

Agendo sui controlli di modo S_0 e S_1 si seleziona uno dei quattro modi seguenti di funzionamento:

- caricamento parallelo ($S_0 = S_1 = H$): in corrispondenza del fronte di salita del clock, i dati presenti agli ingressi $A-D$ vengono caricati nei flip-flop e quindi riportati sulle uscite Q_A e Q_D ; in questa situazione lo scorrimento dei dati è inibito.
- Scorrimento verso destra ($S_0 = H, S_1 = L$): il fronte attivo del clock determina lo scorrimento dei dati da Q_A verso Q_D .
- Scorrimento verso sinistra ($S_0 = L, S_1 = H$): il fronte attivo del clock determina lo scorrimento dei dati da Q_D verso Q_A .
- Inibizione del clock ($S_0 = L, S_1 = L$): il fronte attivo di clock non provoca effetti.

L'integrato presenta due ingressi seriali, uno dei quali (*SHIFT RIGHT SERIAL INPUT*) inserisce i dati nel flip-flop A durante lo scorrimento verso destra, mentre l'altro (*SHIFT LEFT SERIAL INPUT*) inserisce i dati nel flip-flop D durante lo scorrimento verso sinistra.

7.6 I contatori asincroni

Def. I **contatori** (*counters*) sono circuiti sequenziali in grado di contare il numero degli impulsi applicati sull'ingresso di clock, esprimendo sulle uscite il risultato in codice binario.

È definito **modulo** il numero dei possibili **stati** di un contatore, cioè il numero delle combinazioni delle uscite attraverso le quali si svolge il conteggio.

Il conteggio può essere effettuato per valori **crescenti** (*up counters*) o **decrescenti** (*down counters*).

Vengono definiti **contatori asincroni** (*ripple counters*) quelli per cui la commutazione delle uscite non avviene in un unico istante, ma si propaga con un certo ritardo dall'uscita meno significativa a quella più significativa, a causa della struttura del contatore.

I contatori in cui la commutazione delle uscite avviene nello stesso istante, sono detti **sincroni** (*synchronous counters*) (vedi par. 7.7).

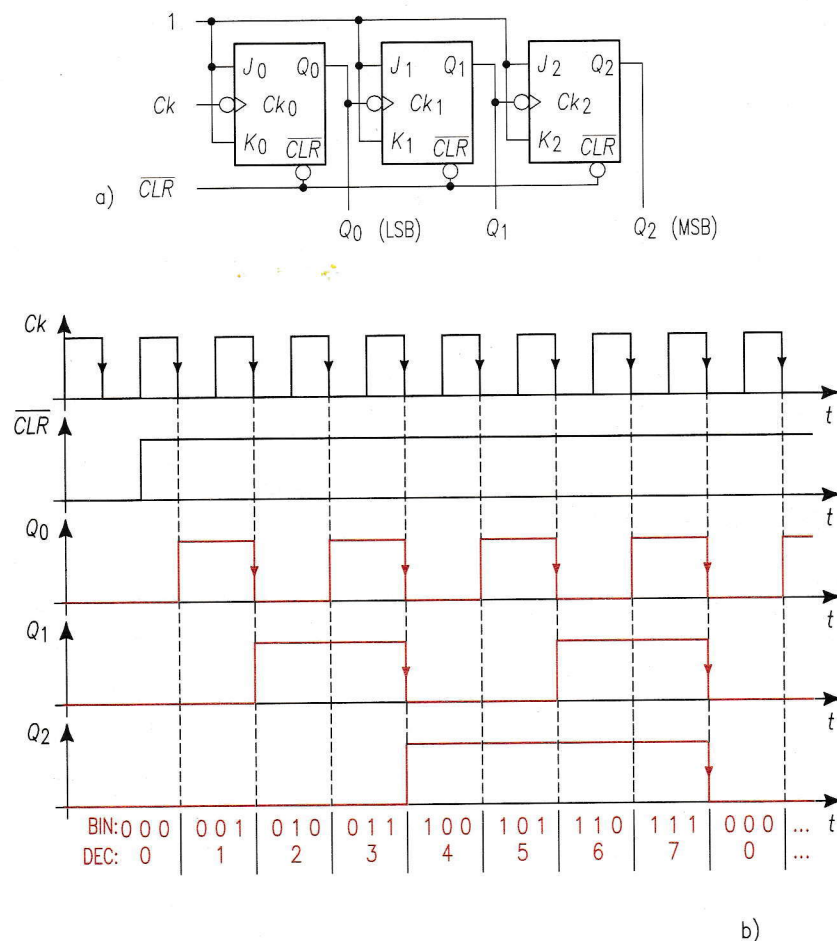
7.6.1 Il contatore asincrono crescente

Un **contatore asincrono crescente** (*up counter*) può essere realizzato collegando alcuni flip-flop JK come indicato nella fig. 7.47a, dove si nota che:

- tutti gli ingressi J e K dei flip-flop sono collegati al valore logico 1 (modo *Toggle*).
- Il segnale di cui si devono contare gli impulsi (CK) è applicato all'ingresso di clock del flip-flop a sinistra.
- L'uscita Q di ogni flip-flop è collegata all'ingresso di clock del flip-flop successivo e quindi ogni flip-flop è sincronizzato da un segnale diverso; da ciò consegue che, nonostante l'utilizzo di bistabili sincroni, la commutazione delle uscite non avviene nello stesso istante, a causa del ritardo con cui si propagano i segnali di clock.
- Gli ingressi di clock sono sensibili ai fronti di discesa del segnale applicato.
- Il risultato del conteggio viene letto sulle uscite Q dei flip-flop.
- Tutti gli ingressi asincroni di *Clear* sono collegati insieme per comandare l'azzeramento asincrono delle uscite del contatore.

Il **funzionamento del contatore asincrono crescente** è evidenziato dal diagramma temporale di fig. 7.47b:

- ogni flip-flop inverte lo stato logico in presenza del fronte di discesa del proprio segnale



Contatore asincrono UP con flip-flop JK: a) schema; b) diagrammi temporali.

Fig. 7.47

- di clock, che proviene dall'uscita del flip-flop alla sua sinistra o dall'ingresso CK;
- considerando come bit meno significativo (LSB) l'uscita Q_0 e come bit più significativo (MSB) l'uscita Q_2 , si ottiene una progressione ciclica di numeri binari da 000_2 a 111_2 , per un totale di otto diversi stati del circuito;
- il circuito in esame è quindi un *contatore asincrono up con modulo 8*.

In generale il modulo di un contatore dipende dal numero n dei flip-flop utilizzati secondo la relazione

$$\text{mod} = 2^n$$

e quindi, con la struttura di fig. 7.47a, è possibile ottenere un numero di stati pari solo alle potenze di 2. Più avanti nel paragrafo si descrive la tecnica per ridurre il modulo di un contatore, ottenendo così un numero qualunque di stati.

È possibile ottenere un identico funzionamento utilizzando il circuito di fig. 7.48, in cui ogni flip-flop D commuta in corrispondenza del fronte di discesa del clock sul proprio ingresso.

Confrontando la struttura di un contatore asincrono con lo schema a blocchi di un circuito sequenziale generico (fig. 7.1), si nota che la rete combinatoria è assente e le uscite del circuito coincidono con le variabili di stato, cioè con le uscite dei flip-flop.

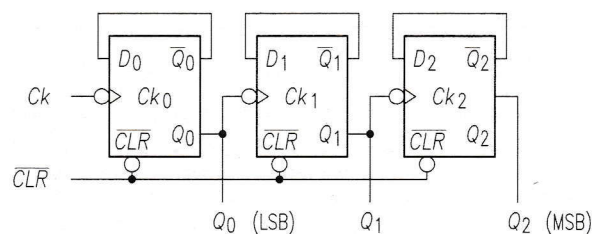


Fig. 7.48 Contatore asincrono UP con flip-flop D.

7.6.2 Il conteggio decrescente

Un **contatore decrescente** (*down counter*) può ottenersi modificando il circuito di fig. 7.47a in uno dei seguenti modi:

- 1) utilizzando dei flip-flop *positive edge triggered* (fig. 7.49a);
- 2) collegando ogni uscita \bar{Q} all'ingresso di clock del flip-flop successivo (fig. 7.49b);
- 3) prelevando le uscite del contatore sulle uscite \bar{Q} dei flip-flop (fig. 7.49c).

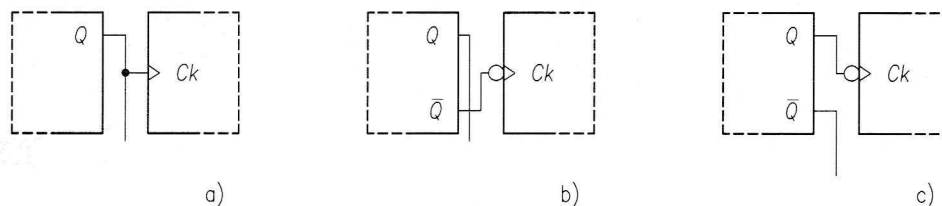


Fig. 7.49 Possibili modifiche per ottenere contatori DOWN a partire dallo schema UP di fig. 7.47.

Ad ogni modifica del circuito, secondo i tre schemi descritti, il contatore inverte il verso del conteggio; si veda a proposito l'esempio 7.12.

ESEMPIO 7.12

Disegnare il diagramma temporale delle uscite del contatore di fig. 7.50a.

Soluzione

Si osservi che il circuito è stato ottenuto da quello di fig. 7.47a attraverso una delle tre modifiche sopra elencate e precisamente si sono utilizzati dei flip-flop sensibili al fronte di salita invece che a quello di discesa. Si ottiene quindi un conteggio decrescente (*down*), come si può verificare dal diagramma temporale di fig. 7.50b.

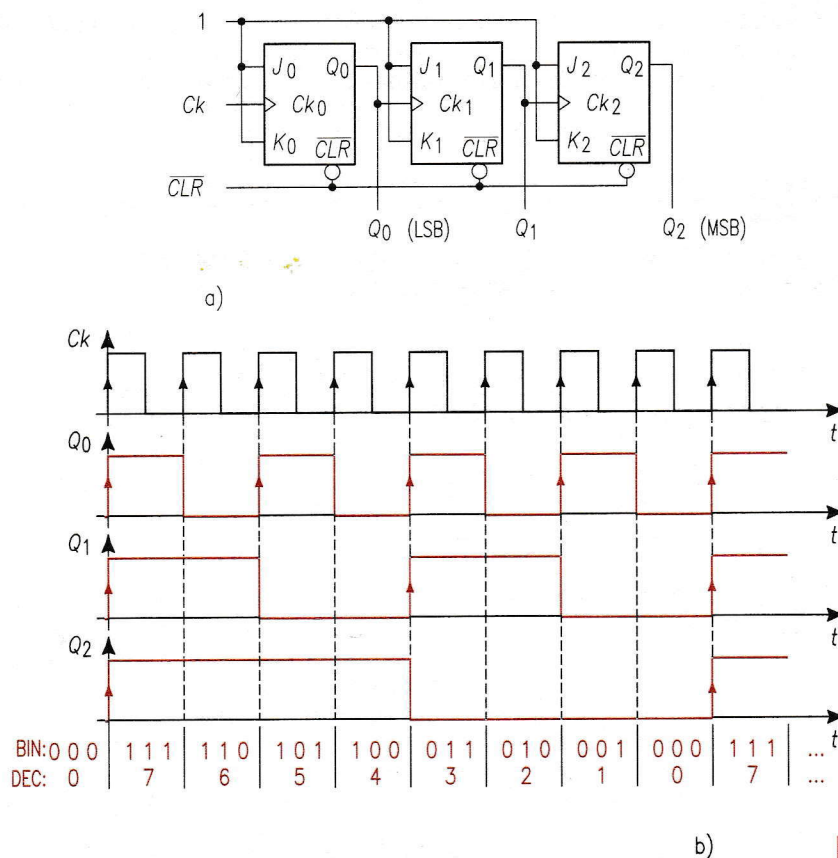


Fig. 7.50

Si veda l'esercitazione di laboratorio n° 25 ob. 1

7.6.3 I contatori asincroni con modulo qualunque

Con i contatori up e down analizzati fino ad ora si possono solamente ottenere valori del modulo pari a potenze di due. Quindi con due flip-flop si avranno contatori modulo quattro, con tre flip-flop modulo otto, con quattro modulo sedici, ecc.

Per ottenere valori del modulo diversi dalle potenze dei due è necessario utilizzare un numero sufficiente di flip-flop e *troncare la sequenza di conteggio* al valore desiderato, utilizzando l'ingresso asincrono *Clear (CLR)* del contatore.



Il **progetto di un contatore asincrono con modulo qualunque** viene eseguito attraverso i seguenti passi:

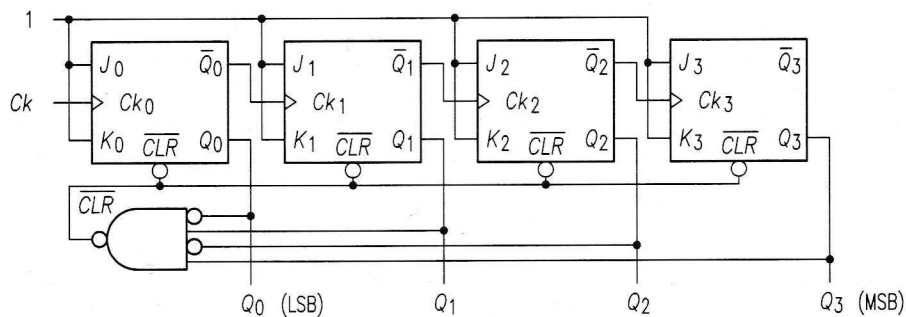
- 1) si determina il numero n dei flip-flop necessari: n è il più piccolo esponente di 2 tale che $2^n \geq \text{mod}$.
- 2) Si definisce lo schema da utilizzare in base al tipo di conteggio desiderato (*up* o *down*).
- 3) S'individua la prima combinazione delle uscite che si vuole eliminare dal ciclo di conteggio.
- 4) Si realizza una rete combinatoria che decodifica questa combinazione e genera il comando di *Clear*, che riporta il contatore nello stato iniziale. In

questo modo la combinazione decodificata permane in uscita solo per il breve tempo necessario a provocare il *reset* del contatore.

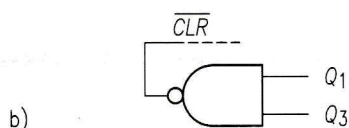
- 5) Si valuta la possibilità di semplificare la rete di decodifica (si veda l'esempio 7.13).

ESEMPIO 7.13

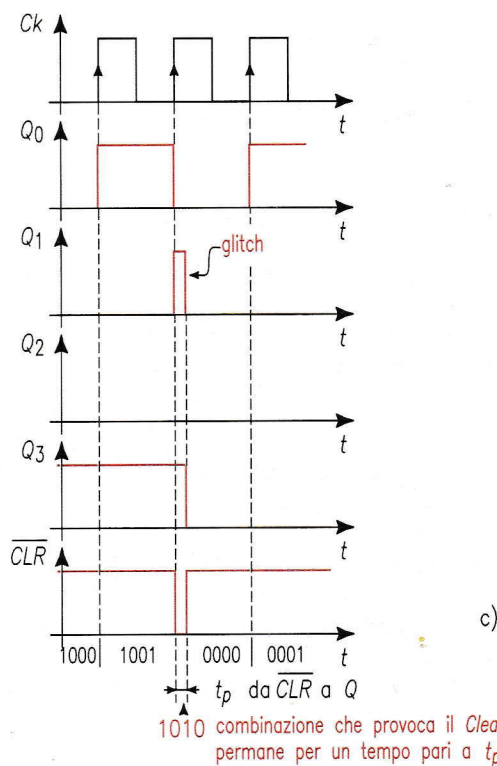
Progettare un contatore asincrono up modulo 10; si dispone di flip-flop JK positive edge triggered, con ingresso asincrono \overline{CLR} attivo BASSO.



a)



b)



c)

Fig. 7.51

Soluzione

- 1) Sono necessari quattro flip-flop; infatti $2^4 = 16 > 10$, mentre $2^3 = 8 < 10$.
- 2) Per ottenere un contatore up con flip-flop positive edge triggered, è necessario applicare, rispetto allo schema di fig. 7.47a, due modifiche tra quelle rappresentate in fig. 7.49; si sceglie di collegare agli ingressi di clock le uscite \overline{Q} (fig. 7.51a).
- 3) Il circuito deve contare da 0000_2 a 1001_2 e quindi la combinazione 1010_2 , da eliminare, deve provocare il reset del contatore.
- 4) La rete combinatoria che decodifica le uscite è costituita dalla porta NAND a quattro ingressi, che comanda a livello BASSO l'ingresso \overline{CLR} , quando la combinazione d'uscita è 1010_2 , resettando il contatore.
- 5) Si noti che le uscite Q_3 , e Q_1 , durante il conteggio, assumono per la prima volta contemporaneamente valore 1 in corrispondenza della combinazione da decodificare; di conseguenza è sufficiente una porta NAND a due ingressi, collegata come in fig. 7.51b.

Si veda l'esercitazione di laboratorio n° 25 ob. 2.



La combinazione che provoca il comando di *Clear* rimane sulle uscite per un periodo pari al tempo di propagazione dei flip-flop tra l'ingresso asincrono \overline{CLR} e l'uscita, generando brevi impulsi (*glitch*) su alcune delle uscite (fig. 7.51c). Questa combinazione indesiderata non provoca problemi in certe applicazioni, come nel caso in cui il valore raggiunto dal contatore debba essere visualizzato su un display, poiché l'occhio umano non riesce a percepirla. Al contrario, se le uscite vengono usate come ingressi di altri circuiti sequenziali, si possono creare malfunzionamenti ed è preferibile optare per l'utilizzo dei contatori sincroni che non presentano questi problemi.

7.6.4 Frequenza massima d'ingresso

Il collegamento in cascata dei segnali di clock dei flip-flop produce un aggiornamento progressivo delle uscite, che si propaga da quella meno a quella più significativa, raggiungendo il ritardo maggiore quando tutte le uscite devono commutare da 1 a 0 (fig. 7.52), o viceversa nel caso di conteggio decrescente.

All'aumentare del numero dei bistabili, e della frequenza del segnale d'ingresso, si può verificare la situazione per cui i bit meno significativi commutano a causa di un nuovo impulso in ingresso, mentre quelli più significativi stanno ancora aggiornandosi a causa dell'impulso precedente.

Nonostante questo fenomeno il contatore funziona regolarmente ed una volta esauriti gli impulsi in ingresso, e trascorso un tempo di assestamento pari alla somma dei tempi di propagazione dei flip-flop, è possibile leggere sulle uscite il risultato del conteggio.

Se si effettua la **lettura delle uscite a fine conteggio**, la massima frequenza applicabile al

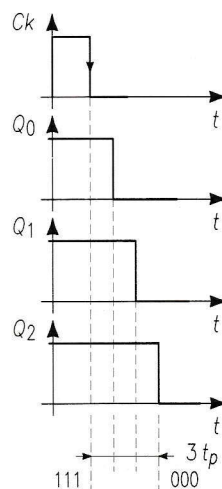


Fig. 7.52

Ogni flip-flop introduce un ritardo t_p rispetto al fronte di clock in ingresso e quindi le uscite di un contatore asincrono commutano in successione.

contatore coincide con la massima frequenza applicabile ai singoli flip-flop, che si può dedurre dai data sheets; in particolare è il primo flip-flop che riceve la frequenza di clock maggiore.

Nel caso in cui sia necessario effettuare la **lettura delle uscite durante il conteggio**, bisogna che in ogni ciclo di clock sia presente un intervallo di tempo in cui le uscite rimangono stabili.

Per calcolare la massima frequenza applicabile ci si riferisce al caso peggiore in cui tutte le n uscite commutano. Il periodo di clock deve risultare maggiore del tempo totale di propagazione, per cui:

$$T_{CK} > n \cdot t_p \quad \text{e quindi} \quad f_{CK} < \frac{1}{n \cdot t_p}$$

ESEMPIO 7.14

Calcolare la massima frequenza applicabile all'ingresso di un contatore asincrono modulo 16, realizzato con quattro flip-flop JK 74LS109. Il risultato dev'essere letto durante il conteggio, garantendo un tempo di stabilità delle uscite pari a 40 ns ogni ciclo di clock.

Soluzione

Dall'analisi dei data sheets dell'integrato si ricava che il tempo di propagazione massimo, dall'ingresso di clock all'uscita Q del flip-flop, vale:

$$t_{p\max} = 30 \text{ ns.}$$

Il massimo tempo di aggiornamento del contatore, calcolato nel caso in cui tutte le uscite commutano, è dato da:

$$4 \cdot t_{p\max} = 120 \text{ ns}$$

Di conseguenza il minimo periodo di clock che garantisce un intervallo di tempo di stabilità pari a 40 ns è

$$T_{CK\min} = 120 + 40 = 160 \text{ ns}$$

a cui corrisponde una frequenza massima d'ingresso pari a

$$f_{\max} = \frac{1}{T_{CK\min}} = 6250 \text{ kHz}$$

7.6.5 I contatori asincroni integrati

I contatori integrati in commercio vengono suddivisi in decadici e binari.

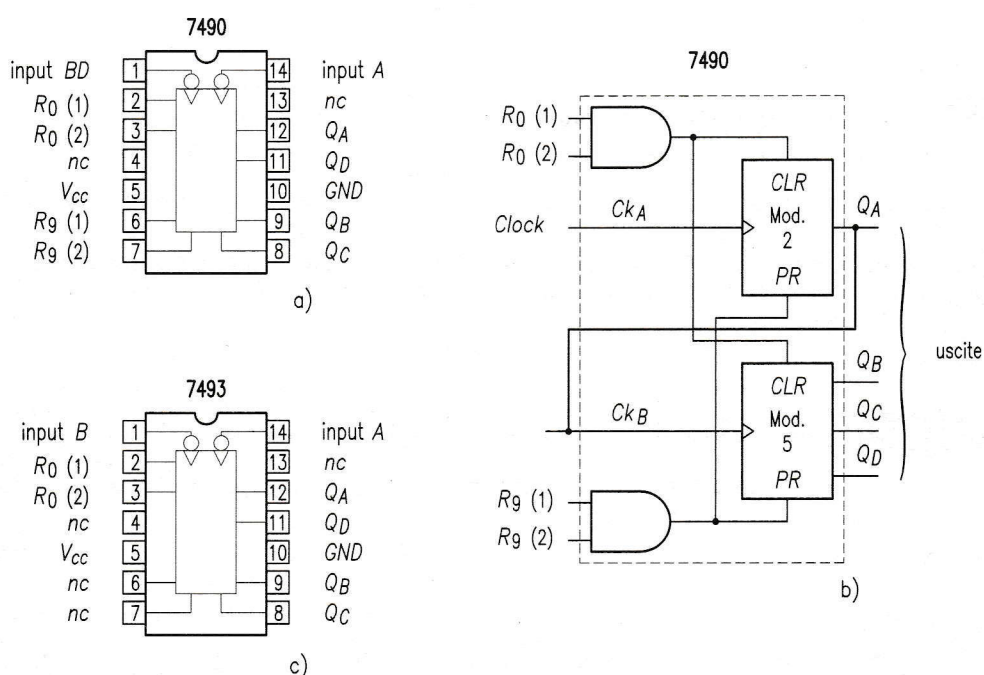
Def. Sono detti **contatori decadici** (*decade counters*) quelli con modulo pari a dieci e che forniscono il risultato del conteggio su quattro uscite, in codice BCD. Questi contatori sono orientati alla visualizzazione dei risultati su display, mediante decoder BCD/7 segmenti. Collegando in cascata diversi contatori decadici è possibile visualizzare il risultato in forma decimale, con unità, decine, centinaia, ecc.

Def. Sono detti **contatori binari** (*binary counters*) quelli che durante il ciclo di conteggio presentano sulle uscite tutte le possibili combinazioni. Un contatore binario con n uscite, realizzato con n flip-flop, avrà un ciclo costituito da 2^n stati.

Uno dei più comuni contatori decadici asincroni integrati è il TTL 7490 (fig. 7.53a), che è costituito da un contatore modulo 2 ed un altro modulo 5; i due stadi possono essere utilizzati separatamente oppure collegati in cascata, come indicato nella fig. 7.53b, ottenendo così un contatore modulo 10. Il 7490 può essere resettato ponendo contemporaneamente a 1 i due ingressi asincroni $R_{0(1)}$ e $R_{0(2)}$, che sono messi in AND internamente, oppure presettato a nove (1001_2) ponendo a 1 gli ingressi $R_{9(1)}$ e $R_{9(2)}$.

L'integrato 7493 (fig. 7.53c) è un contatore binario asincrono, la cui struttura interna è simile a quella del 7490 con due differenze:

- al posto del blocco contatore modulo 5 c'è un contatore modulo 8 che, posto in cascata con quello modulo 2, permette di realizzare un conteggio a 16 stati sulle quattro uscite;
- mancano gli ingressi asincroni di *preset*.



Contatori asincroni integrati: a) pin-out del 7490; b) pin-out del 7493; c) schema logico del 7490. **Fig. 7.53**

È possibile **espandere il modulo di conteggio** collegando in cascata più integrati contatori, come schematizzato nella fig. 7.54. Si noti che il bit di uscita più significativo (Q_D) di ogni integrato viene inviato all'ingresso di clock del contatore successivo; poiché tale ingresso è sensibile al fronte di discesa il contatore viene incrementato quando quello a monte termina un ciclo, commutando Q_D da 1 a 0.

Collegando in cascata due contatori decadici (fig. 7.54a), si ottiene un contatore modulo 100; il risultato del conteggio codificato in BCD con unità e decine, può essere visualizzato su due display, dopo la decodifica BCD/7 segmenti.

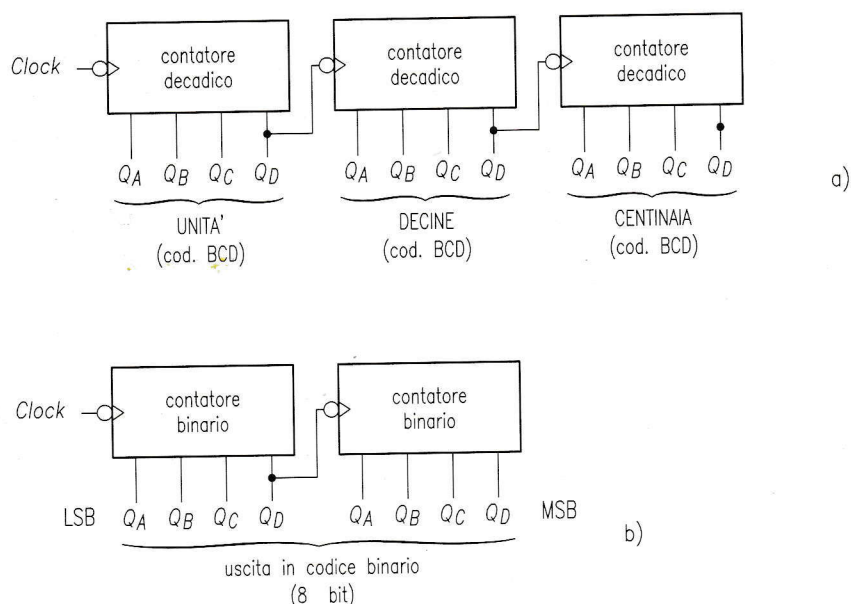


Fig. 7.54 Espansione del modulo di conteggio con contatori a) decadici e b) binari.

Collegando in cascata due contatori binari a 4 bit si ottiene un contatore binario a 8 bit, il cui modulo vale $2^8 = 256$.

Per **ridurre il modulo di conteggio**, come già illustrato in precedenza, si utilizzano gli ingressi asincroni di *Reset* per troncare la sequenza all'ultimo stato desiderato; si veda a proposito l'esempio 7.15.

ESEMPIO 7.15

Realizzare un contatore modulo 12 ed uno modulo 14 utilizzando l'integrato 7493.

Soluzione

Modulo 12

Il ciclo di conteggio modulo 12 va dallo stato 0000_2 (0_{10}) allo stato 1011_2 (11_{10}); di conseguenza la combinazione da decodificare per generare il segnale di *Reset* è la 1100_2 (12_{10}), che viene così esclusa dal ciclo.

Si noti che la combinazione $Q_C = Q_D = 1$, si presenta all'interno del ciclo, solo in occasione del *Reset* del contatore; non è quindi necessario utilizzare un AND a quattro ingressi per riconoscere la combinazione 1100_2 , ma è sufficiente porre in AND i due bit più significativi, sfruttando la porta interna all'integrato, come è evidenziato nella fig. 7.55a.

Modulo 14

Nel secondo caso la combinazione da decodificare vale 1110_2 (14_{10}) ed è perciò necessario porre in AND le tre uscite più significative, per generare il *reset* asincrono; per fare ciò si può utilizzare un AND a due ingressi in combinazione con la porta AND interna all'integrato (fig. 7.55b).

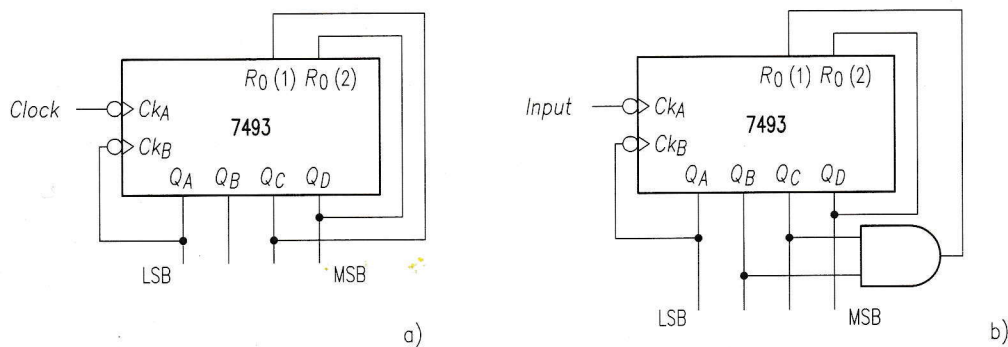


Fig. 7.55

7.6.6 I divisori di frequenza

Analizzando il diagramma temporale del contatore asincrono di fig. 7.47 si nota che, se il segnale d'ingresso è costituito da un'onda quadra con frequenza f_0 , all'uscita Q_0 del primo flip-flop la frequenza risulta dimezzata e vale $f_0/2$.

Questo si verifica perché Q_0 commuta ad ogni fronte di discesa del clock e sono quindi necessari due periodi di clock per completare un periodo di Q_0 .

La frequenza all'uscita Q_1 è ulteriormente dimezzata e vale $f_0/4$; alle uscite successive si otterrà: $f_0/8$, $f_0/16$, ecc.

Si deduce quindi che un contatore binario può essere utilizzato come **divisore di frequenza per potenze di 2** (fig. 7.56).

Per ottenere la **divisione per un intero qualunque** si deve troncare il conteggio al valore desiderato e prelevare il segnale dall'uscita che commuta più lentamente durante il ciclo; si veda a proposito l'esempio 7.16.

In generale, troncando il conteggio, il segnale in uscita non risulterà simmetrico ed avrà quindi un duty-cycle diverso dal 50%, come si può vedere nell'esempio 7.16. Quando è necessaria la simmetria del segnale d'uscita, è sufficiente scomporre il divisore in due blocchi, di cui quello in uscita dev'essere un flip-flop divisore per due; naturalmente questa tecnica è valida solo nei casi di divisione per un numero pari.

Il divisore per 6 dell'esempio 7.16, volendo ottenere un segnale d'uscita simmetrico, deve essere costituito da un divisore per 3 seguito da un divisore per 2.

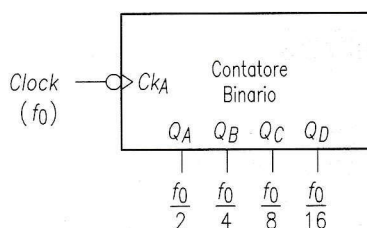


Fig. 7.56

Sulle uscite di un contatore asincrono binario si trovano segnali di frequenza pari a quella di clock.

ESEMPIO 7.16

Realizzare un divisore di frequenza per 6, utilizzando il contatore asincrono integrato 7490.

Soluzione

Si realizza un contatore modulo 6 resettando il ciclo in corrispondenza della combinazione 0110_2 (6_{10}) (fig. 7.57a).

Come si nota dal diagramma temporale di fig. 7.57b, il segnale sull'uscita Q_C completa un periodo ogni sei periodi di clock, quindi la frequenza risulta divisa per sei.

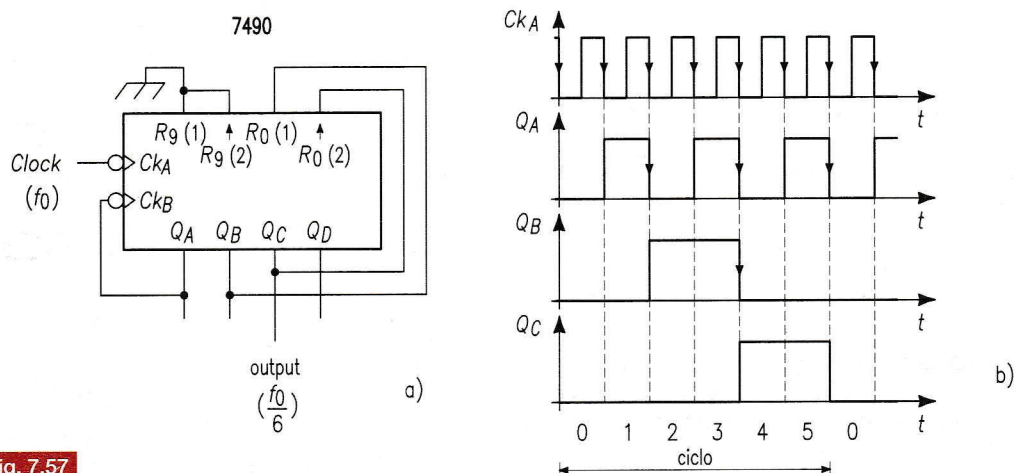


Fig. 7.57

7.7 I contatori sincroni

Def. Si definiscono **contatori sincroni** (*synchronous counters*) quelli in cui tutte le commutazioni delle uscite avvengono nel medesimo istante, in sincronismo con i fronti attivi del segnale d'ingresso. Per ottenere tale sincronismo il segnale d'ingresso deve essere collegato agli ingressi di clock di tutti i flip-flop del contatore.

Lo **schema a blocchi di un contatore sincrono**, rappresentato nella fig. 7.58, evidenzia:

- il blocco **memoria**, contenente n flip-flop dello stesso tipo (D, SR, JK o T), tutti sincronizzati dallo stesso segnale di clock che reca gli impulsi da contare;
- le n uscite del contatore che coincidono con le uscite degli n flip-flop del blocco memoria, dette **variabili di stato**, poiché individuano lo stato del contatore in un dato istante;
- la **rete combinatoria di eccitazione**, che ha il compito di determinare in quale stato (**stato futuro**) dovrà portarsi il contatore in corrispondenza del prossimo colpo di clock, in base allo stato (**stato presente**) in cui si trova in un dato istante. La rete combinatoria riceve in ingresso le variabili di stato, cioè le uscite dei flip-flop, e genera in uscita i valori che saranno letti dagli ingressi dei flip-flop all'arrivo del successivo fronte attivo di clock.
- Rispetto allo schema generale di una rete sequenziale (fig. 7.1), il contatore sincrono non possiede ingressi, a parte quello di clock che fornisce la temporizzazione, ed è quindi costretto ad evolvere secondo una sequenza predefinita di stati, scandita dai fronti attivi del clock.