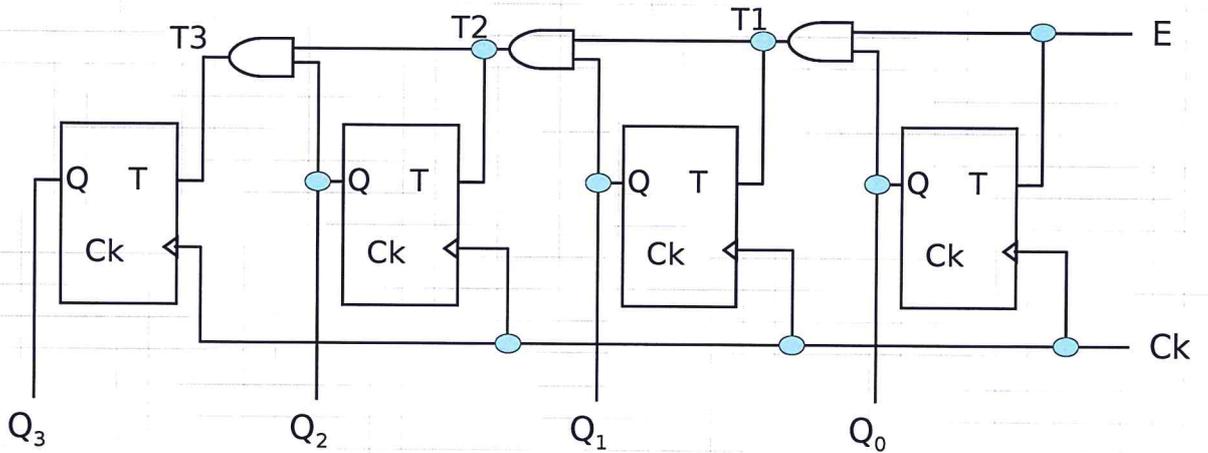
46

# Contatore sincrono modulo $2^n$

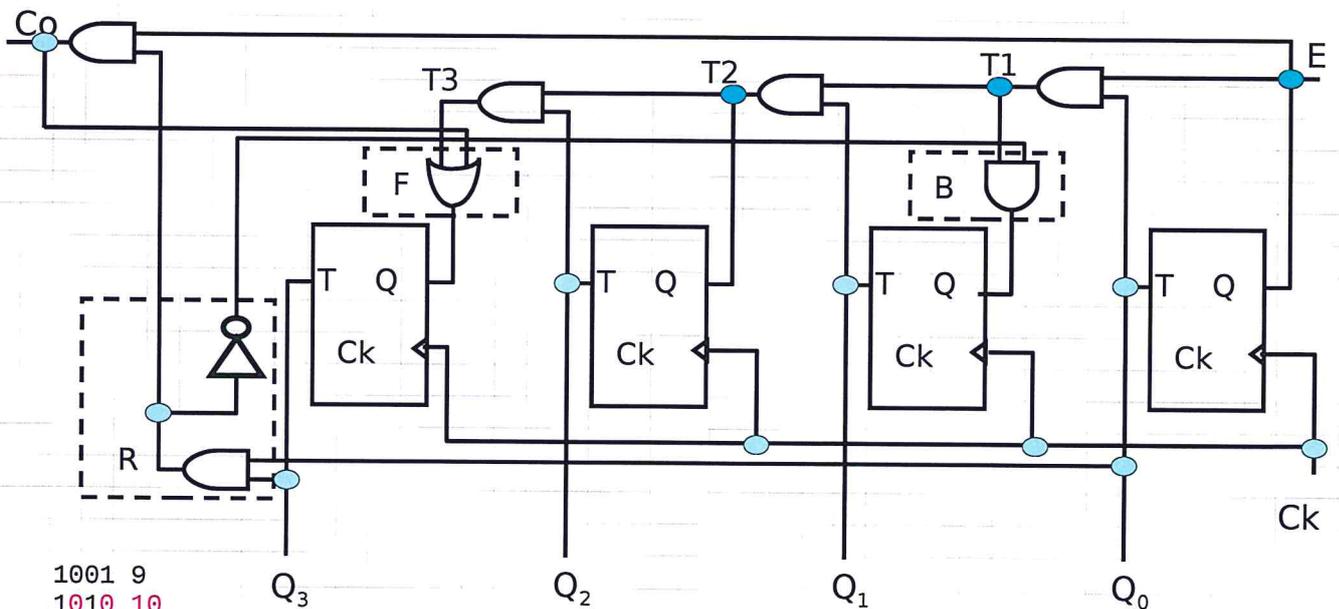
$$T_i = E Q_{i-1} Q_{i-2} \dots Q_0$$

$$T_{i+1} = E Q_i Q_{i-1} Q_{i-2} \dots Q_0 = T_i Q_i$$

Un bit commuta se:  
 1) il contatore è abilitato  
 2) i bit di peso inferiore al clock precedente sono tutti a 1



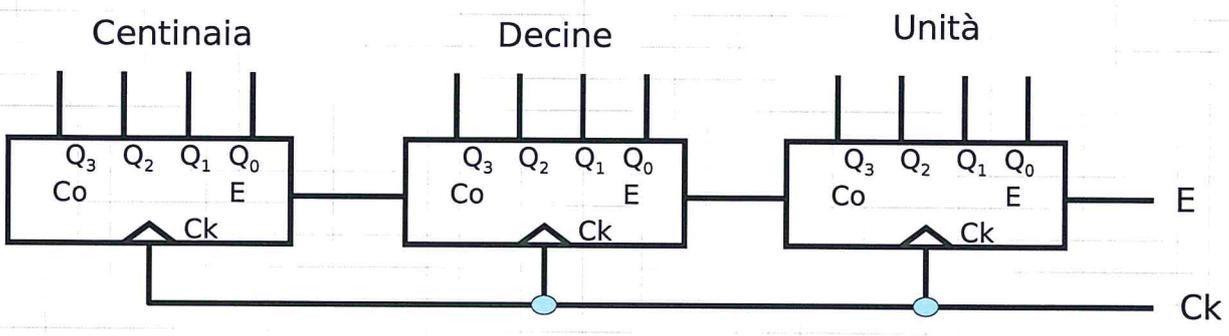
# Contatore sincrono modulo 10



1001 9  
 1010 10  
 0000 0  
 F-B-

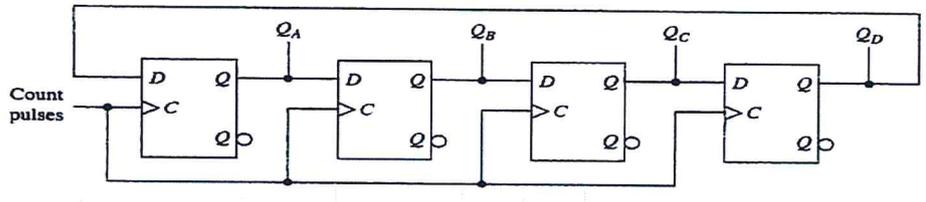
17.9

# Contatori decadici in cascata

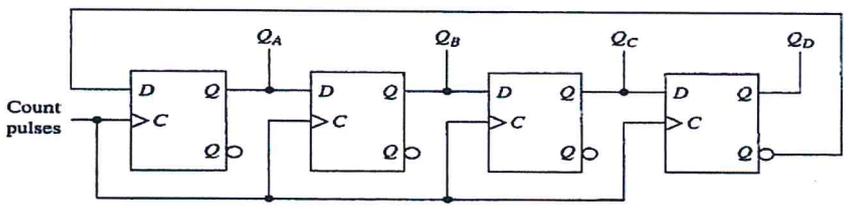


17.10

# Contatori ad anello e Johnson

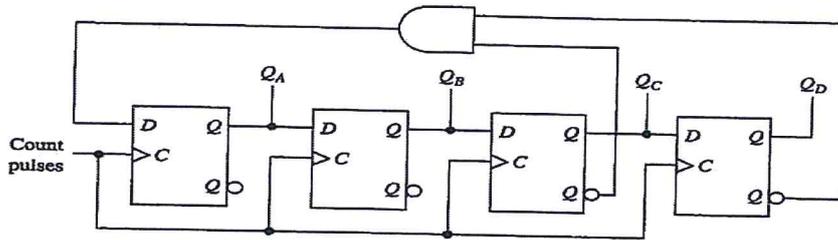


$Q_A$	$Q_B$	$Q_C$	$Q_D$
1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1
1	0	0	0
etc.			



$Q_A$	$Q_B$	$Q_C$	$Q_D$	And-gate inputs
0	0	0	0	$\overline{Q_A} \overline{Q_D}$
1	0	0	0	$Q_A \overline{Q_D}$
1	1	0	0	$Q_B \overline{Q_C}$
1	1	1	0	$Q_C \overline{Q_D}$
1	1	1	1	$Q_A Q_D$
0	1	1	1	$\overline{Q_A} Q_B$
0	0	1	1	$\overline{Q_B} Q_C$
0	0	0	1	$Q_C Q_D$
0	0	0	0	
etc.				

# Contatore Johnson mod 7



$Q_A$	$Q_B$	$Q_C$	$Q_D$
0	0	0	0
1	0	0	0
1	1	0	0
1	1	1	0
0	1	1	1
0	0	1	1
0	0	0	1
0	0	0	0

And-gate inputs  
 $\overline{Q_A} \overline{Q_D}$   
 $Q_A \overline{Q_B}$   
 $Q_B \overline{Q_C}$   
 $Q_C \overline{Q_D}$   
 $\overline{Q_A} Q_B$   
 $\overline{Q_B} Q_C$   
 $\overline{Q_C} Q_D$

USCITA  
 SIMILE AL  
 RING.

## Il problema dell'inizializzazione

- > Una macchina sequenziale con  $n$  flip-flop ha  $2^n$  stati
- > Se la sequenza del contatore ha  $m < 2^n$  valori, esistono valori non compresi nel conteggio
  - > Occorre porsi la domanda: come procede la sequenza se accidentalmente il contatore viene a trovarsi in uno di questi  $2^n - m$  valori?
  - > Possibili risposte
    - > In modo arbitrario, ma tornando dopo un numero finito di cicli a un valore appartenente alla sequenza (POSS OSSERVARE LO STATO?)
    - > In modo arbitrario, e senza mai tornare a un valore valido della sequenza
      - > Questa condizione è da evitare
      - > All'accensione lo stato dei flip-flop è (in genere) casuale

INGRESSI ASINCRONI