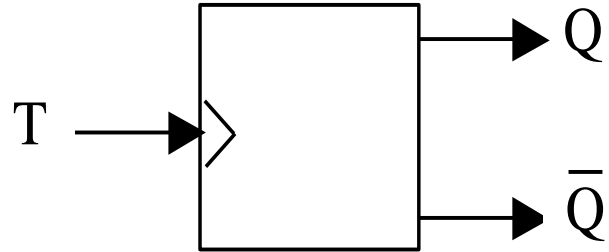


Flip-Flop a commutazione

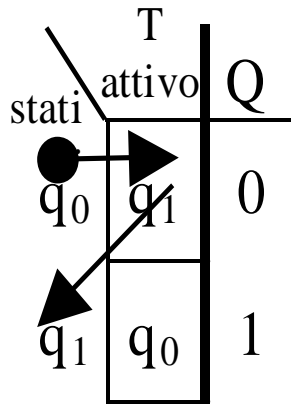
- un valore di ingresso provoca la *commutazione* dello stato piuttosto che indicarne esplicitamente il valore
- due tipi
 - **T**: ha solo l'ingresso di commutazione
 - **JK**: analogo all'RS, con ingressi $JK=11$ corrispondenti all'ingresso di commutazione

Flip-Flop T



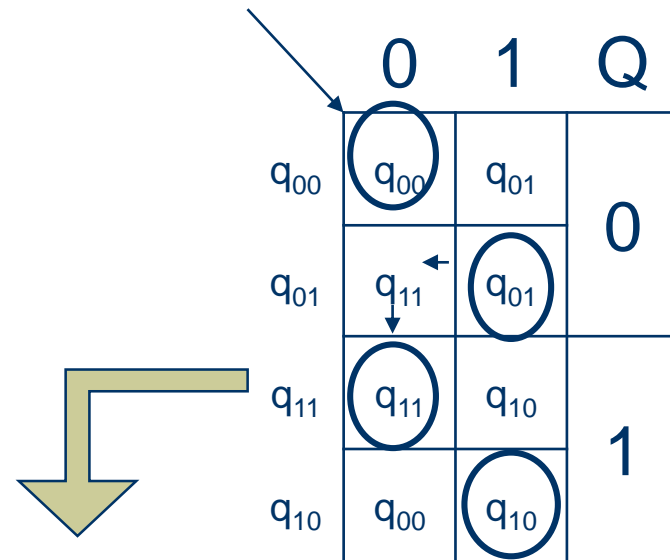
- Equazione di stato: $Q = \bar{Q}_p T + Q_p \bar{T}$
- La macchina che implementa questo funzionamento deve essere necessariamente asincrona o sincrona impulsiva

Flip-Flop T



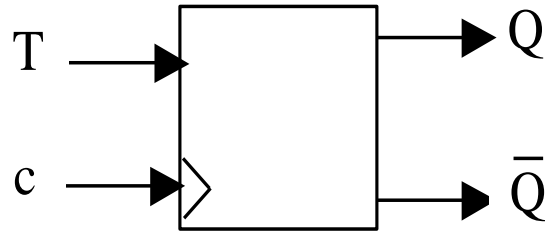
Flip-flop T come macchina
sincrona impulsiva

Flip-flop T come macchina
asincrona



q_{00} : l'uscita è 0 e sarà tale anche dopo la variazione di T;
 q_{01} : l'uscita è 0 e sarà 1 dopo la variazione di T;
 q_{11} : l'uscita è 1 e sarà tale anche dopo la variazione di T;
 q_{10} : l'uscita è 1 e sarà 0 dopo la variazione di T.

Flip-Flop T abilitato



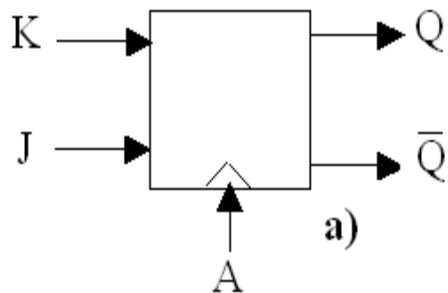
L'impulso sull'abilitazione c fa commutare il flip-flop se T è attivo.

In sostanza, T fa da abilitazione, mentre c diventa l'ingresso di commutazione

Flip-Flop JK

E' un flip-flop sincronizzato definito come segue:

- quando l'ingresso impulsivo di sincronizzazione A è attivo, i due ingressi a livello J, K si comportano rispettivamente come S, R se sono attivi separatamente..
- mentre producono la commutazione se lo sono simultaneamente.



$$Q = j\bar{Q}_p + Q_p\bar{k} = AJ\bar{Q}_p + Q_p\bar{K} + Q_p\bar{A}$$

Flip-Flop JK

KJ	00	01	11	10	Q
stat0	q ₀	q ₁	q ₁	q ₀	0
stat1	q ₁	q ₁	q ₀	q ₀	1

JK sincrono impulsivo

KJ	A=0				A=1				Q
stat0	00	01	11	10	00	01	11	10	0
q ₀₀	q ₀₀	q ₀₁	q ₀₁	q ₀₀	q ₀₀	q ₀₀	q ₀₀	q ₀₀	0
q ₀₁	q ₀₀	q ₀₁	q ₀₁	q ₀₀	--	q ₁₁	q ₁₁	--	
q ₁₀	q ₁₁	q ₁₁	q ₁₀	q ₁₀	--	--	q ₀₀	q ₀₀	1
q ₁₁	q ₁₁	q ₁₁	q ₁₀	q ₁₀	q ₁₁	q ₁₁	q ₁₁	q ₁₁	

JK asincrono edge-triggered

□ TABLE 4-10
Flip-Flop Excitation Tables

(a) JK Flip-Flop

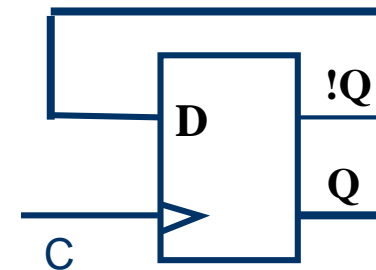
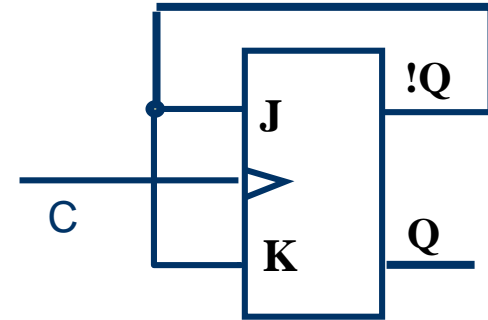
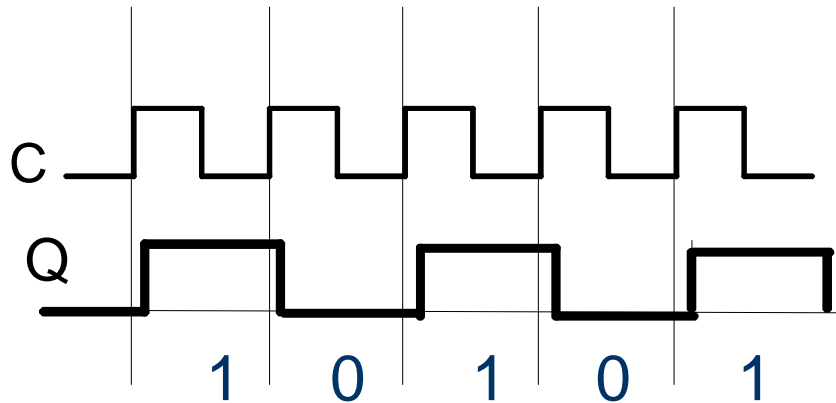
Q(t)	Q(t+1)	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

Contatori binari

- ◆ Un contatore è un registro che evolve secondo una sequenza predefinita di stati ordinati all'applicazione di un impulso di ingresso
 - L'impulso di ingresso o *impulso di conteggio* può coincidere o meno con il clock e può essere quindi presentato ad intervalli costanti o casuali
- ◆ Se la sequenza degli stati segue la sequenza dei numeri binari, il contatore si dice **binario**: un contatore binario a n bit è formato da n flip-flop e può contare in binario da 0 a $2^n - 1$ (contatore modulo 2^n)

Contatore modulo 2

- ♦ Il contatore binario più semplice è il **contatore modulo 2**, in grado di contare 2 impulsi (da 0 a 1) del segnale di conteggio C
- ♦ Esso può essere realizzato utilizzando un semplice flip-flop di tipo D o di tipo JK edge-triggered



Q è l'uscita di conteggio

Contatori - classificazione

- ◆ I contatori si dividono in due categorie:
 - Contatori sincroni
 - Il segnale di tempificazione di tutti i flip-flop è fornito da un segnale di conteggio comune
 - Contatori a cascata (ripple-counter) o asincroni
 - L'uscita di un flip-flop attiva l'elemento successivo: il segnale di tempificazione di ciascun flip-flop è connesso all'uscita di altri flip-flop posti più a monte

Contatori sincroni

- ◆ I contatori sincroni sono caratterizzati da un segnale di sincronizzazione comune a tutti i flip-flop
- ◆ I contatori sincroni possono essere progettati come ogni altro circuito sequenziale sincrono, partendo cioè dalla loro rappresentazione mediante automa/tabella degli stati
 - Non hanno bisogno in generale di ingressi esterni oltre il segnale di conteggio e l'eventuale segnale di abilitazione (o enable **EN**), e le loro uscite sono prese dalle uscite dei flip-flop che li compongono
 - Le uscite assumono tipicamente lo stesso valore dello stato futuro, uguale al valore dello stato presente aumentato di 1. Spesso i contatori hanno un'uscita detta divisore o riporto **CO**, alta quando si raggiunge il valore massimo di conteggio

Progetto di contatori sincroni – esempio (1/4)

- ◆ La figura riporta un esempio di tabella degli stati per un contatore sincrono a 4 bit senza ingressi e uscite aggiuntivi
 - Come si può osservare nella tabella, ciascun segnale Q_i diventa alto quando tutti i segnali Q_j precedenti sono alti
 - Quando il circuito raggiunge lo stato 1111, passerà, nella successiva transizione, allo stato 0000, iniziando un nuovo ciclo di conteggio

Stato presente				Stato futuro			
Q_3	Q_2	Q_1	Q_0	Q_3	Q_2	Q_1	Q_0
0	0	0	0	0	0	0	1
0	0	0	1	0	0	1	0
0	0	1	0	0	0	1	1
0	0	1	1	0	1	0	0
0	1	0	0	0	1	0	1
0	1	0	1	0	1	1	0
0	1	1	0	0	1	1	1
0	1	1	1	1	0	0	0
1	0	0	0	1	0	0	1
1	0	0	1	1	0	1	0
1	0	1	0	1	0	1	1
1	0	1	1	1	1	0	0
1	1	0	0	1	1	0	1
1	1	0	1	1	1	1	0
1	1	1	0	1	1	1	1
1	1	1	1	0	0	0	0

Progetto di contatori sincroni – esempio (2/4)

Effettuando tutti i passi del progetto per flip-flop di tipo JK si ottengono le seguenti espressioni:

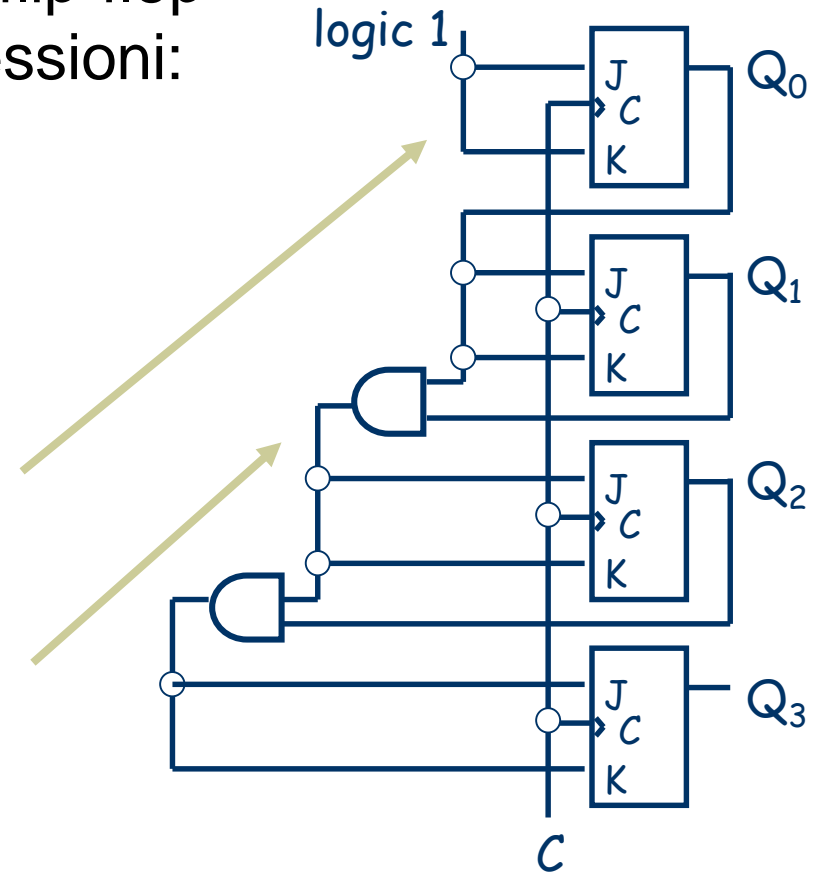
$$J_{Q_0} = K_{Q_0} = 1$$

$$J_{Q_1} = K_{Q_1} = Q_0$$

$$J_{Q_2} = K_{Q_2} = Q_0 Q_1$$

$$J_{Q_3} = K_{Q_3} = Q_0 Q_1 Q_2$$

- Il flip-flop cui corrisponde il bit meno significativo è complementato ad ogni fronte di salita di C
- Ciascuno dei flip-flop successivi è complementato, in corrispondenza del fronte di salita del segnale di conteggio, solo quando tutti i bit delle posizioni precedenti meno significative sono uguali a 1



Progetto di contatori sincroni – esempio (3/4)

Il **segnale di abilitazione EN** potrebbe essere facilmente aggiunto modificando le espressioni dei segnali di posizionamento in questo modo:

$$J_{Q_0} = K_{Q_0} = 1 \cdot EN$$

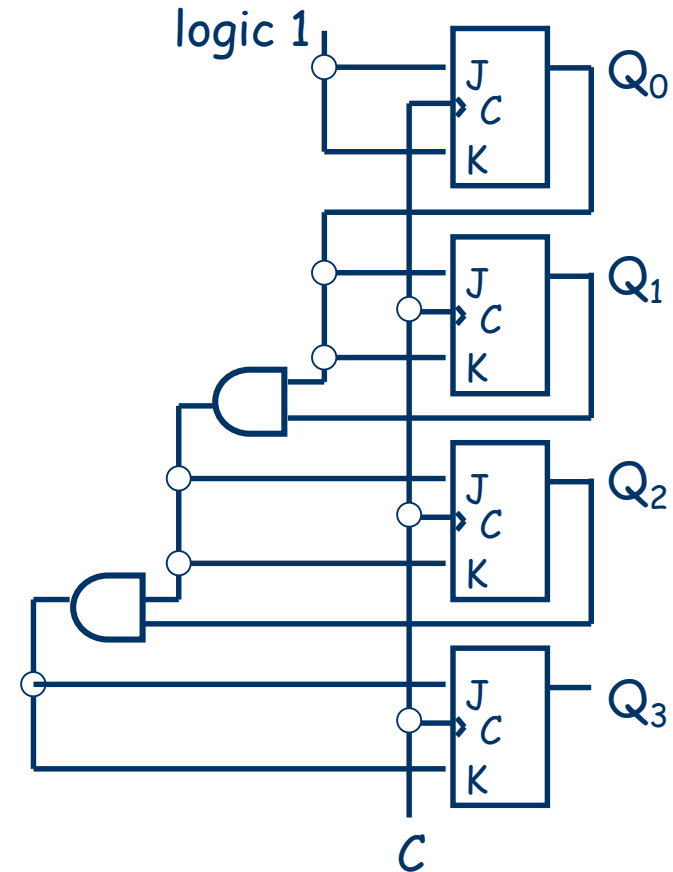
$$J_{Q_1} = K_{Q_1} = Q_0 \cdot EN$$

$$J_{Q_2} = K_{Q_2} = Q_0 \cdot Q_1 \cdot EN$$

$$J_{Q_3} = K_{Q_3} = Q_0 \cdot Q_1 \cdot Q_2 \cdot EN$$

L'**uscita CO** è alta quando le uscite dei flip-flop sono tutte alte e $EN=1$:

$$CO = Q_0 \cdot Q_1 \cdot Q_2 \cdot Q_3 \cdot EN$$



Progetto di contatori sincroni – esempio (4/4)

- ◆ Ripetendo il progetto utilizzando flip-flop di tipo D si ottiene il circuito in figura (completo di EN e CO)
- ◆ Le espressioni dei segnali di posizionamento sono le seguenti:

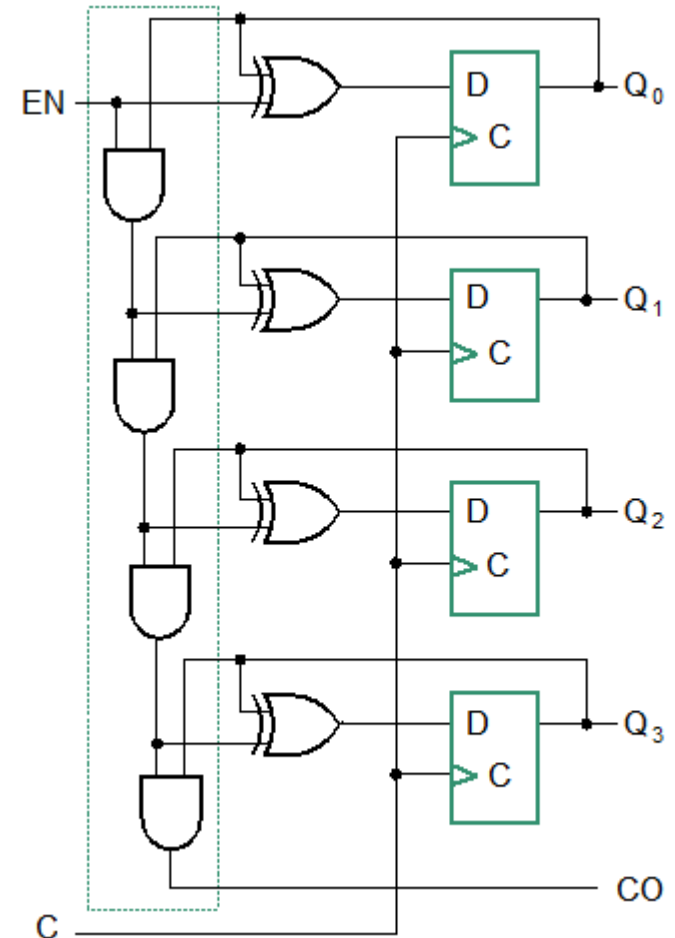
$$D_{Q_0} = Q_0 \text{ XOR } EN$$

$$D_{Q_1} = Q_1 \text{ XOR } (Q_0 \cdot EN)$$

$$D_{Q_2} = Q_2 \text{ XOR } (Q_0 \cdot Q_1 \cdot EN)$$

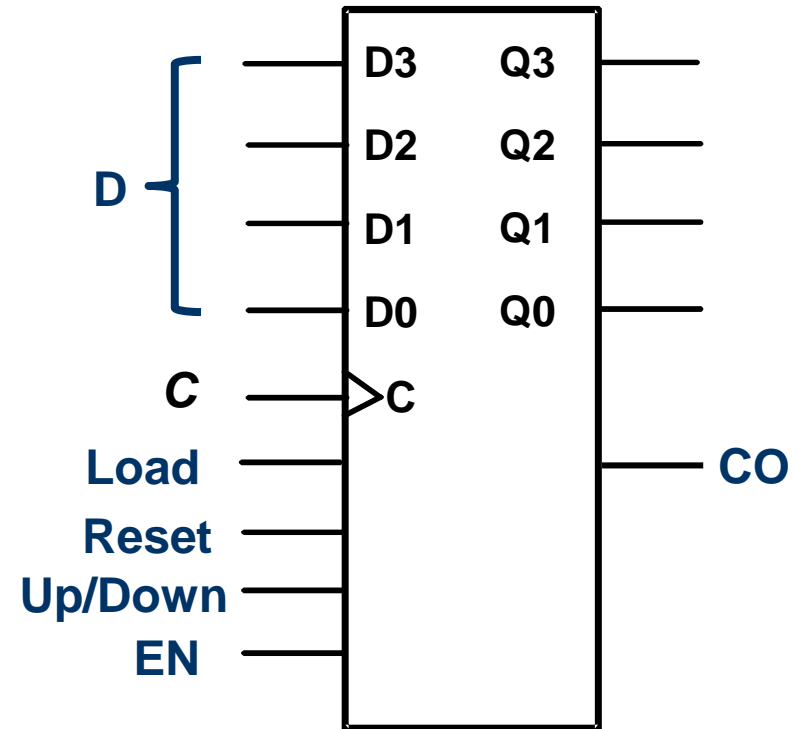
$$D_{Q_3} = Q_2 \text{ XOR } (Q_0 \cdot Q_1 \cdot Q_2 \cdot EN)$$

- ◆ L'uscita CO è alta quando le uscite dei flip-flop sono tutte alte e EN=1



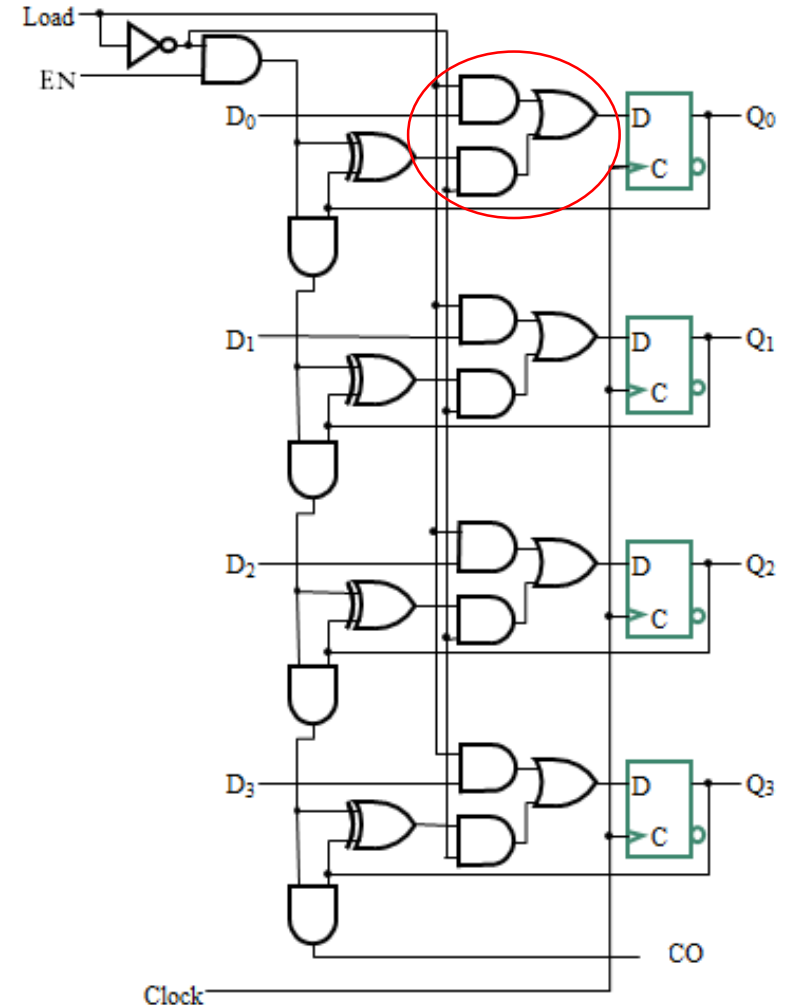
Contatori sincroni – modello generale

- ◆ Nel modello generale, un contatore sincrono può avere un set aggiuntivo di ingressi:
 - ◆ **Enable**: per attivare il contatore
 - ◆ **Load**: per caricare un valore iniziale di conteggio **D**
 - ◆ **Reset**: per resettare il contatore
 - ◆ **Up/Down**: per impostare il verso del conteggio (a crescere o a decrescere)
- ◆ ...e di uscite:
 - ◆ **Uscita divisore/riporto conteggio CO** (alta quando si raggiunge il valore massimo di conteggio)
- ◆ Una rete combinatoria definisce la dipendenza dello stato prossimo dallo stato corrente



Contatori sincroni a caricamento parallelo

- Per realizzare contatori a caricamento parallelo (in cui sia possibile caricare il valore iniziale di conteggio) si utilizza un circuito **multiplexer** in ingresso a ciascun flip-flop
 - Per il circuito in figura, il dato D del flip-flop *i-esimo* dipende da D_i esterno se $Load=1$, oppure dal precedente stato secondo il normale funzionamento del contatore se $Load=0$ e $EN=1$



Progetto di contatori generici

- ◆ Un contatore generico modulo-N ripete una sequenza di N stati, che può seguire il conteggio binario o può essere arbitraria. Esso può essere sempre progettato, come visto in precedenza, come macchina sequenziale partendo dalla sua tabella degli stati
- ◆ **Esempio:** contatore BCD realizzato mediante flip-flop D

Stato presente				Stato futuro				Y
Q ₈	Q ₄	Q ₂	Q ₁	D ₈	D ₄	D ₂	D ₁	
0	0	0	0	0	0	0	1	0
0	0	0	1	0	0	1	0	0
0	0	1	0	0	0	1	1	0
0	0	1	1	0	1	0	0	0
0	1	0	0	0	1	0	1	0
0	1	0	1	0	1	1	0	0
0	1	1	0	0	1	1	1	0
0	1	1	1	1	0	0	0	0
1	0	0	0	1	0	0	1	0
1	0	0	1	0	0	0	0	1

$$D_1 = \overline{Q_1}$$

$$D_2 = Q_2 \oplus Q_1 \overline{Q_8}$$

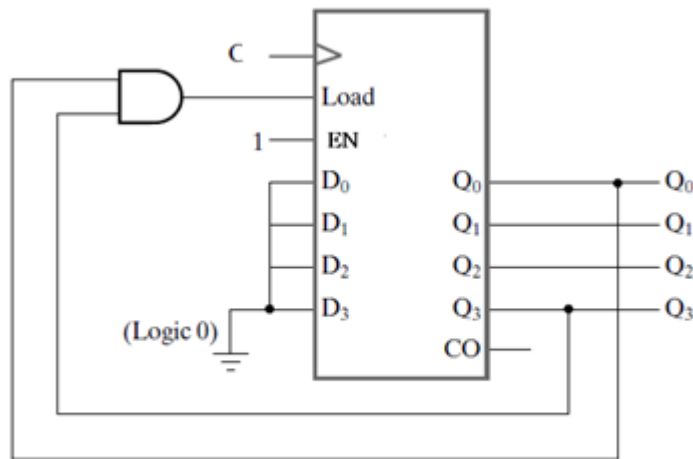
$$D_4 = Q_4 \oplus Q_1 Q_2$$

$$D_8 = Q_8 \oplus (Q_1 Q_8 + Q_1 Q_2 Q_4)$$

$$CO = Q_1 Q_8$$

Progetto di contatori generici – utilizzo di contatori a caricamento parallelo

- ◆ In alternativa, un contatore generico modulo-N può essere realizzato utilizzando un contatore a caricamento parallelo modulo- 2^n tale che $2^n > N$



- Quando l'uscita assume valore 1001, il segnale di Load viene abilitato. Sul prossimo fronte attivo del segnale di conteggio, il contatore verrà caricato con il valore 0000 e ricomincerà il conteggio (Load sincrono)
- Lo stesso comportamento può essere ottenuto abilitando il segnale di Reset quando si raggiunge l'uscita 1001 (considerando che il segnale di Reset ha effetto sul prossimo fronte attivo del segnale di conteggio se è sincrono)

Note:

- nel caso di Load e Reset ad effetto immediato (asincroni), è necessario attendere che l'uscita si porti al valore N e non al valore N-1
- la rete combinatoria utilizzata per ottenere il comportamento visto (*Load attivo con configurazione 1001, cioè $Load = Q_0 \cdot Q_3$*) deve essere opportunamente progettata e minimizzata

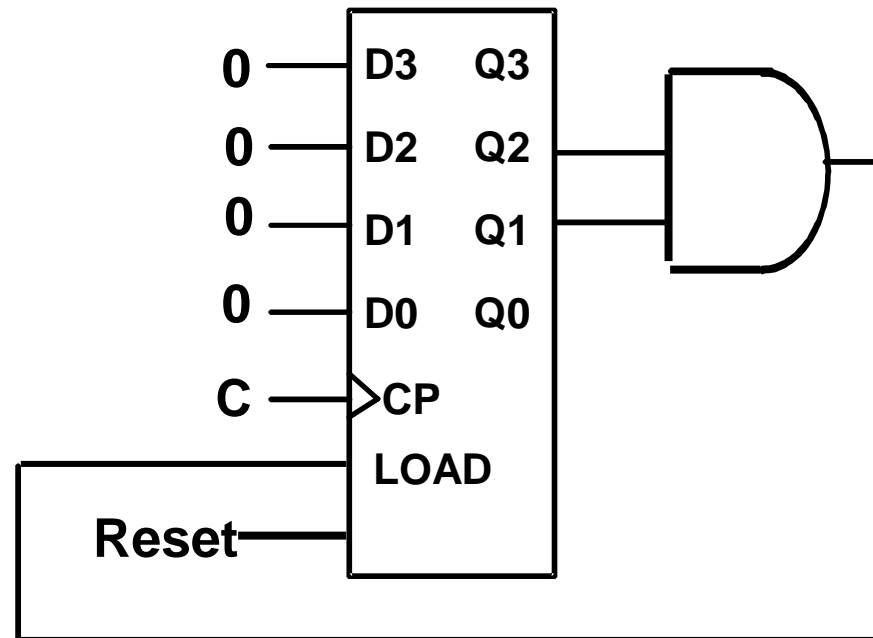
Progetto di contatori generici – utilizzo di contatori a caricamento parallelo

- ◆ Esempio: realizzare un contatore modulo-7 a partire da uno modulo-16

Load	Q_1Q_0			
	00	01	11	10
Q_3Q_2				
00				
01			-	1
11	-	-	-	-
10	-	-	-	-

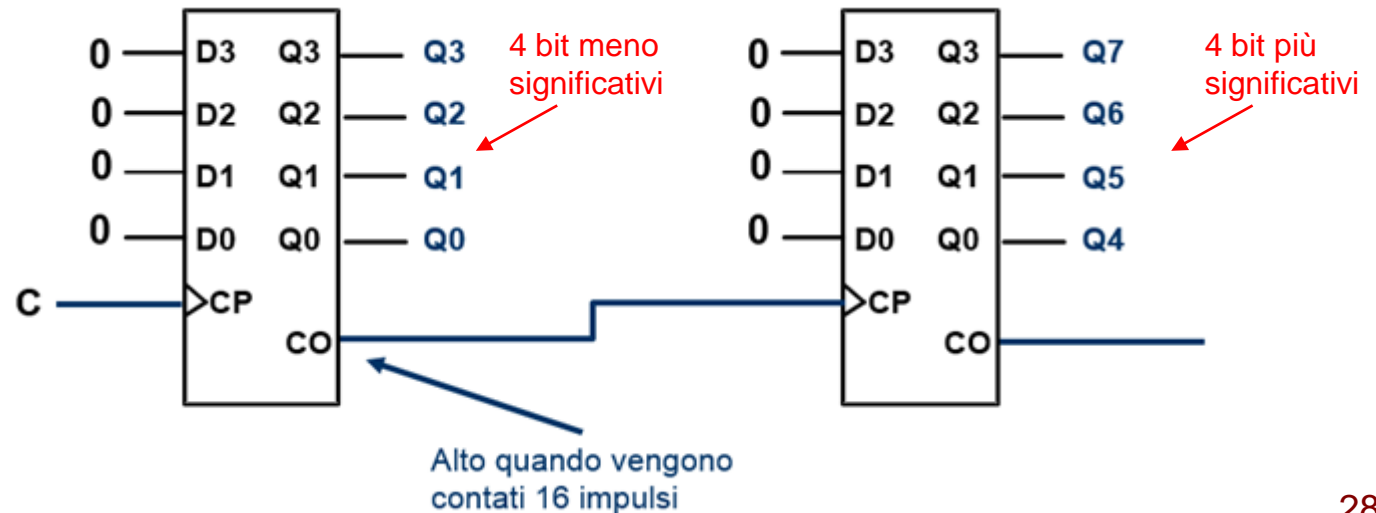
$$\text{Load} = Q_2Q_1$$

Il Load (sincrono) si deve alzare in corrispondenza del valore 0110 (6)



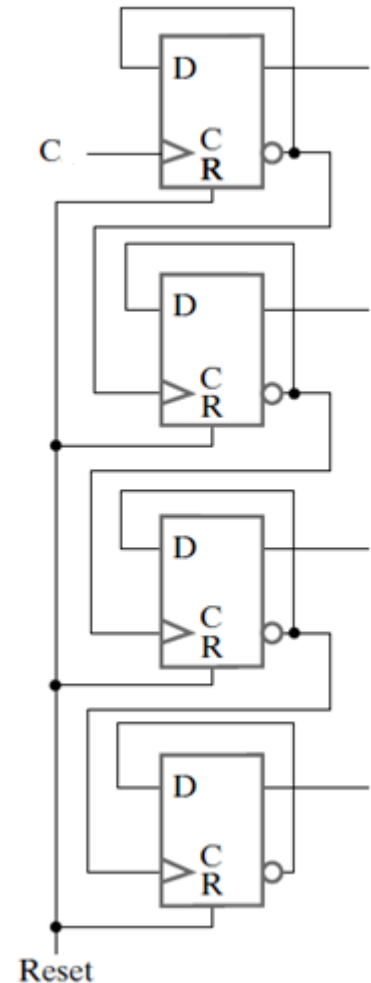
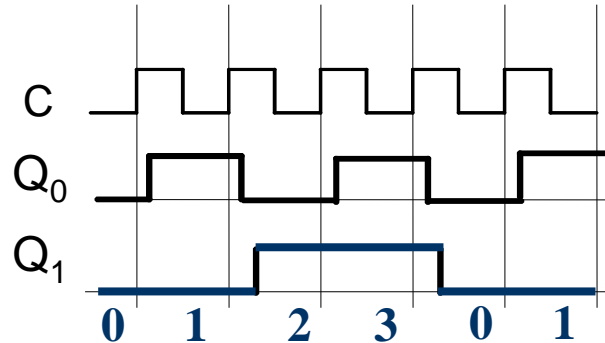
Progetto di contatori per composizione

- ◆ Un contatore generico modulo-N (con N qualsiasi) può essere realizzato componendo opportunamente contatori di modulo inferiore (anche diversi fra loro)
 - Il modulo del contatore composto è dato dal prodotto dei moduli dei contatori componenti
- ◆ Ogni sotto-contatore genera un'uscita divisore CO che, quando diventa 1, attiva il sottocontatore collegato ad esso in cascata
 - Esempio: per realizzare un contatore modulo-256 è possibile usare due contatori modulo-16



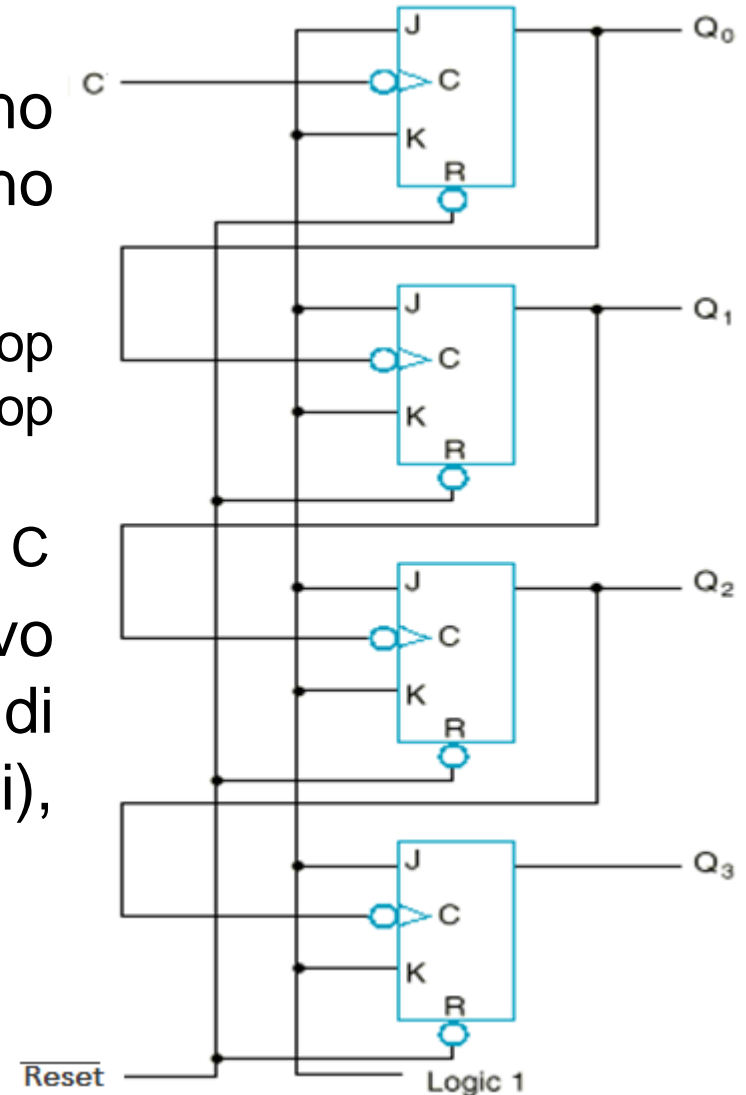
Contatori a cascata (es. flip-flop D)

- ◆ Un contatore **modulo-2ⁿ** può essere realizzato in maniera **asincrona** utilizzando una architettura a *cascata*, in cui il segnale di tempificazione di ciascun flip-flop dipende dall'uscita del flip-flop precedente
 - Nel circuito in figura, il flip-flop *i*-esimo è attivato dall'uscita \overline{Q} del flip-flop (*i*-1)-esimo
 - Un segnale di Reset esterno collegato ai reset di tutti i flip-flop può essere usato per azzerare il valore di conteggio in maniera asincrona



Contatori a cascata (es. flip-flop JK)

- ◆ Nel circuito in figura, i flip-flop sono utilizzati in modalità *toggle* (J e K sono connessi ad un segnale di 1 logico)
 - Il segnale di tempificazione C del flip-flop i -esimo è connesso con l'uscita Q del flip-flop $(i-1)$ -esimo
 - I flip-flop commutano sul fronte di discesa di C
- ◆ Un livello 0 sul segnale di Reset (attivo basso), che è connesso con i segnali di reset dei flip-flop (anche essi attivi bassi), azzerava i registri in modo asincrono



Contatori a cascata: valutazioni

I contatori a cascata:

- ◆ sono di semplice realizzazione
- ◆ vengono attivate solo le parti del circuito interessate alle transizioni
 - adatti a circuiti che richiedono basso consumo di potenza

Ma...

- ◆ particolarmente lenti per un alto numero di bit
- ◆ più critici da dimensionare a causa della dipendenza dei ritardi