

CAPITOLO 2

CIRCUITI SEQUENZIALI

I *circuiti sequenziali* sono caratterizzati dal fatto che, in un dato istante, le uscite non dipendono solo dai livelli logici presenti sugli ingressi nello stesso istante ma anche dai livelli logici assunti dalle uscite negli istanti precedenti. In altri termini, il circuito è in grado di conservare *memoria* della sequenza degli stati assunti dagli ingressi e dalle uscite.

Le strutture circuitali che realizzano questa funzione di memoria sono varie ma contengono tutte un *collegamento di reazione* (o *feedback*) che riporta in ingresso lo stato delle uscite.

Anche le modalità con cui le variazioni degli ingressi vengono recepite dai circuiti sono varie: in alcuni circuiti (*latch*) la semplice variazione dello stato degli ingressi determina un nuovo stato delle uscite; in altre strutture (*latch con abilitazione*) il circuito avverte la variazione degli ingressi solo se è attivato un ingresso supplementare di abilitazione; in altri casi ancora (*flip-flop*) è invece presente un segnale di comando o di cadenza, detto *clock*, che controlla la risposta del circuito e definisce esattamente la temporizzazione delle commutazioni delle uscite.

Oltre a tali strutture relativamente semplici, esistono strutture più complesse in grado di memorizzare un numero maggiore di bit: fra esse si annoverano i *registri* e i *contatori*.

2.1 Reti asincrone e sincrone

Il blocco fondamentale che distingue una rete combinatoria da una rete sequenziale è quello denominato *rete di memoria*; ad esso è affidato il compito di «ricordare» la storia dell'evoluzione delle variabili di entrata. Esso connette l'uscita con l'ingresso tramite un cammino di reazione, così come indicato in Fig. 2.1 dove è illustrato lo schema a blocchi di una rete sequenziale.

Se il blocco di memoria contiene dei semplici circuiti di ritardo, la rete sequenziale è detta *asincrona*.

Nelle reti asincrone un cambiamento delle variabili di entrata X produce un cambiamento delle variabili secondarie future Y che sono funzioni dello stato attuale delle X e delle “vecchie” y . Per un certo tempo transitorio, che è appunto il tempo di ritardo, lo stato presente y è diverso da quello futuro

Y. Lo stato logico delle variabili interne e di uscita cambierà continuamente finché si raggiungerà uno «stato stabile» per il quale si avrà:

$$Y_i = y_i \quad \text{per } i = 1, 2, \dots, m . \tag{2.1}$$

Quando si verifica tale uguaglianza la rete sequenziale ha raggiunto lo stato di regime e le uscite sono finalmente stabili. Il tempo transitorio necessario al raggiungimento dello stato stabile dipende fondamentalmente dalla rete di memoria nella quale, peraltro, si considerano concentrati tutti i ritardi delle porte logiche costituenti il sistema.

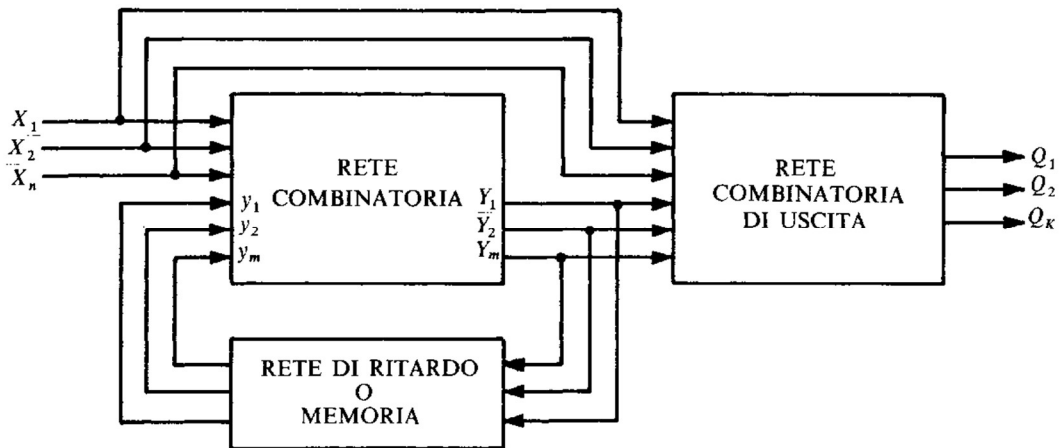


Fig. 2.1 – Schema a blocchi di una rete sequenziale

Si definiscono reti sequenziali *sincrone* quelle nelle quali la variazione dello stato interno di memoria e quindi dell'uscita avviene solo se è presente un particolare *impulso di sincronismo* detto *clock* che nel seguito si indicherà con *CK*.

In Fig. 2.2 si mostra il tipico andamento di un treno di impulsi di sincronismo.

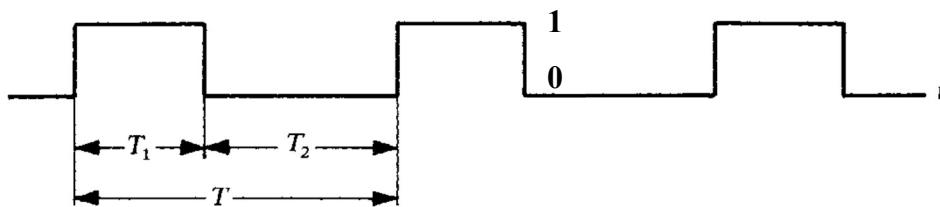


Fig. 2.2 – Rappresentazione temporale di un treno di impulsi di sincronismo

Il tempo T_1 definisce la durata dello stato logico alto 1, mentre T_2 quello basso 0. Il tempo $T = T_1 + T_2$ è detto *periodo* e il suo inverso $f = 1/T$ si chiama *frequenza* o *pulse repetition rate (PRR)*. Si definisce *duty cycle D* il rapporto:

$$D = \frac{T_1}{T}. \quad (2.2)$$

Ad esempio $D = 0,5$ (oppure 50%) sta ad indicare che $T_1 = T_2$; in tal caso il treno di impulsi è detto *onda quadra*.

Il passaggio dal livello basso a quello alto è denominato *fronte di salita* o *fronte positivo*; quello opposto è detto *fronte di discesa* o *fronte negativo*.

I circuiti sequenziali sincroni possono essere comandati o sui *livelli (level triggered)* o sui *fronti (edge triggered)*. Ad esempio se il cambiamento di stato avviene in sincronismo con il fronte di salita il circuito è detto *di tipo PET (Positive Edge Triggered)* se invece è attivato sul fronte di discesa è detto *di tipo NET (Negative Edge Triggered)*. In Fig. 2.3 si mostra il simbolismo d'uso per i segnali di clock.

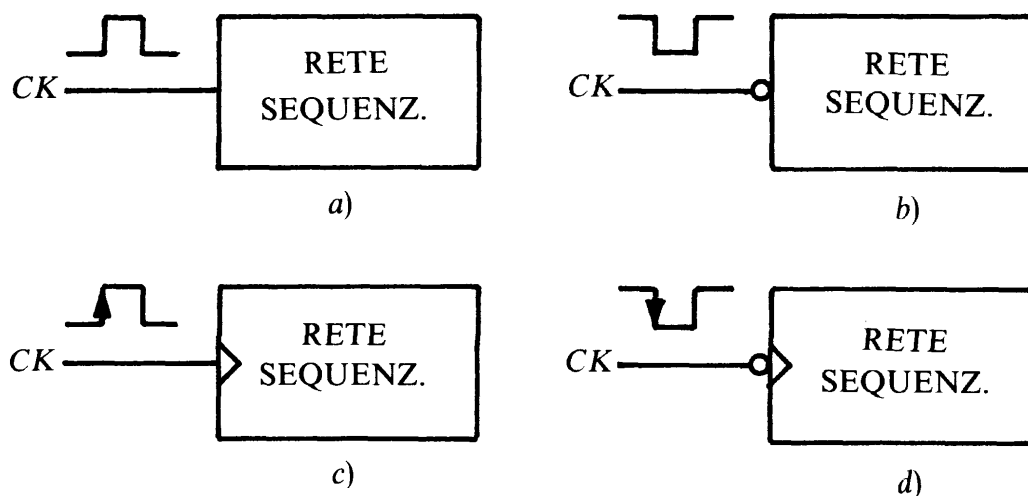


Fig. 2.3 – Simbolismo per il segnale di clock. In a) rete sequenziale attiva sul livello alto di CK, in b) attiva sul livello basso, in c) attiva sul fronte di salita e in d) attiva sul fronte di discesa

2.2 Latch

Il *latch SR* (derivato dal verbo «to latch» che significa «bloccare») è uno dei più semplici circuiti sequenziali asincroni in grado di memorizzare un'informazione binaria a un bit. Esso consta di due ingressi S e R denominati *SET* (letteralmente «porre») e *RESET* (letteralmente «riporre») e di

un'uscita Q . In genere si dispone anche dell'uscita negata \bar{Q} . In Fig. 2.4 è mostrato il simbolo logico del latch SR insieme alla tabella della verità.

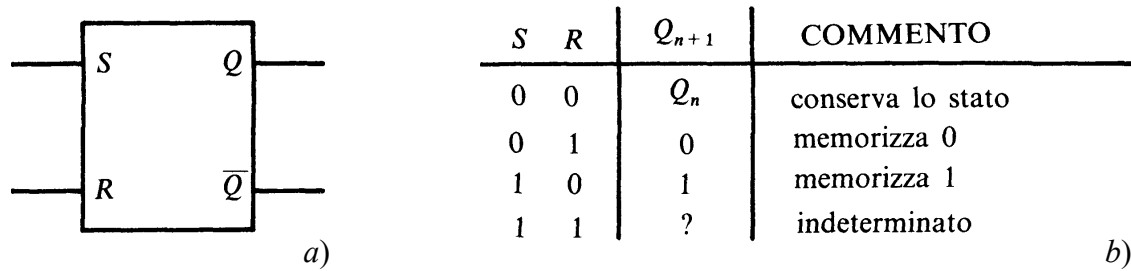


Fig. 2.4 – a) Simbolo logico e b) tabella di verità di un latch SR

La prima riga della tabella della verità indica che la combinazione di entrata $S = R = 0$ lascia *inalterato* lo stato logico della linea d'uscita Q , giacché con Q_n si è indicato lo stato presente dell'uscita e con Q_{n+1} , quello che l'uscita assume dopo aver imposto i valori di entrata S e R . In altre parole, se il latch si trova nello stato $Q_n = 0$ e si applica in entrata $S = R = 0$ l'uscita rimane nello stesso stato cioè: $Q_{n+1} = Q_n = 0$; analogamente se $Q_n = 1$ e $S = R = 0$ lo stato futuro coincide sempre con Q_n per cui: $Q_{n+1} = Q_n = 1$. Ovviamente l'uscita \bar{Q} assume il valore complementare di Q e pertanto non è stata indicata nella tabella della verità.

Dalla seconda riga si deduce che se $S = 0$ e $R = 1$ (operazione di *reset*) l'uscita Q si porterà nello stato $Q_{n+1} = 0$ indipendentemente dal valore precedente 0 o 1 di Q_n .

La terza riga indica che se $S = 1$ e $R = 0$ (operazione di *set*) l'uscita si porterà nello stato $Q_{n+1} = 1$ sia se $Q_n = 0$ che se $Q_n = 1$.

Nell'ultima combinazione $S = R = 1$ è stato associato il simbolo $Q_{n+1} = ?$ per indicare la presenza di uno stato *indeterminato* nell'uscita del latch (portare S e R a 1 equivale a forzare l'uscita Q contemporaneamente a 1 e a 0 che, evidentemente, è un'incongruenza logica e va quindi evitata).

In Fig. 2.5 si mostra lo schema a blocchi circuitale (e la relativa tabella di verità) di un latch realizzato con porte NOR. Analizziamo il suo comportamento:

1) Se $S = R = 0$, lo stato di uscita delle porte NOR dipende dai valori delle uscite stesse. Supponiamo che lo stato iniziale sia $Q_n = 0$ e valutiamo quello futuro Q_{n+1} . Dal circuito si ricava che l'uscita del NOR2 vale

$$\bar{Q}_{n+1} = \overline{S + Q_n} = \overline{0 + 0} = 1. \tag{2.3}$$

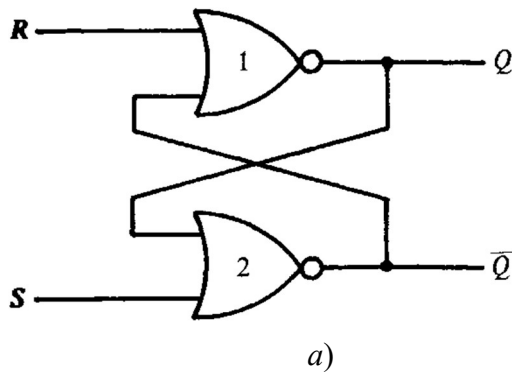
Tale valore è riportato in ingresso al NOR1, per cui

$$Q_{n+1} = \overline{R + \bar{Q}_{n+1}} = \overline{0 + 1} = 0. \tag{2.4}$$

Analogamente se fosse stato $Q_n = 1$:

$$\overline{Q_{n+1}} = \overline{S + Q_n} = \overline{0 + 1} = 0 \tag{2.5}$$

$$Q_{n+1} = \overline{\overline{R + \overline{Q_{n+1}}}} = \overline{0 + 0} = 1. \tag{2.6}$$



S	R	Q_n	Q_{n+1}	\overline{Q}_{n+1}
0	0	0	0	1
0	0	1	1	0
0	1	0	0	1
0	1	1	0	1
1	0	0	1	0
1	0	1	1	0
1	1	0	0	0
1	1	1	0	0

Fig. 2.5 – Latch a porte NOR: a) schema logico e b) tabella di verità

In definitiva si sono verificate le prime due righe della tabella di Fig. 2.5b che corrispondono sinteticamente alla prima riga della tabella di Fig. 2.4b. L'analisi svolta mostra che per $S = R = 0$ lo stato di uscita si conserva ma non è a priori determinato quale esso sia. Infatti, nell'istante in cui si fornisce alimentazione al circuito sia Q che \overline{Q} sono inizialmente nello stato basso per cui entrambe le uscite dei NOR tendono a 1. A causa dei ritardi di propagazione interni alle porte, però, l'uscita di uno dei due NOR va a 1 prima dell'altro NOR in modo casuale definendo così lo stato iniziale del sistema. Se, ad esempio, $\overline{Q} = 1$ prima di Q l'uscita del NOR1 è forzata nello stato basso; questo livello logico, riportandosi in ingresso al NOR2, conferma $\overline{Q} = 1$ e quindi $Q = 0$. Vedremo nel seguito che è possibile imporre lo stato iniziale mediante opportuni ingressi ausiliari detti di *preassegnazione*.

- 2) Se $S = 0$ e $R = 1$, l'uscita del NOR1 è forzata nello stato basso poiché per qualunque valore di Q_n si ha

$$Q_{n+1} = \overline{\overline{Q_n} + R} = \overline{1} = 0. \tag{2.7}$$

Tale stato è riportato in ingresso dal NOR2 la cui uscita diventa

$$\overline{Q}_{n+1} = \overline{S + Q_n} = \overline{0 + 0} = 1. \tag{2.8}$$

In definitiva si ha $Q_{n+1} = 0$ indipendentemente dallo stato iniziale. In tal modo si sono verificate la terza e quarta riga della tabella di Fig. 2.5b.

- 3) Se $S = 1$ e $R = 0$, l'uscita del NOR2 è forzata nello stato basso poiché per qualunque valore di Q_n si ha

$$\overline{Q_{n+1}} = \overline{S + Q_n} = \overline{1} = 0. \quad (2.9)$$

Tale valore è riportato in ingresso al NOR1, per cui

$$Q_{n+1} = \overline{\overline{Q_{n+1}} + R} = \overline{0 + 0} = 1. \quad (2.10)$$

Si è quindi dimostrato che per $S = 1$ e $R = 0$ risulta sempre $Q_{n+1} = 1$, come indicato nella quinta e sesta riga della tabella di Fig. 2.5b.

- 4) Se $S = R = 1$, entrambe le uscite Q e \overline{Q} sono forzate allo stato logico 0 indipendentemente da quello precedente. Le uscite non sono più l'una il complemento dell'altra. Inoltre se si riportano gli ingressi nello stato di riposo $S = R = 0$ può accadere:
- S è riconosciuto basso prima di R per cui in ingresso si ha, anche se per un breve istante, $S = 0$ e $R = 1$, che rappresenta la condizione di *RESET* e quindi: $Q = 0$ e $\overline{Q} = 1$;
 - R è riconosciuto basso prima di S per cui si ha la configurazione transitoria $S = 1$ e $R = 0$, che rappresenta la condizione di *SET* che porta: $Q = 1$ e $\overline{Q} = 0$.

In definitiva la combinazione $S = R = 1$ va evitata poiché non è possibile determinare univocamente lo stato logico dell'uscita quando le entrate tornano nello stato di riposo $S = R = 0$.

Benché il latch possa essere facilmente realizzato con porte logiche, esistono sul mercato circuiti integrati che contengono quattro latch analoghi a quelli descritti. Si possono citare i tipi CMOS 4043 (a porte NOR) e 4044 (realizzato con porte NAND) che presentano uscite three-state, e il tipo TTL 74279 (a porte NAND).

Oltre agli ingressi dati, i circuiti latch possono presentare un ingresso supplementare EN (*enable: abilitazione*) come il circuito mostrato in Fig. 2.6a; tale ingresso controlla il funzionamento del latch secondo la tabella di verità di Fig. 2.6b. Quando $EN = 0$, il latch viene mantenuto nello stato di memoria. In questo caso, il latch è virtualmente isolato dalla sorgente dei dati ed è insensibile ad eventuali variazioni dello stato di S e di R . Quando $EN = 1$, il dispositivo è abilitato e funziona secondo le modalità indicate in precedenza. Il segnale di abilitazione EN viene anche chiamato segnale di *gating* o di *stroke*, per cui il circuito di Fig. 2.6a viene spesso indicato come *gated* o *strobed latch SR* (o più semplicemente, *latch SR con abilitazione*).

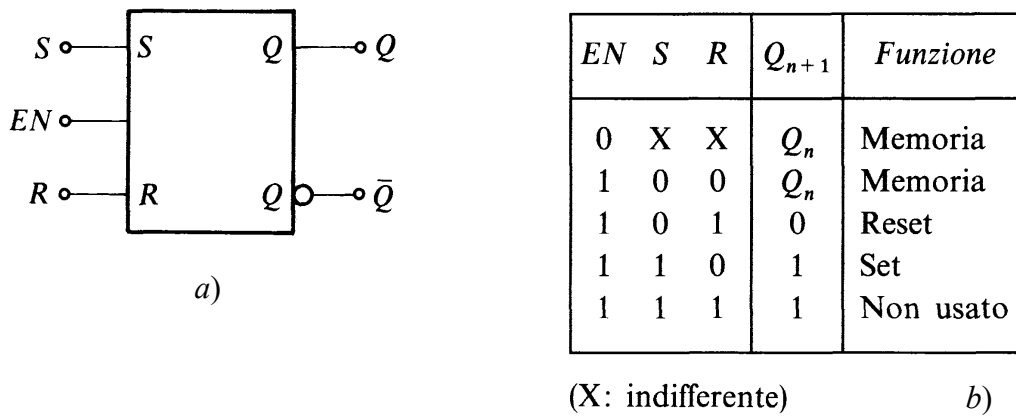


Fig. 2.6 – Latch con abilitazione: a) simbolo circuitale e b) tabella di verità

Un’interessante applicazione dei latch è quella degli *interuttori antirimbazzo (anti-bounce)*. Gli interruttori meccanici presentano l’inconveniente che, durante il loro funzionamento, possono generare degli impulsi non desiderati che potrebbero modificare il comportamento di eventuali reti digitali ad essi collegati. Quando si chiude un interruttore o un deviatore, a causa dell’elasticità della lamina meccanica interna, non si ha un’istantanea e permanente chiusura del contatto. Quest’ultimo subisce un certo numero di oscillazioni prima di assumere definitivamente la posizione di interruttore chiuso. Durante tutta la fase dei rimbalzi meccanici lo stato logico dell’uscita Q può variare tra 0 e 1 un numero di volte del tutto casuale. In teoria, questi impulsi spuri si potrebbero eliminare ponendo tra uscita e massa un condensatore di opportuna capacità (filtro passa-basso) che li cortocircuiti, tuttavia in tal modo si avrebbe una perdita nella velocità di risposta.

Tramite un latch SR, come mostrato in Fig. 2.7, si possono eliminare le oscillazioni in transitorio.

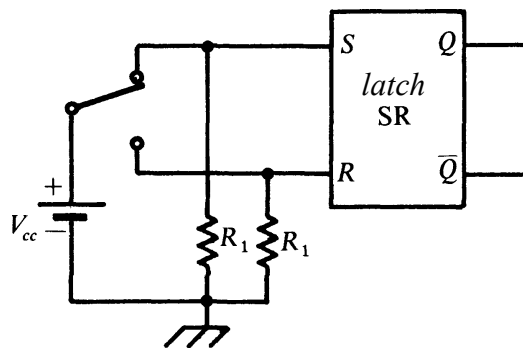


Fig. 2.7 – Circuito antirimbazzo realizzato con un latch SR

Quando il deviatore è nella posizione S risulta $SR = 10$ per cui $Q = 1$ e $\overline{Q} = 0$. Quando il deviatore si porta dalla posizione S alla R , in una prima fase la lamella del deviatore non tocca i punti S e R per cui $SR = 00$ e l'uscita conserva lo stato $Q = 1$. Nell'istante in cui la lamella tocca il punto R si ha: $SR = 01$. L'uscita diventa: $Q = 0$ e $\overline{Q} = 1$; i piccoli rimbalzi meccanici fanno assumere agli ingressi SR i valori $SR = 00$ ed $SR = 01$ che confermano $Q = 0$. In definitiva l'uscita Q ha cambiato il suo stato logico ma senza rimbalzi.

2.3 Flip-flop (o multivibratore bistabile)

I *flip-flop* sono dispositivi caratterizzati dal fatto di potere commutare tra due stati stabili (e per tale motivo essi sono denominati *multivibratori bistabili*); essi sono provvisti di ingressi dati e di un ingresso supplementare a cui viene applicato un segnale di sincronizzazione o di *clock*. La commutazione delle uscite, benché determinata dallo stato degli ingressi, si verifica solo in sincronismo con una ben precisa transizione del segnale di clock; al di fuori di questa transizione le uscite sono insensibili ad eventuali variazioni dello stato degli ingressi. Il segnale di clock, per questa sua funzione di far scattare ossia di innescare le commutazioni del dispositivo, viene anche chiamato *segnale di trigger* o di *comando*.

Considerando le varie modalità con cui il clock interviene, si possono distinguere le due grandi categorie dei flip-flop *edge-triggered* (comandati da un fronte d'onda) e dei flip-flop *pulse-triggered* (comandati da un impulso), altrimenti detti *master-slave*.

Volendo invece tenere conto della funzionalità globale e precisamente delle relazioni fra lo stato degli ingressi dati e lo stato delle uscite, si giunge ad una diversa classificazione che individua i flip-flop di tipo SR , di tipo D , di tipo JK .

2.3.1 Flip-flop SR edge-triggered

I *flip-flop SR* non sono altro che dei *latch SR di tipo sincrónico* (e per tale motivo i latch sono spesso chiamati *flip-flop di tipo asincrono*). Il funzionamento pertanto è analogo a quello dei latch SR , tranne per il fatto che nei flip-flop il funzionamento è definito dal livello logico del segnale di sincronismo, cioè dal clock. Anche il simbolo circuitale è analogo a quello del latch (cfr. Fig. 2.4a), tranne per il fatto che nei flip-flop deve figurare un ulteriore ingresso di clock.

In base a quanto detto, è facile riconoscere nello schema di Fig. 2.8a un flip-flop SR . La latch è abilitata al funzionamento solo se il segnale di clock è al livello alto, cioè $CK = 1$. Quando $CK = 0$ le uscite delle porte pilota portano nello stato di riposo entrambe le uscite della latch, ovvero 00. Si perviene in tal modo facilmente alla tabella della verità di Fig. 2.8b.

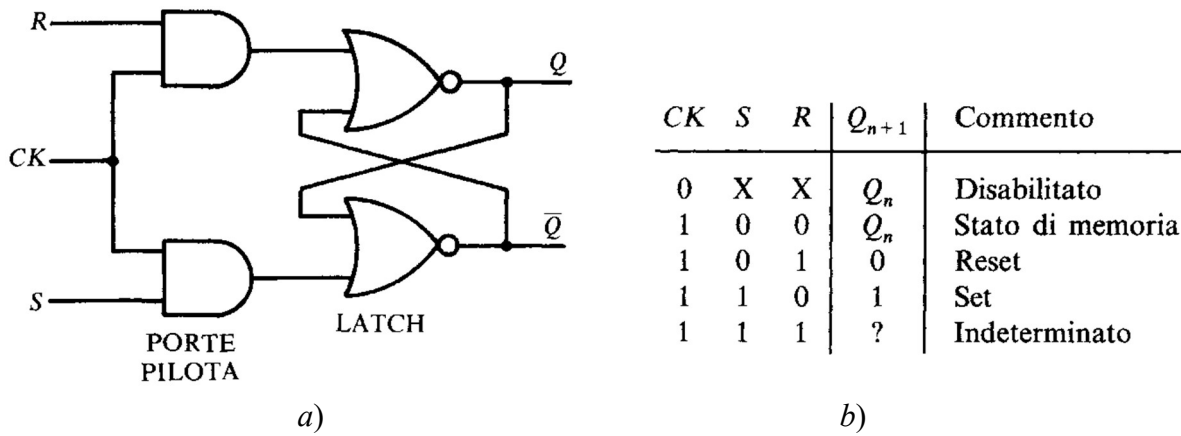


Fig. 2.8 – Flip-flop SR: a) schema logico e b) tabella di verità

Per ottenere un flip-flop attivo sul livello basso del segnale di clock è sufficiente far precedere CK da una porta NOT.

Nel modo descritto in Fig. 2.8, il flip-flop è sensibile al livello logico del segnale di clock. Per rendere un flip-flop sensibile al *fronte di salita* (PET: *positive edge-triggered*) o a quello di *discesa* (NET: *negative edge-triggered*) è necessario che il segnale di clock prima di giungere alle porte pilota assuma una forma impulsiva. Ciò è realizzato mediante alcuni circuiti in grado di fornire un impulso se comandati da un fronte d'onda, che prendono il nome di *multivibratori monostabili*. Un esempio è costituito dall'integrato 74121 che, se connesso opportunamente ad una resistenza e ad una capacità esterne, fornisce un impulso la cui durata dipende dai valori di tali componenti esterni (cfr. Cap. 3).

In Fig. 2.9 e in Fig. 2.10 si riportano due semplici soluzioni circuitali in grado di realizzare un flip-flop di tipo PET.

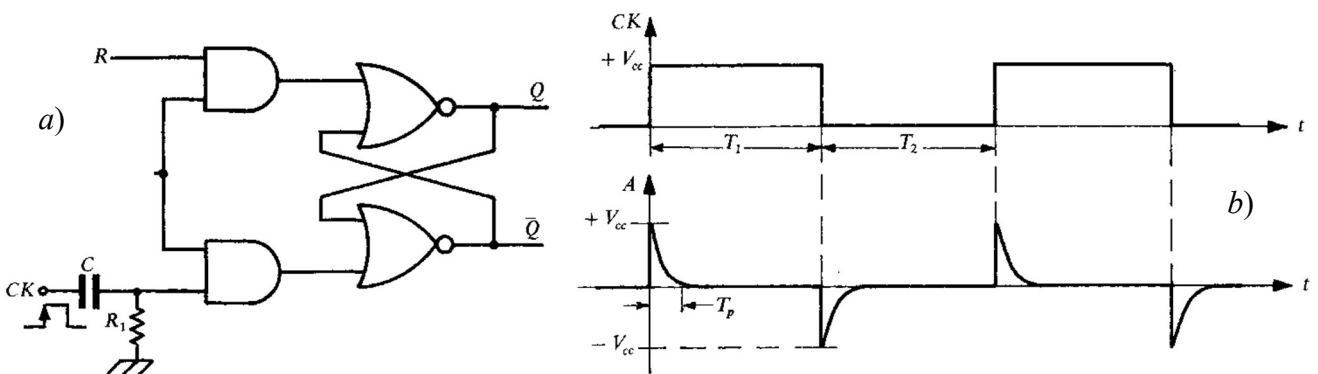


Fig. 2.9 – Flip-flop SR di tipo PET: a) schema circuitale con monostabile ottenuto tramite un circuito R-C; b) risposta temporale del monostabile

Il circuito monostabile di Fig. 2.9a è realizzato tramite la rete passiva R_1-C (circuito derivatore o passa-alto) che rende impulsivo il segnale di clock se la costante di tempo $R_1 \cdot C$ è sufficientemente piccola. La freccia verso l'alto nel segnale di clock indica che il dispositivo è di tipo PET.

In Fig. 2.9b si mostra la tipica risposta temporale di un circuito $R-C$. Si noti che il circuito è sensibile solo agli impulsi positivi del segnale A di Fig. 2.9b mentre gli impulsi negativi vengono assorbiti dai diodi di protezione interni alle porte logiche (si veda più avanti nel Cap. 6). In definitiva, le porte pilota sono abilitate solo per un tempo T_p indipendentemente dal periodo T del segnale di clock e in sincronismo con il fronte di salita di CK .

Un altro circuito di tipo PET è quello di Fig. 2.10a. Il diagramma di tempificazione di Fig. 2.10b mostra che l'impulso di durata T_p viene realizzato sfruttando il *tempo di ritardo di propagazione* della porta NOT (o di più porte dello stesso tipo, se è necessario aumentare il ritardo).

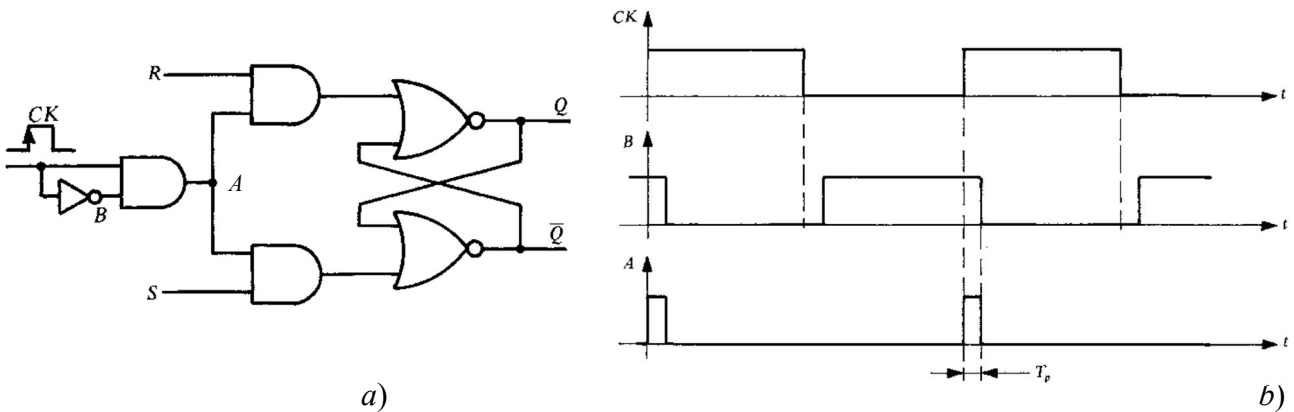


Fig. 2.10 – *Flip-flop SR di tipo PET: a) schema circuitale con monostabile ottenuto sfruttando il ritardo delle porte logiche; b) risposta temporale del monostabile*

2.3.2 *Flip-flop SR master-slave*

Per ottenere un flip-flop SR attivo sul fronte di discesa si utilizzano tecniche analoghe a quelle viste nel paragrafo precedente o, in alternativa, si ricorre allo schema di Fig. 2.11.a, noto come flip-flop SR master-slave (sebbene quest'ultimo – come vedremo – non sia esattamente equivalente ad un flip-flop NET).

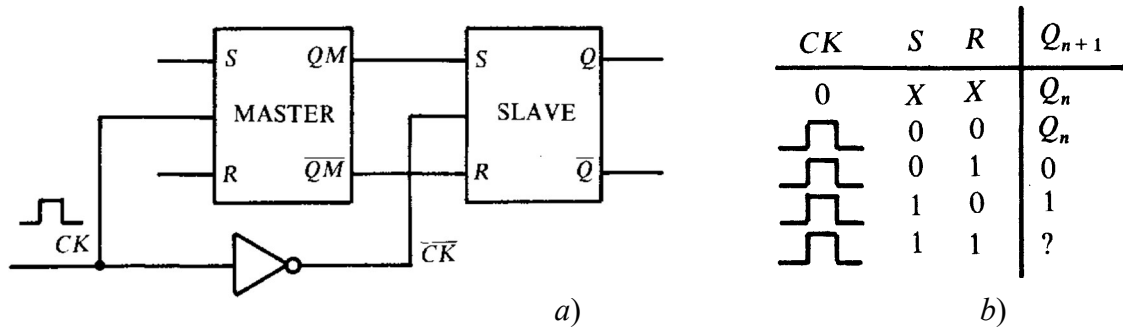


Fig. 2.11 – Flip-flop SR di tipo master-slave: a) schema circuitale e b) tabella di verità

Il circuito è formato da due flip-flop SR posti in cascata. Il primo flip-flop, detto *master*, è attivo sul livello alto di CK, il secondo, detto *slave*, è attivo su quello basso di CK per la presenza della porta NOT. Per $CK = 0$ il master è bloccato, lo slave è abilitato ($\overline{CK} = 1$) e la sua uscita Q dipende dallo stato di memoria Q_M ; in altri termini, per $Q_M = 1$ si ha $Q = 1$ mentre per $Q_M = 0$ si ha $Q = 0$. Pertanto, in tutti i casi, il flip-flop slave *conserva* lo stato del master.

Durante l'intervallo di tempo in cui $CK = 1$ il master è abilitato per cui le uscite Q_M e $\overline{Q_M}$ si portano nello stato imposto dalle entrate SR. Lo slave, però, essendo disabilitato ($\overline{CK} = 0$), non modifica il suo stato di uscita. Nell'istante in cui CK compie la transizione da alto a basso, il master si blocca e sulle uscite Q e \overline{Q} dello slave viene trasferito il contenuto di Q_M e $\overline{Q_M}$.

In Fig. 2.12 si mostrano le tempificazioni per un flip-flop SR master-slave che meglio sintetizzano il funzionamento del dispositivo.

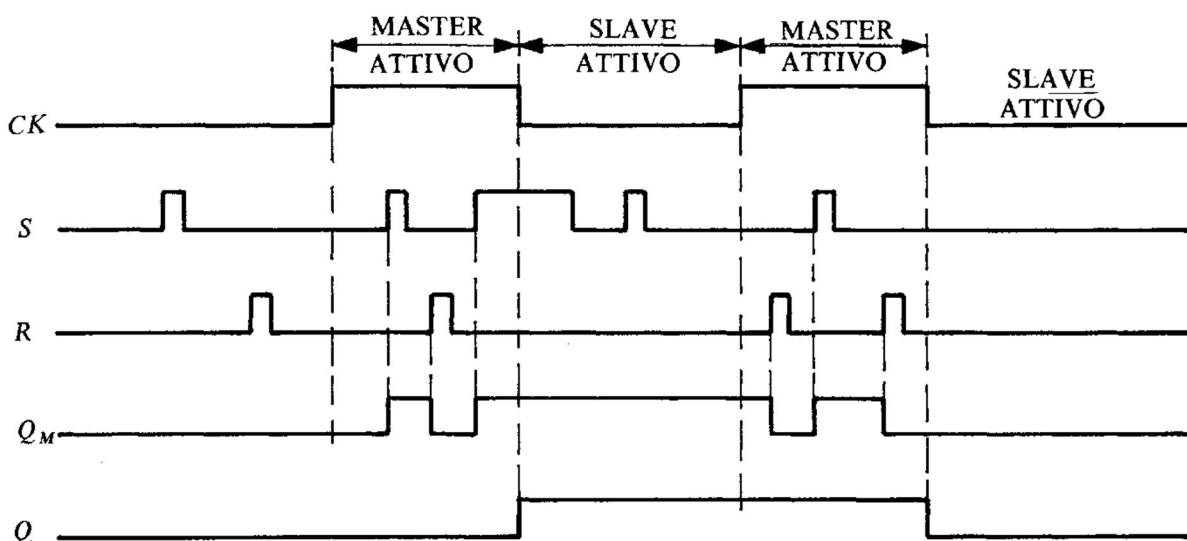


Fig. 2.12 – Diagramma di tempificazione di un flip-flop SR di tipo master-slave

Per tale dispositivo si è supposto che lo stato iniziale sia $Q_M = Q = 0$. Si osservi come il contenuto del master è trasferito in uscita in sincronismo con il *fronte di discesa* del clock.

Attenzione! Un flip-flop master-slave non è esattamente equivalente ad uno di tipo NET (nonostante che il simbolo circuitale sia lo stesso). Infatti, sebbene entrambi siano sensibili al fronte d'onda di discesa del clock, il NET attiva l'uscita dopo aver "visto" un fronte di discesa all'ingresso di sincronismo, mentre per un master-slave è importante che vi sia stata l'evoluzione completa di un impulso di clock (cioè di una semionda). Le informazioni vengono memorizzate nello slave quando il clock è alto e vengono fornite all'uscita quando il clock è basso. Si veda più avanti il diagramma temporale delle uscite di diversi tipi di flip-flop JK.

2.3.3 Flip-flop SR con ingressi di preassegnazione

Nei flip-flop sincronizzati le operazioni di set ($Q = 1$) o di reset ($Q = 0$) si ottengono impostando opportunamente i valori degli ingressi SR e inviando, successivamente, l'impulso di sincronismo CK. In molte applicazioni è utile, invece, impostare lo stato d'uscita del flip-flop in maniera indipendente dal valore di CK, S e R. Ciò si realizza munendo il flip-flop di ulteriori due ingressi, generalmente attivi sul livello basso, denominati PRESET (PR) e CLEAR (CL). Il primo forza l'uscita ad assumere il valore $Q = 1$ mentre il secondo forza l'uscita a $Q = 0$.

In Fig. 2.13a si mostra il simbolo logico di un flip-flop SR attivo sul livello alto di CK e munito di ingressi di preassegnazione PR e CL; la relativa tabella di verità è riportata in Fig. 2.13b. Si noti che, in questo caso, la combinazione $PR = CL = 0$ non è ammessa.

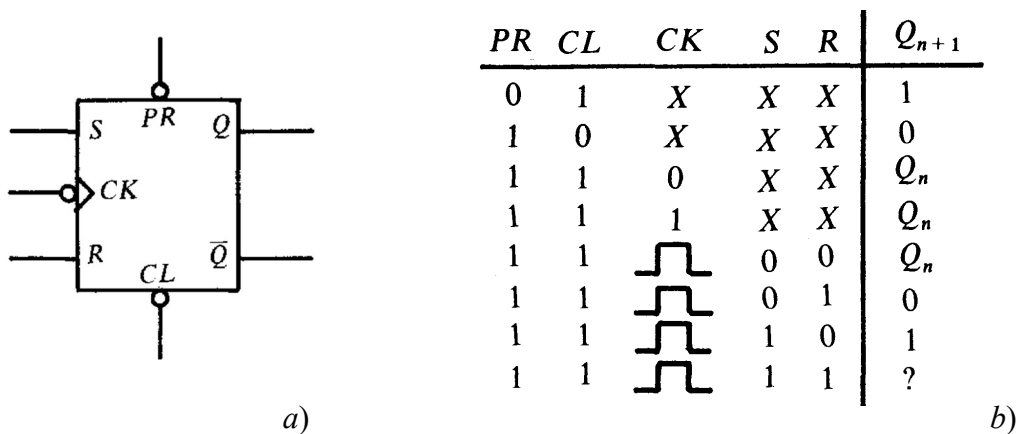


Fig. 2.13 – Flip-flop SR con ingressi di preassegnazione:
 a) simbolo circuitale e b) tabella di verità

2.3.4 *Flip-flop JK*

Il *flip-flop JK* è un dispositivo con due entrate di comando J e K che operano in modo analogo alle entrate SR di un flip-flop *set-reset* con la differenza che se $J = K = 1$, l'uscita Q cambia stato. In tal modo viene eliminata l'indeterminazione che si ha in un SR quando si pone $S = R = 1$.

In Fig. 2.14 si mostrano i simboli circuitali e le tabelle di verità di tre tipi di flip-flop JK : master-slave, PET, NET. Le prime quattro righe della tabella della verità coincidono con quelle di un flip-flop SR avente $S = J$ e $R = K$; l'ultima riga della tabella indica che per $J = K = 1$ l'uscita cambia stato rispetto a quello che aveva in precedenza, vale a dire $Q_{n+1} = \overline{Q_n}$.

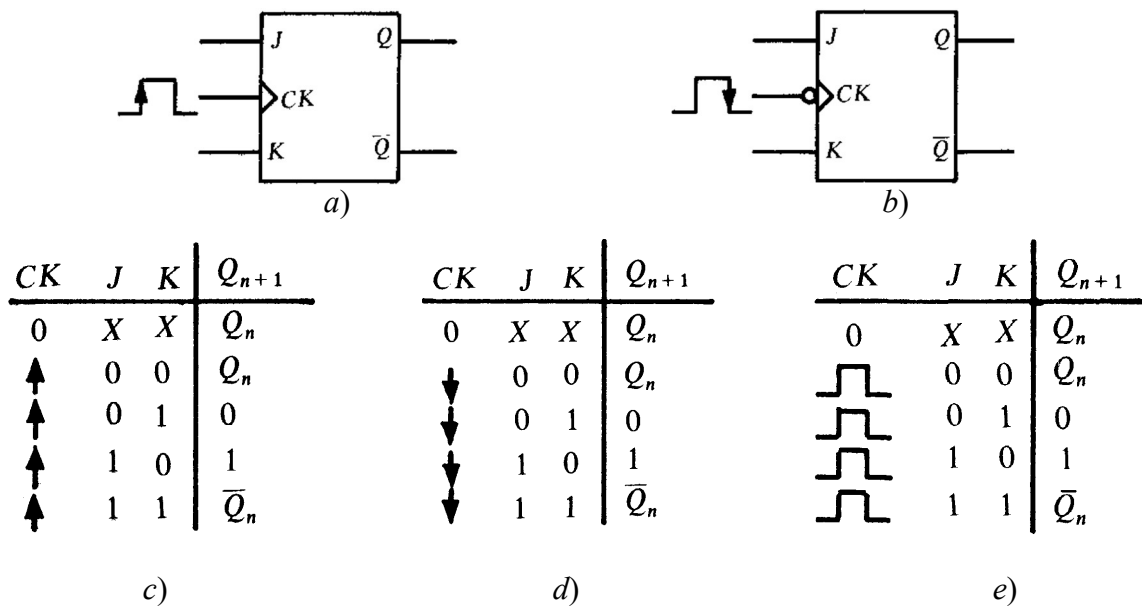


Fig. 2.14 – *Flip-flop JK: simbolo circuitale a) tipo PET, b) tipo NET o master-slave; tabella di verità: c) tipo PET, d) tipo NET, e) tipo master-slave*

La Fig. 2.15 mostra le forme d'onda d'uscita di tre flip-flop rispettivamente master-slave, PET e NET nell'ipotesi che lo stato iniziale sia $Q = 0$. L'analisi del diagramma di Fig. 2.15 evidenzia il diverso modo di operare dei tre tipi di flip-flop.

Anche per i flip-flop JK si possono implementare le reti di preassegnazione di *PRESET* e *CLEAR* simili e quelle esaminate per il flip-flop SR .

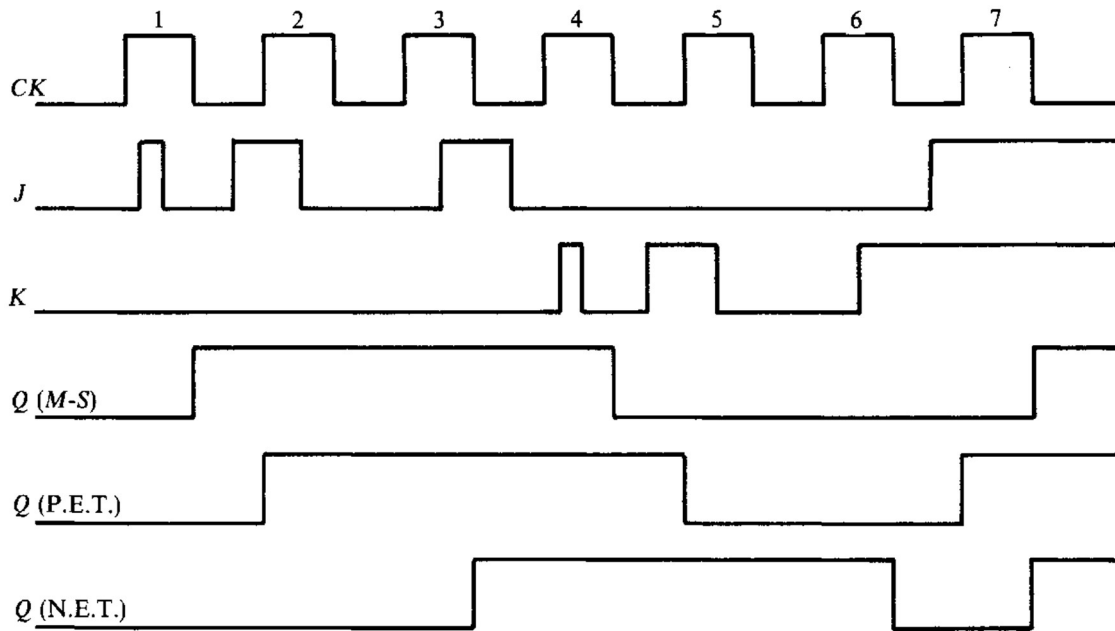


Fig. 2.15 – Forme d'onda d'uscita di flip-flop JK: master-slave, PET e NET

2.3.5 Flip-flop D

Il *flip-flop D* è un tipo di flip-flop avente una sola entrata denominata *D* (*Dato*) e due uscite complementari Q e \bar{Q} . L'uscita Q assume lo stesso stato logico dell'ingresso D dopo che si è applicato il segnale di sincronismo CK . In Fig. 2.16 si mostra il simbolo logico (Fig. 2.16a) e la tabella della verità (Fig. 2.16b) nella quale si prescinde dall'azione del clock.

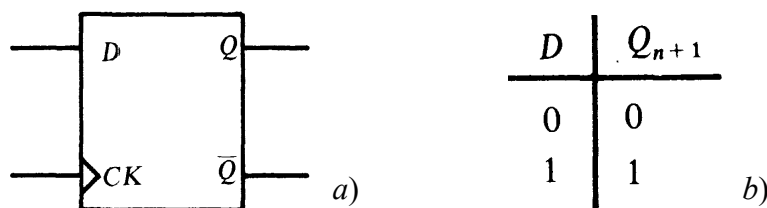


Fig. 2.16 – Flip-flop D: a) simbolo circuitale e b) tabella di verità (clock sottinteso)

I flip-flop *D* pertanto si comportano da *celle di memoria*: essi memorizzano il dato presente all'ingresso D non appena il clock risulta attivo. In commercio sono disponibili con diversi schemi logici atti a rendere il dispositivo attivo sul fronte di salita del clock (PET) o su quello di discesa (NET) o con struttura master-slave. Anche per i flip-flop *D* si possono implementare le reti di preassegnazione di *PRESET* e *CLEAR* simili e quelle esaminate per il flip-flop SR.

2.3.6 Flip-flop T (Toggle)

Il *flip-flop toggle*, o semplicemente *T*, è un flip-flop con un ingresso *T* e due uscite *Q* e \bar{Q} . Se $T = 1$, le uscite cambiano stato ad ogni impulso di clock mentre se $T = 0$, le uscite conservano lo stato logico precedente.

Di seguito si riporta in Fig. 2.17a il simbolo logico di un flip-flop *T* e in Fig. 2.17b la tabella della verità in cui si prescinde dall'azione del clock.

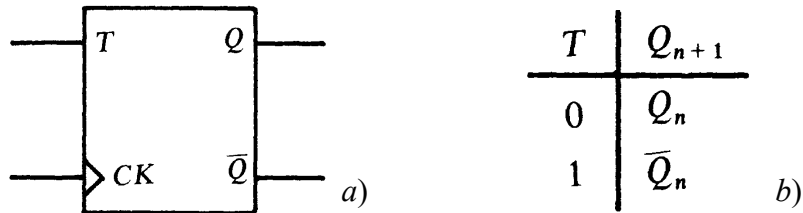


Fig. 2.17 – *Flip-flop T: a) simbolo circuitale e b) tabella di verità (clock sottinteso)*

Un flip-flop toggle si può realizzare mediante un flip-flop *JK* collegando insieme *J* con *K*. In Fig. 2.18 si mostra la soluzione circuitale insieme alla tabella della verità. Da tale tabella si deduce che un flip-flop *T* si può considerare un caso particolare di *JK* in cui si sono utilizzate solo la seconda e la quinta riga delle tabella della verità di Fig. 2.14 (relativa ad un flip-flop *JK*).

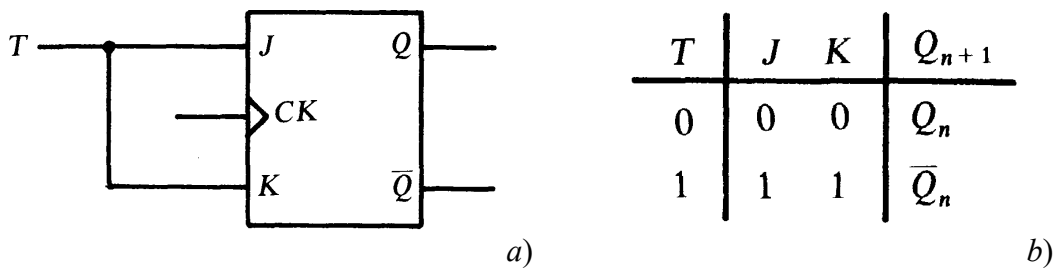


Fig. 2.18 – *a) Flip-flop T ricavato da un JK ponendo $J = K = T$ e b) relativa tabella di verità (clock sottinteso)*

I flip-flop di tipo toggle non sono disponibili sotto forma di circuiti integrati, data la semplicità con la quale possono essere ottenuti a partire da un tipo *JK*.

2.3.7 Flip-flop commerciali

In commercio sono disponibili numerosi circuiti integrati, contenenti tipicamente uno, due o quattro flip-flop attivi sul livello o sulle transizioni del segnale di clock spesso forniti di ingressi asincroni di *PRESET* e *CLEAR* o semplicemente del solo *CLEAR*.

Relativamente alla famiglia TTL citiamo ad esempio i flip-flop *D* 7474 e 7475. Il primo è di tipo PET e contiene due flip-flop indipendenti muniti di *PRESET* e *CLEAR*. Il 7475, invece, contiene quattro flip-flop attivi sul livello alto della linea di abilitazione *E*. Esempi di flip-flop *JK* TTL sono i master-slave 7473 e 7476. Il 7473 è munito solo dell'ingresso asincrono di *CLEAR*, mentre il 7476 è fornito anche di *PRESET*.

Le caratteristiche elettriche generali dei flip-flop TTL dipendono dalla serie di appartenenza e sono riportate in modo dettagliato sui manuali tecnici. In particolare, per la serie standard, la massima frequenza del segnale di clock è tipicamente intorno ai 30 MHz mentre la potenza massima dissipata per ogni flip-flop è di circa 50 mW.

Per quanto concerne i flip-flop della serie CMOS, oltre a quelli della serie 74C, compatibili nella piedinatura con i corrispondenti della serie TTL, esistono numerosi tipi delle serie 4000. Solo a titolo di esempio, citiamo il 40174, che contiene sei flip-flop *D* di tipo PET muniti di clear e tutti comandati dallo stesso segnale di clock, e il 4042, contenente quattro flip-flop *D* attivi sul livello del segnale di clock comune a tutti i flip-flop. In particolare se il segnale denominato «polarity» è 0, i dati di ingresso sono trasferiti in uscita sul livello basso del clock, viceversa se polarity è 1 il trasferimento avviene sul livello alto del clock.

Nelle Fig. 2.19 e Fig. 2.20, infine, si riportano i pin-out di alcuni flip-flop della famiglia TTL e CMOS.

2.4 Registri a scorrimento (o *shift register*)

I *registri* sono dispositivi in grado di memorizzare temporaneamente un certo numero di bit. Poiché, come si è detto, latch e flip-flop consentono di memorizzare un bit, i registri possono essere realizzati disponendo secondo configurazioni adeguate più latch o più flip-flop, che vengono a costituire le *celle* o gli elementi del registro. Essi vengono utilizzati in una vasta gamma di applicazioni, in genere quando occorre mantenere in memoria dei dati per un certo tempo in attesa che altri dispositivi abbiano completato la loro elaborazione. Esempi sono la memorizzazione dei dati di uscita di codificatori, dei dati di ingresso di sistemi di visualizzazione, dei codici di ingresso di comparatori.

I *registri a scorrimento*, o *shift register*, costituiscono un particolare sottoinsieme dell'insieme dei registri. Essi sono formati da una catena di celle di memoria ad un bit interconnesse fra di loro; ad ogni impulso di clock essi consentono lo *scorrimento dei bit* da una cella alla cella immediatamente adiacente. Come si vedrà, l'uso di shift register rende più flessibili e agevoli le procedure di trasferimento dei dati ai registri (caricamento o scrittura) e dai registri (prelevamento o lettura) e consente l'implementazione di speciali funzioni di uso frequente.

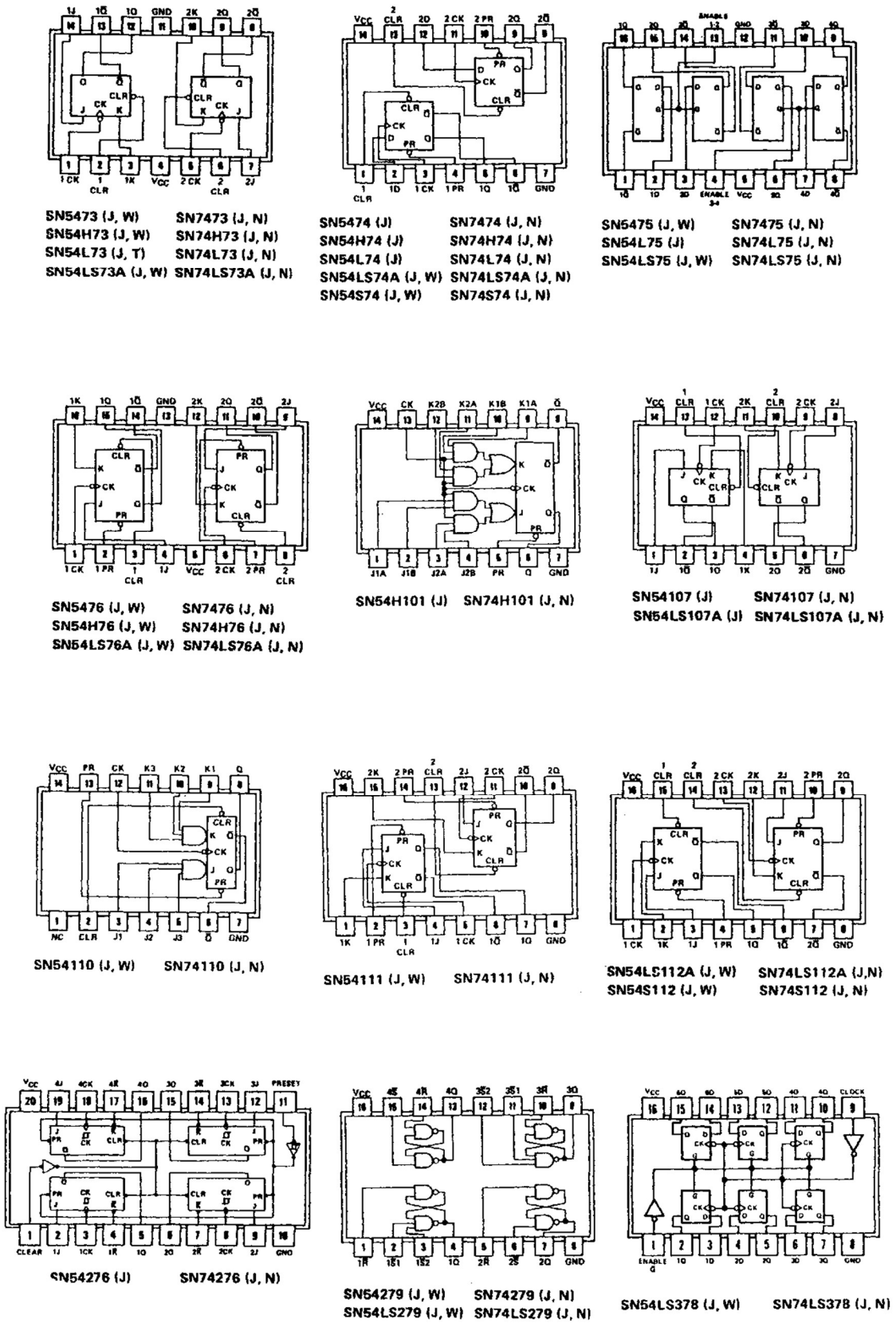
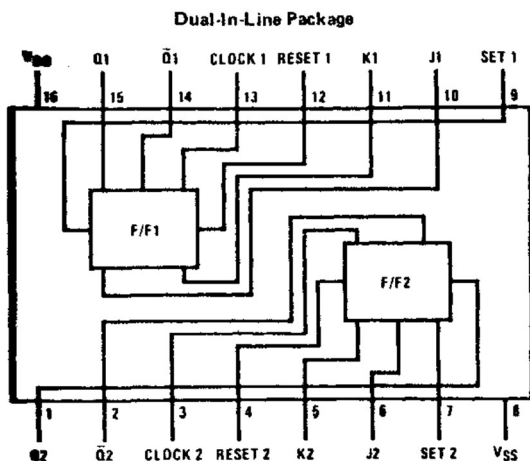


Fig. 2.19 – Pin-out di alcuni flip-flop TTL

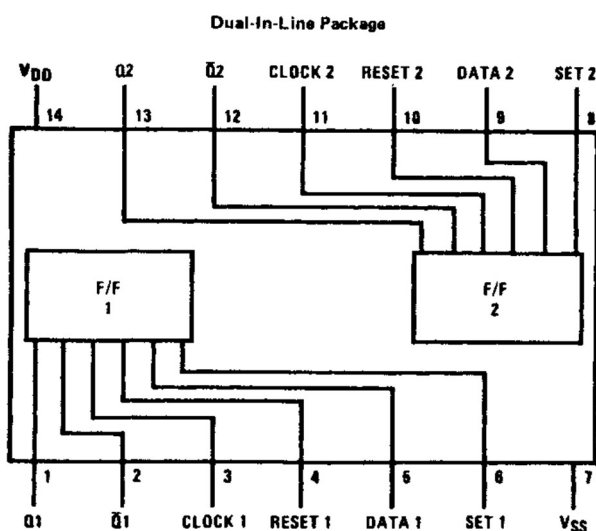
CD4027BM/CD4027BC Dual J-K Master/Slave Flip-Flop with Set and Reset



CL [†]	*t _{n-1} INPUTS					*t _n OUTPUTS	
	J	K	S	R	Q	Q	Q̄
↗	1	X	0	0	0	1	0
↘	X	0	0	0	1	1	0
↖	0	X	0	0	0	0	1
↗	X	1	0	0	1	0	1
↘	X	X	0	0	X	(No change)	
X	X	X	1	0	X	1	0
X	X	X	0	1	X	0	1
X	X	X	1	1	X	1	1

Where: 1 = High Level
 0 = Low Level
 ▲ = Level Change
 X = Don't Care
 * = t_{n-1} refers to the time interval prior to the positive clock pulse transition
 † = t_n refers to the time intervals after the positive clock pulse transition

CD4013BM/CD4013BC Dual D Flip-Flop



CL [†]	D	R	S	Q	Q̄
↗	0	0	0	0	1
↘	1	0	0	1	0
↖	x	0	0	0	Q̄
x	x	1	0	0	1
x	x	0	1	1	0
x	x	1	1	1	1

No change
 † = Level change
 x = Don't care case

Fig. 2.20 – Pin-out di alcuni flip-flop CMOS

In Fig. 2.21a e b sono illustrati in maniera schematica due registri a quattro celle, rispettivamente con scorrimento verso destra e verso sinistra.

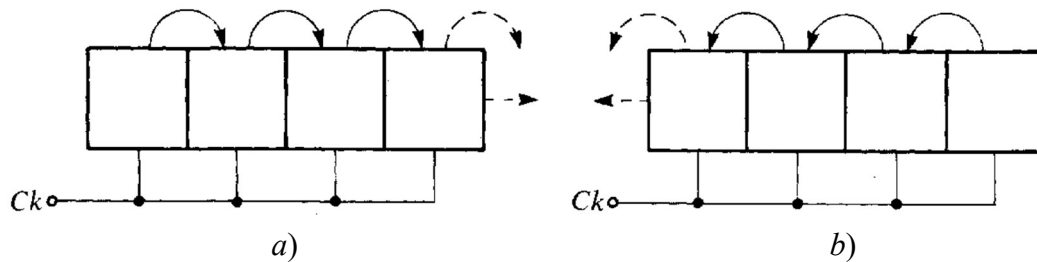


Fig. 2.21 – Rappresentazione schematica di registri a scorrimento
a) verso destra; b) verso sinistra

2.4.1 Registri SISO e SIPO

In Fig. 2.22a è illustrato un registro a scorrimento costituito da quattro flip-flop di tipo D connessi in cascata e comandati da un clock comune. Ad ogni transizione attiva (fronte positivo) del segnale di clock, il dato presente sull'ingresso D_i connesso all'uscita Q_{i-1} , viene trasferito sull'uscita Q_i . Così se ad esempio il registro si trova nello stato $Q_0 Q_1 Q_2 Q_3 = 0000$ e $D = D_0 = 1$, dopo la prima transizione attiva del clock sulle uscite si avrà la configurazione $Q_0 Q_1 Q_2 Q_3 = 1000$; dopo la seconda transizione del clock si avrà $Q_0 Q_1 Q_2 Q_3 = 1100$; successivamente si avrà 1110 e poi 1111. Pertanto, dopo quattro impulsi di clock il dato $D = 1$ risulta caricato in tutti gli elementi del registro; ovviamente lo scorrimento comporta la perdita dei dati precedentemente memorizzati.

I diagrammi di Fig. 2.22b illustrano lo stato delle uscite del registro considerando di partire da uno stato iniziale $Q_i = 0$ e $D = 1$, di mantenere $D = 1$ per un tempo pari a quattro cicli di clock, poi di porre $D = 0$ per un ciclo di clock e infine nuovamente $D = 1$.

I dati digitali che devono essere memorizzati possono essere presentati agli ingressi di un registro in forma *seriale* o *parallela*. Nel primo caso, tutti i bit di una parola vengono trasferiti su un'unica linea in sequenza ordinata, un bit dopo l'altro. Il trasferimento in parallelo richiede invece tante linee quanti sono i bit della parola. Come l'operazione di scrittura dei dati, così anche la lettura, ossia il prelievamento dei dati da un registro, può essere effettuata in modo seriale o parallelo.

Il registro illustrato in Fig. 2.22a consente la scrittura di una parola a 4 bit in modo seriale attraverso il terminale di ingresso esterno D . Se la lettura viene effettuata prelevando i bit che, in sincronismo con il segnale di clock, si presentano sul terminale di uscita Q_3 , si ottiene un dato di uscita in formato seriale. Il registro funziona in questo caso secondo una modalità *ingresso serie – uscita serie* e viene indicato con la sigla *SISO*: *serial input – serial output*.

I 4 bit memorizzati nel registro di Fig. 2.22a possono essere prelevati anche contemporaneamente, in qualsiasi istante, sulle uscite $Q_0 Q_1 Q_2 Q_3$, qualora ovviamente tali uscite siano disponibili su terminali esterni. In tal caso il circuito funziona con una modalità *ingresso serie – uscita parallelo* e viene perciò ad essere un registro di tipo *SIPO: serial input – parallel output*.

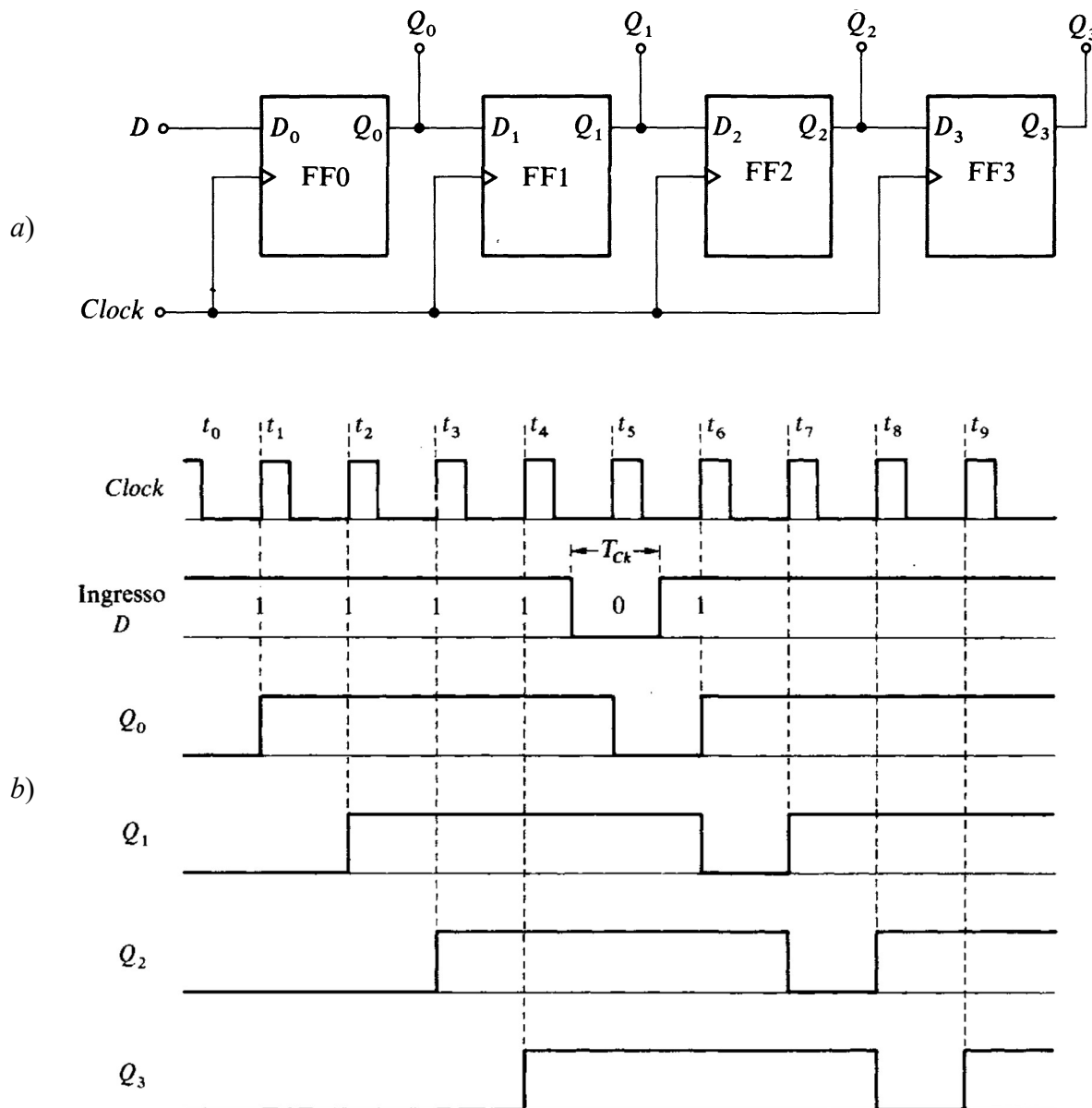


Fig. 2.22 – a) Registro a scorrimento di tipo SISO o SIPO. b) Forme d'onda d'uscita in risposta ad una sequenza d'ingresso 111101

2.4.2 Registri PIPO e PISO

Il registro illustrato in Fig. 2.23 consente il caricamento dei dati in forma parallela. I quattro bit della parola da memorizzare vengono applicati ai terminali di ingresso dati $P_0 P_1 P_2 P_3$; se l'ingresso

di controllo PE (*parallel enable*: abilitazione caricamento parallelo) è attivo (alto), i dati applicati ai terminali P_i attraversano le porte AND-OR e si presentano sugli ingressi D_i .

La prima transizione positiva del segnale di clock attiva i flip-flop, che memorizzano i dati impostati; naturalmente lo stato dei terminali P_i deve rimanere stabile finché PE è attivo per evitare la scrittura di dati errati. A questo punto PE viene disabilitato. Subito dopo l'operazione di scrittura in parallelo, ossia durante il ciclo di clock successivo, i dati sono disponibili per essere prelevati, in formato parallelo, dalle uscite Q_i ; in questo caso il registro viene utilizzato in modo *ingresso parallelo – uscita parallelo* o *PIPO: parallel input – parallel output*. Si noti però che il fronte attivo del clock immediatamente successivo a quello che ha determinato la memorizzazione causa lo scorrimento (di una posizione verso destra) dei bit nel registro e, soprattutto, la perdita del bit contenuto nell'ultimo flip-flop. Per evitare lo scorrimento bisognerebbe inserire un ulteriore ingresso di controllo che inibisca il segnale di clock.

In alternativa, i dati possono essere prelevati in forma seriale sull'uscita Q_3 durante i quattro cicli di clock che seguono il comando di memorizzazione. In questo modo si realizza un procedimento di *scrittura parallelo – lettura serie* o *PISO: parallel input – serial output*.

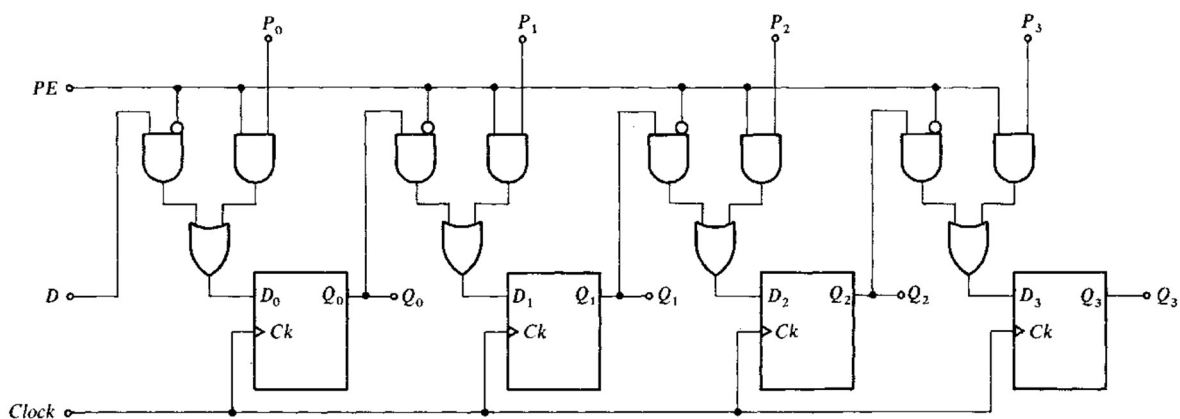


Fig. 2.23 – Registro a scorrimento di tipo PIPO o PISO

La possibilità di impiegare i registri a scorrimento con le modalità indicate si rivela indispensabile in diverse applicazioni. Dovendo trasferire dati digitali da un sistema ad un altro (ad esempio da un elaboratore ad una stampante), è molto spesso necessario effettuare sui bit della parola, ossia su ciascun dato, una *conversione di formato* da parallelo a seriale o viceversa. All'interno di un sistema digitale, infatti, i bit di una parola vengono di solito trasferiti in forma parallela, ad esempio su 8, o 16, o 32 linee; se i due sistemi si trovano ad una certa distanza, può non essere conveniente utilizzare

8 (o 16, o 32) linee per la trasmissione dei dati stessi e quindi si preferisce trasferire i bit uno alla volta, in formato seriale, su un'unica linea.

I registri a scorrimento trovano un'altra importante applicazione nel trasferimento seriale dei dati. Essi possono essere usati per variare la velocità dei dati (*data rate*) ovvero il numero di bit che, nell'unità di tempo, transita su una linea. È questa una situazione che si verifica quando un dispositivo relativamente lento, ad esempio un nastro magnetico, deve fornire dati seriali ad un dispositivo che lavora con una frequenza di clock più elevata.

Per completare il quadro, citiamo infine i cosiddetti shift register *universali* o *bidirezionali*. Questi sono registri che possono effettuare lo scorrimento sia a destra che a sinistra. La selezione del tipo di scorrimento è affidata ad una linea opportuna. Il registro in Fig. 2.23 è in effetti universale se si considera che i dati possono essere caricati serialmente attraverso l'ingresso D, quando PE è inibito.

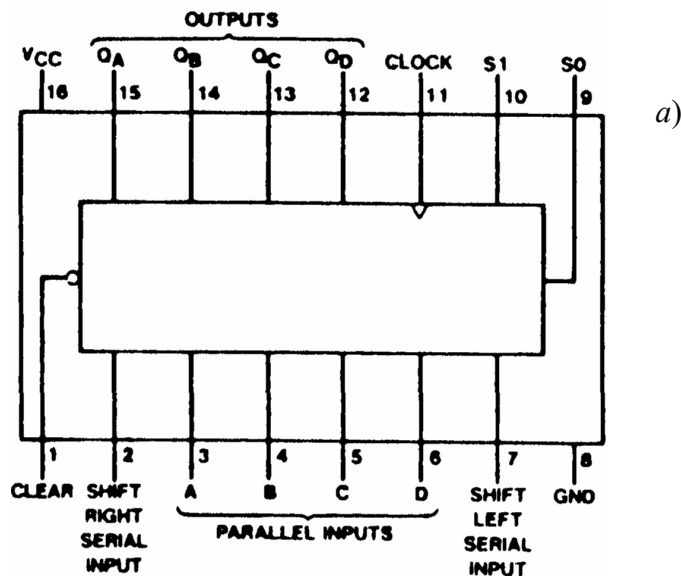
2.4.3 Registri a scorrimento commerciali

Gli shift register sono disponibili in forma integrata in un'ampia gamma di tipi con capacità di 4, 8 o anche 16 bit. Nella Tab. 2.1 vengono riportate le sigle e le caratteristiche degli integrati più comuni. Essi sono sempre dotati di un ingresso di clock e di un ingresso di azzeramento comuni a tutti i flip-flop. Alcuni tipi consentono il caricamento parallelo dei dati attuato mediante ingressi asincroni o in sincronismo con il segnale di clock (74198, 74194). In altri casi è disponibile solo un ingresso dati attraverso il quale i bit possono essere applicati serialmente, uno alla volta (74164, 4014B). La lettura può essere sempre effettuata in forma seriale mentre solo con certi dispositivi è possibile una modalità di uscita parallela (74164, 74194). Qualche componente permette anche lo scorrimento bidirezionale (74198, 74194). Alcuni integrati vengono prodotti con uscite three-state o con terminali dati bidirezionali, che possono fungere sia da ingressi sia da uscite, oppure porsi in stato di alta impedenza (74LS671, 74LS323, 4076B).

In ogni caso l'uso dei registri integrati, specie quelli più complessi, richiede un attento esame delle tabelle funzionali e dei diagrammi temporali, che riportano le forme d'onda degli ingressi e delle uscite correlate al segnale di clock e ai necessari segnali di controllo.

A tal proposito giova fornire qualche informazione sul già citato registro universale 74194, del quale si riporta il pin-out e la tabella di verità in Fig. 2.24. Esso è un PIPO bidirezionale a 4 bit; presenta 4 bit d'ingresso per il caricamento parallelo denominati *DCBA*, 2 bit d'ingresso per il caricamento seriale denominati *R* (*Right*) e *L* (*Left*): il primo è l'ingresso seriale per scorrimento a destra, il secondo è l'ingresso seriale per scorrimento a sinistra. Il clock è attivo sui fronti positivi e l'ingresso *CLEAR* funziona in logica negativa. Presenta infine altre due linee d'ingresso, denominate *S*₁ e *S*₀ con le quali si selezionano 4 possibili modi di funzionamento:

- $S_1 S_0 = 00$ il clock è inibito;
- $S_1 S_0 = 01$ è attivato lo scorrimento a destra e $Q_A = R$;
- $S_1 S_0 = 10$ è attivato lo scorrimento a sinistra e $Q_D = L$;
- $S_1 S_0 = 11$ è attivato il caricamento parallelo.



b)

Clear	Mode		Clock	Serial				Parallel				Outputs			
	S1	S0		Left	Right	A	B	C	D	QA	QB	QC	QD		
										QA0	QB0	QC0	QD0		
L	X	X	X	X	X	X	X	X	X	L	L	L	L		
H	X	X	L	X	X	X	X	X	X	QA0	QB0	QC0	QD0		
H	H	H	↑	X	X	a	b	c	d	a	b	c	d		
H	L	H	↑	X	H	X	X	X	X	H	QAn	QBn	QCn		
H	L	H	↑	X	L	X	X	X	X	L	QAn	QBn	QCn		
H	H	L	↑	H	X	X	X	X	X	QBn	QCn	QDn	H		
H	H	L	↑	L	X	X	X	X	X	QBn	QCn	QDn	L		
H	L	L	X	X	X	X	X	X	X	QA0	QB0	QC0	QD0		

Fig. 2.24 – a) Pin-out e b) tabella della verità del registro universale 74194

Tab. 2.1 – Rassegna dei principali shift register integrati

Descrizione	Sigla	Numero di pin	Modi di funzionamento				Tecnologia						Note			
			SR	SL	Load	Hold	TTL	LS	S	ALS	AS	CMOS		HC	HCT	
PIPO	8 bit	'198	24	*	*	*	*	*								2 ingressi seriali. Linee di I/O bidirezionali 3-state; Clear. Linee di I/O bidirezionali 3-state. Ingresso seriale. Ingresso seriale. 2 ingressi seriali. 2 ingressi seriali. Uscite 3-state con latch; Clear asincrono. Uscite 3-state con latch; Clear sincrono. Uscite 3-state.
		'199	24	*	*	*	*	*								
		'299	20	*	*	*	*		*	*	*	*	*	*		
	'323	20	*	*	*	*		*	*	*	*	*	*			
	5 bit	'96	16	*		*		*	*							
	4 bit	'95	14	*	*	*		A	B		*	*				
		'194	16	*	*	*	*	*	A	*	*	*	*	*		
		'195	16	*	*	*	*	*	A	*	*	*	*	*		
'671		20	*	*	*	*	*	*								
'672	20	*	*	*	*	*	*									
4076B	16	*		*	*					*						
SIPO	16 bit	'673	24	*		*	*	*							Linea di I/O seriale bidirezionale 3-state. Clear asincrono. Uscite 3-state con latch e uscita seriale.	
		'164	14	*			*	*	*		*	*	*			
	'595	16	*				*	*			*	*	*			
PISO	16 bit	'674	24	*		*	*	*							Linea di I/O seriale bidirezionale 3-state. Uscite Q e Q̄.	
		'165	16	*		*	*	*	A	*	*	*	*			
	'166	16	*		*	*	*	A	*	*	*	*	*			
SISO	8 bit	'91	14	*				A	*						Lunghezza variabile; 2 ingressi seriali.	
		4014B	16	*		*					*					
SISO	1-64 bit	4557B	16	*							*					

* Disponibile nella tecnologia indicata.
A, B Disponibile nella tecnologia indicata nella versione A o B.

2.5 Contatori

I contatori digitali sono circuiti sequenziali in grado di contare gli impulsi di clock applicati al proprio ingresso. In linea di principio, un contatore presenta un solo ingresso (clock) e n bit d'uscita. Applicando in ingresso un impulso di clock, la parola binaria d'uscita s'incrementa di un'unità.

Com'è noto, con una parola binaria di n bit sono possibili 2^n configurazioni diverse che corrispondono ai numeri decimali compresi tra 0 e $(2^n - 1)$. Un contatore può assumere tutte le possibili combinazioni oppure solo una parte.

Si definisce *modulo* il numero di configurazioni distinte che il contatore può assumere. Il contatore si dice *binario* se il suo modulo vale proprio 2^n .

I contatori si realizzano collegando dei flip-flop opportunamente tra loro.

I contatori integrati *non binari* più diffusi sono quelli *decadici* noti anche come *contatori BCD*. Essi sono a modulo 10 e pertanto assumono tutte le configurazioni binarie comprese tra 0 e 9.

Un contatore può essere progettato anche per contare indietro (*down counter*); altri, invece, sono in grado di contare sia in avanti che indietro (*up/down counter*).

Secondo il modo con cui sono collegati gli ingressi di clock dei vari flip-flop, i contatori si dividono in due categorie: *asincroni* e *sincroni*. Nei primi, un flip-flop cambia stato dopo la commutazione del flip-flop che lo precede; in questo caso il comando di commutazione si propaga da un flip-flop all'altro (*ripple counter*), in modo simile a quanto visto per i sommatore *ripple*. Nei secondi l'ingresso di clock del contatore è applicato simultaneamente all'ingresso di clock di tutti i flip-flop. I contatori sincroni sono più veloci di quelli asincroni ma ovviamente risultano più complessi.

Alcuni contatori integrati presentano delle linee supplementari come, ad esempio, l'ingresso di *CLEAR* che serve per l'azzeramento del dispositivo, l'ingresso di *LOAD* che permette il caricamento nel contatore di una configurazione binaria prestabilita, gli ingressi di *clock up* e *clock down* nei contatori bidirezionali, un'uscita di riporto per il collegamento in cascata, ecc.

I contatori vengono utilizzati, oltre che per contare determinati eventi (posti sotto forma di impulsi di clock), anche per misurare il tempo, la velocità e, soprattutto, per *dividere la frequenza*.

2.5.1 Contatori asincroni

Un contatore binario in avanti costituito, ad esempio, da tre bit deve soddisfare la tavola di Fig. 2.25, nota come *tavola degli stati*.

CK	Q_3	Q_2	Q_1	N
0	0	0	0	0
1	0	0	1	1
2	0	1	0	2
3	0	1	1	3
4	1	0	0	4
5	1	0	1	5
6	1	1	0	6
7	1	1	1	7
8	0	0	0	0
9	0	0	1	1
...

Fig. 2.25 – Tavola degli stati di un contatore binario a 3 bit

Essa rappresenta l'evoluzione temporale del contatore, supposto inizialmente azzerato, in funzione del numero di impulsi di clock applicati in ingresso. Con N (che viene anche indicato come *stato*) si indica il corrispondente valore decimale assunto dal contatore. Si noti che, dopo il settimo impulso di clock, il contatore assume lo stato $Q_3 Q_2 Q_1 = 111$; all'ottavo impulso di clock il contatore assume la configurazione iniziale e così via.

Per realizzare il circuito asincrono che soddisfa la precedente tavola degli stati basta osservare il comportamento del generico bit Q di uscita in funzione del bit che lo precede. In particolare si nota che l'uscita Q_1 commuta ad ogni impulso di clock per cui essa sarà l'uscita di un flip-flop di tipo T avente l'ingresso $T = 1$ eccitato dall'ingresso di clock del contatore. L'uscita Q_2 , invece, commuta solo quando la linea Q_1 passa dallo stato alto allo stato basso (transizione negativa) per cui essa sarà l'uscita di un altro flip-flop di tipo T con $T = 1$ master-slave o NET il cui ingresso di clock, è collegato all'uscita Q_1 . L'uscita Q_3 , infine, commuta solo quando la linea Q_2 passa dallo stato alto a quello basso per cui anche il terzo flip-flop è di tipo T master-slave o NET con $T = 1$.

In sostituzione dei flip-flop T si impiegano quelli di tipo JK in configurazione *toggle*, cioè ponendo $J = K = T$. Nel caso in esame tutti i flip-flop saranno pilotati con $J = K = 1$.

In Fig. 2.26 si mostra lo schema elettrico del contatore binario a tre bit (Fig. 2.26a) e le relative forme d'onda (Fig. 2.26b).

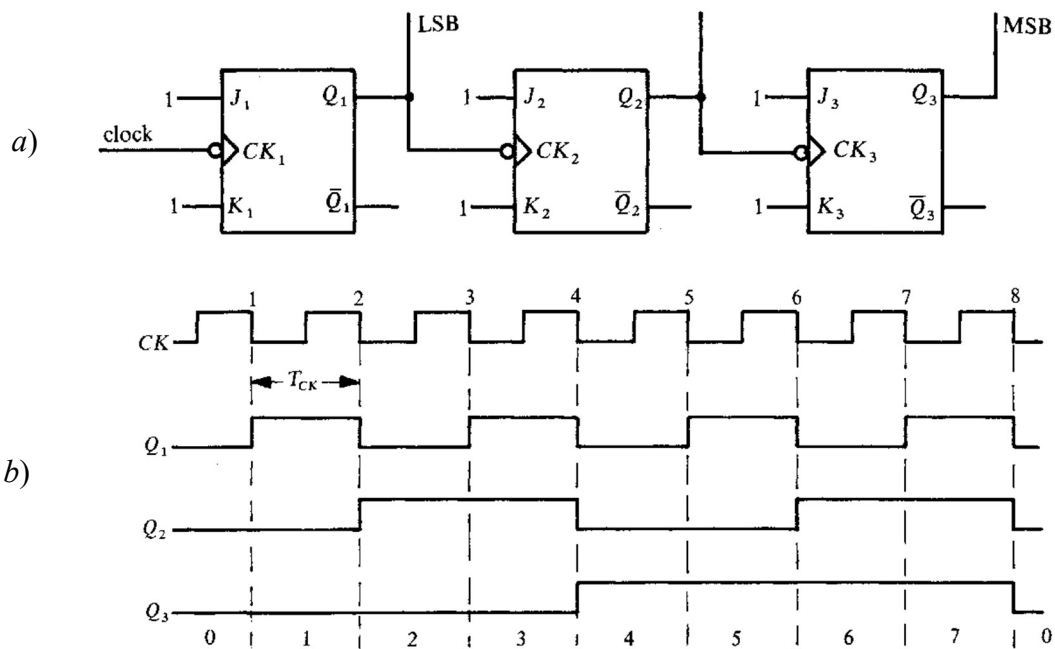


Fig. 2.26 – Contatore asincrono modulo 8: a) schema circuitale e b) relative forme d'onda

Per aumentare la capacità di conteggio basta aggiungere in cascata altri flip-flop con $J = K = 1$ e collegare l'ingresso di clock all'uscita Q del flip-flop precedente. Un contatore ad 8 bit, ad esempio, è realizzato da 8 flip-flop e può contare da 0 a 255.

Se il clock è una forma d'onda periodica di periodo T_{CK} , come ad esempio quella di Fig. 2.26b, anche le uscite Q_1 , Q_2 e Q_3 sono periodiche. Più precisamente: Q_1 è un'onda quadra avente periodo $2 \cdot T_{CK}$, Q_2 è anch'essa un'onda quadra, ma di periodo $4 \cdot T_{CK}$, infine Q_3 è un'onda quadra di periodo

$8 \cdot T_{CK}$. L'uscita del flip-flop n di un contatore binario pilotato da un clock periodico è, quindi, una forma d'onda quadra di periodo pari a $2^n \cdot T_{CK}$ e perciò è un *divisore di frequenza* di clock di una quantità pari a 2^n .

Il periodo del segnale di clock T_{CK} deve essere non inferiore al tempo di propagazione totale della catena di flip-flop. Detto t_p il tempo di propagazione di ciascun flip-flop, se il contatore è costituito da n flip-flop, deve verificarsi $T_{CK} \geq n \cdot t_p$.

La massima frequenza del segnale di clock vale: $f_{max} = 1/(n \cdot t_p)$. Nel caso in esame, supponendo $t_p = 40$ ns, si ottiene $f_{max} \approx 8.3$ MHz.

Un contatore asincrono capace di contare indietro decrementa la configurazione binaria ad ogni impulso di clock applicato in ingresso. Un contatore indietro a tre bit soddisfa la tavola degli stati mostrata in Fig. 2.27.

CK	Q_3	Q_2	Q_1	N
0	1	1	1	7
1	1	1	0	6
2	1	0	1	5
3	1	0	0	4
4	0	1	1	3
5	0	1	0	2
6	0	0	1	1
7	0	0	0	0
8	1	1	1	7
9	1	1	0	6
...

Fig. 2.27 – *Tavola degli stati di un contatore binario indietro a 3 bit*

La tavola degli stati di Fig. 2.27 si ricava dalla tavola di Fig. 2.25 semplicemente commutando tutti i bit delle uscite Q_3 , Q_2 e Q_1 . Il circuito che realizza il contatore indietro a tre bit resta identico a quello mostrato in Fig. 2.26a con la differenza che i bit di uscita coincidono con le uscite negate \bar{Q} dei singoli flip-flop.

Analogamente possono realizzarsi contatori *up/down*, cioè contatori in grado di effettuare i due tipi di conteggio, sia in avanti che indietro. Nei contatori integrati per scegliere la direzione del conteggio è disponibile un'opportuna linea di selezione.

Sinora i contatori esaminati sono stati tutti binari, cioè aventi modulo pari ad una potenza di 2. Un contatore asincrono non binario a n bit a modulo $m < 2^n$ assume tutte le configurazioni comprese tra 0 e $(m - 1)$. Al successivo impulso di clock il contatore riprende la configurazione (o *stato*) 0 interrompendo così la sequenza binaria. Ad esempio, un contatore modulo 5 è realizzato con 3 flip-flop ($n = 3$) e assume tutti gli stati compresi tra 0 e 4. Al successivo impulso di clock, la sequenza viene interrotta in modo che lo stato successivo a 4 sia nuovamente 0.

Il circuito si realizza impiegando flip-flop dotati di ingressi asincroni di *CLEAR* che vengono abilitati quando l'applicazione del clock porta il contatore nello stato corrispondente al modulo richiesto in modo che il contatore si azzeri. Nel caso precedente dopo lo stato 4 si presenta lo stato 5 (modulo richiesto) e ciò deve determinare l'immediato azzeramento del contatore. In pratica l'ultimo stato stabile è 4 mentre lo stato 5 ha una durata pari al tempo di ritardo di propagazione del circuito di pilotaggio degli ingressi di *CLEAR*. Lo stato 5, per questo motivo, prende il nome di *stato instabile*.

In Fig. 2.28 si mostra lo schema elettrico e l'andamento temporale delle uscite di un contatore asincrono modulo 5.

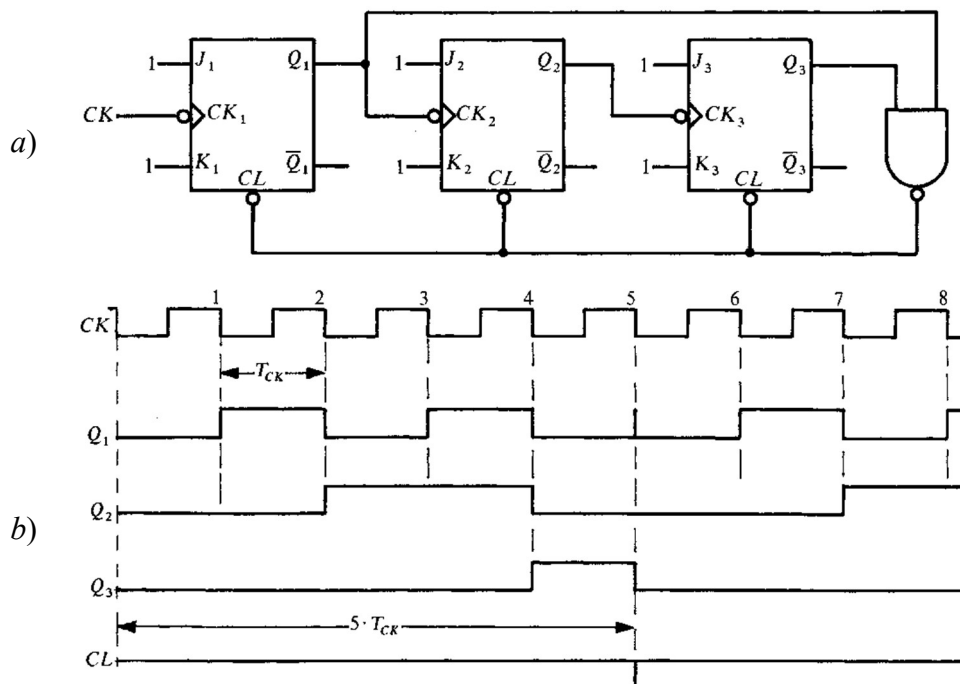


Fig. 2.28 – Contatore asincrono modulo 5: a) schema circuitale e b) relative forme d'onda

Dal circuito si osserva che, raggiunto lo stato 5, sia Q_3 che Q_1 si portano allo stato alto e l'uscita della porta NAND va allo stato basso in modo da resettare immediatamente i flip-flop. Questa operazione, riportando a 0 sia Q_1 che Q_3 , rende la linea *CL* nuovamente ed immediatamente uguale a 1. Il breve impulso a livello basso che si origina sulla linea *CL* prende il nome di *spike* (letteralmente, *punta di spillo*). Anche la linea Q_1 , in questo caso, presenta uno spike.

Si noti infine che tale contatore modulo 5 si comporta anche da divisore di frequenza per 5 (ma con duty cycle diverso da 0,5)

Contatori non binari possono ovviamente essere realizzati di tipo indietro e up-down.

2.5.2 Contatori sincroni

A differenza dei contatori asincroni, in quelli sincroni il clock è applicato simultaneamente a tutti i flip-flop. In questo modo si raggiungono tre obiettivi:

1. i bit di uscita commutano contemporaneamente;
2. la massima frequenza di funzionamento aumenta perché tutti i flip-flop, supposti uguali, commutano con lo stesso tempo di ritardo di propagazione dopo l'applicazione del clock;
3. assenza di spike indesiderati sulle linee di uscita.

Le porte logiche da inserire pilotano solamente gli ingressi *JK* o *T* a seconda del tipo di dispositivo impiegato perché l'ingresso di sincronismo è già impegnato dal clock esterno.

Ad esempio, per la progettazione di un contatore sincrono in avanti a 3 bit (quindi modulo 8) si dovrà tener presente la tabella degli stati già mostrata in Fig. 2.25, ma si dovrà ragionare nel seguente modo:

1. il flip-flop 1 commuta ad ogni impulso di clock, quindi: $T_1 = 1$;
2. il flip-flop 2 commuta quando $Q_1 = 1$, quindi $T_2 = Q_1$;
3. infine, il flip-flop 3 commuta quando $Q_2 = Q_1 = 1$ e quindi $T_3 = Q_2 \cdot Q_1$. Infatti Q_3 passa da 0 a 1 quando si ha il passaggio di stato dal 3 (011) al 4 (100) e dal 7(111) allo 0 (000).

Naturalmente, se si utilizzano flip-flop *JK* si deve porre $J = K = T$.

In Fig. 2.29 si mostra lo schema elettrico del contatore sincrono binario a 3 bit.

Per aumentare la capacità di conteggio basta aggiungere altri flip-flop in modo che per l'*i*-mo dispositivo si abbia:

$$T_i = Q_{i-1} \cdot Q_{i-2} \cdots Q_1. \tag{2.11}$$

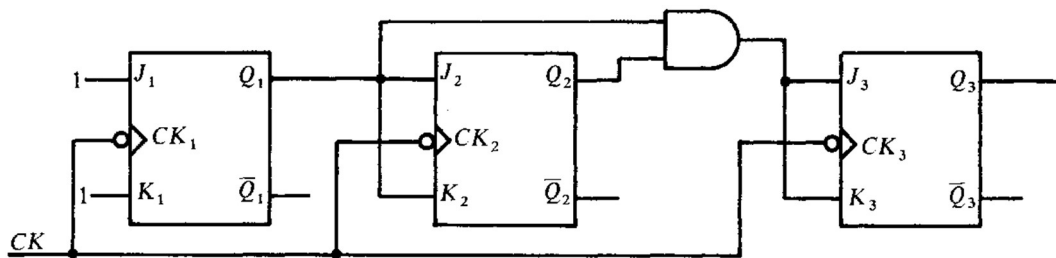


Fig. 2.29 – Contatore sincrono a 3 bit (modulo 8)

In generale per un contatore a *n* bit sono necessarie (*n* – 2) porte AND. Supponendo i dispositivi uguali e indicando con t_{p1} il tempo di ritardo di propagazione di un flip-flop e con t_{p2} quello della porta AND, la successione degli impulsi di clock deve avvenire non prima di un tempo pari a

$$T_{CK} \geq (n - 2) \cdot t_{p2} + t_{p1}. \tag{2.12}$$

La massima frequenza di funzionamento pertanto è

$$f_{\max} = \frac{1}{T_{CK}} = \frac{1}{(n-2) \cdot t_{p2} + t_{p1}} \quad (2.13)$$

Il dispositivo è dunque più veloce del corrispondente dispositivo asincrono. Tuttavia, si deve notare che, sebbene il clock sia fornito contemporaneamente a tutti i flip-flop, il comando tra un flip-flop e l'altro tramite le porte AND avviene in modo *seriale*. La propagazione del comando può avvenire anche in *parallelo*; l'unico inconveniente è che sono necessarie porte AND il cui numero d'ingressi aumenta con l'aumentare del numero di bit.

In Fig. 2.30 sono confrontati due contatori a 6 bit (modulo 64), uno con propagazione del comando in serie e l'altro con propagazione del comando in parallelo. Nella soluzione con propagazione del comando in parallelo il conteggio può avvenire a frequenze più elevate, in quanto il comando $T = 1$ giunge a tutti i flip-flop contemporaneamente dopo un tempo di ritardo dovuto alla porta logica più quello proprio del flip-flop, cioè

$$T_{CK} \geq t_{p2} + t_{p1} \quad (2.14)$$

La massima frequenza di funzionamento vale in questo caso

$$f_{\max} = \frac{1}{t_{p2} + t_{p1}} \quad (2.15)$$

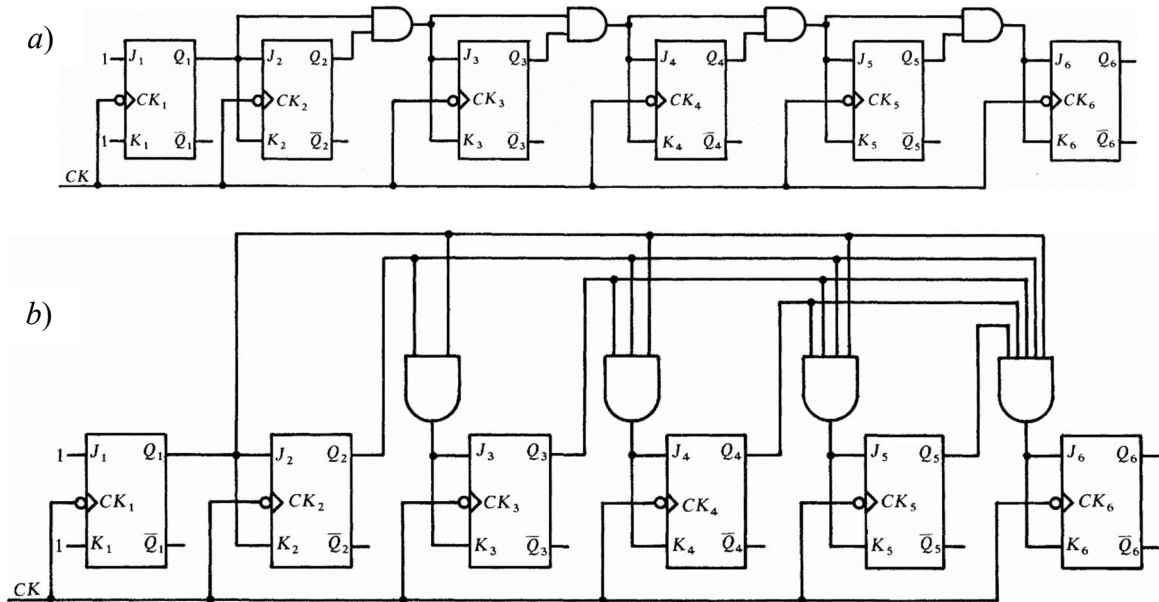


Fig. 2.30 – Contatore sincroni modulo 64 con propagazione del comando: a) in serie e b) in parallelo

Analogamente a quanto visto per i contatori asincroni, è possibile realizzare contatori binari e non, sia indietro che up/down. Nel caso di questi ultimi, è necessario l'impiego di multiplexer a due

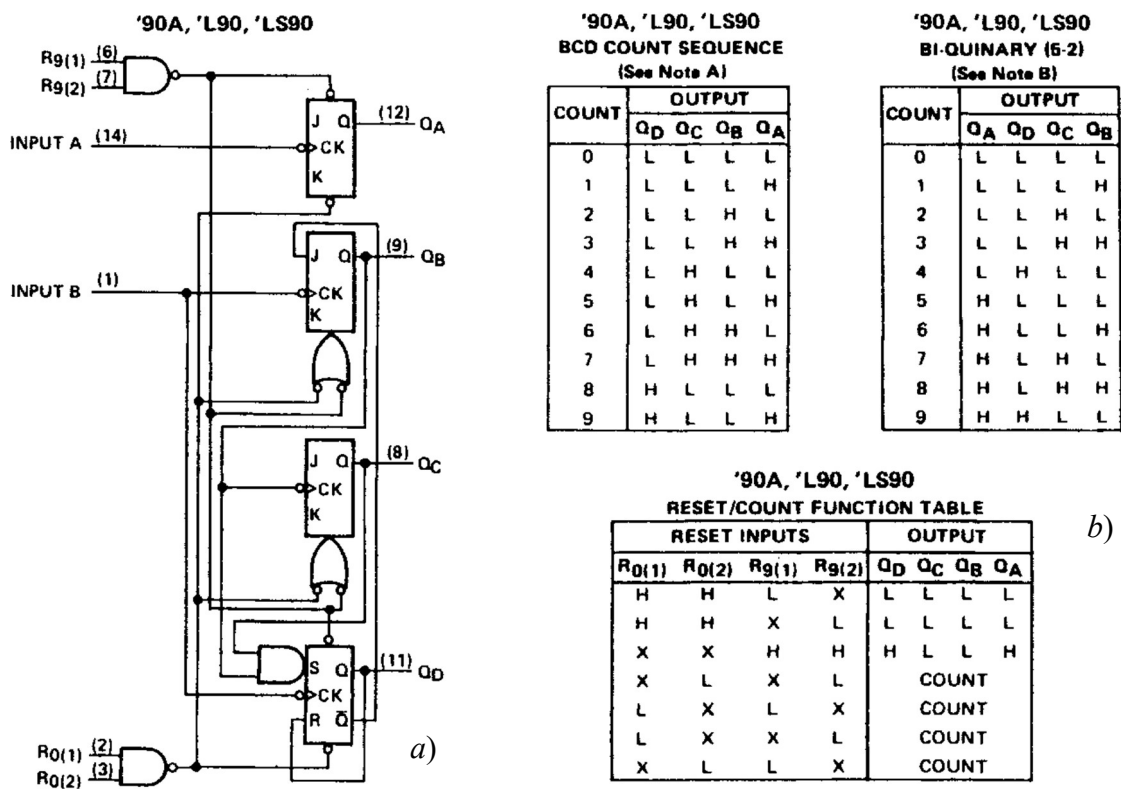
ingressi per ciascun flip-flop (tranne che per il primo) per il pilotaggio dell'ingresso *toggle* (cioè degli ingressi *JK*). La variabile di selezione *S* applicata ai multiplexer comanda la direzione del conteggio.

2.5.3 Contatori commerciali

In commercio sono disponibili contatori asincroni e sincroni integrati da 4 o più bit, la maggior parte dei quali capaci di contare solamente in avanti; essi sono binari e non binari. I non binari più diffusi sono a modulo 10.

I contatori asincroni della famiglia logica TTL sono a 4 bit di tipo binario, decadico e dodecadico. La potenza dissipata e la massima frequenza di funzionamento sono i tipici valori della TTL: 150 mW e 50 MHz.

Tipici esempi di contatori TTL asincroni, disponibili anche come HCMOS, sono il 7490, il 7493 e il 7492. Il 7490, il cui schema logico e le relative tabelle di verità sono mostrate in Fig. 2.31, è costituito da 4 flip-flop *JK* master-slave e porte logiche aggiuntive.



- NOTES: A. Output Q_A is connected to input B for BCD count.
 B. Output Q_D is connected to input A for bi-quinary count.
 C. Output Q_A is connected to input B.
 D. H = high level, L = low level, X = irrelevant

Fig. 2.31 – a) Schema interno del contatore 7490 e b) relative tabelle di verità

Esso è diviso in due sezioni: la prima è il flip-flop *A* indipendente con ingresso di clock indicato con INPUT A montato come contatore modulo 2 (o divisore di frequenza per due) mentre la seconda è realizzata dai tre flip-flop *B*, *C* e *D* montati come contatore modulo 5 con ingresso di clock indicato con INPUT B. In questo modo il dispositivo può funzionare in due modi: come contatore BCD o come contatore bi-quinario (ossia in grado di contare due sequenze di 5 conteggi). Per realizzare il contatore BCD basta collegare l'uscita Q_A all'ingresso INPUT B in modo da effettuare un collegamento asincrono tra il flip-flop *A* e il contatore modulo 5 costituito dai flip-flop *B*, *C* e *D*.

Il circuito possiede due ingressi di *reset* in grado di portare il contatore *nello stato 0*, in maniera asincrona indipendentemente dallo stato in cui si trova, se si verifica $R_0(1) = R_0(2) = 1$. Il circuito, infine, possiede due ingressi di preassegnazione a 9 in grado di portare il contatore *nello stato 9* ($Q_D Q_C Q_B Q_A = 1 0 0 1$) in maniera asincrona indipendentemente dallo stato in cui si trova, se si pone $R_9(1) = R_9(2) = 1$.

Il 7493 è simile al 7490 dal quale differisce per i tre flip-flop *D*, *C* e *B* che qui sono collegati come contatore modulo 8 e per l'assenza di ingressi di preassegnazione. Il circuito interno del 7493 è più semplice del 7490. Per realizzare il contatore binario a 4 bit basta applicare gli impulsi di clock all'ingresso INPUT A e collegare l'uscita Q_A all'ingresso INPUT B prelevando, come uscita, la parola binaria $Q_D Q_C Q_B Q_A$.

In Fig. 2.32 si mostra lo schema interno e le tabelle della verità del contatore 7493.

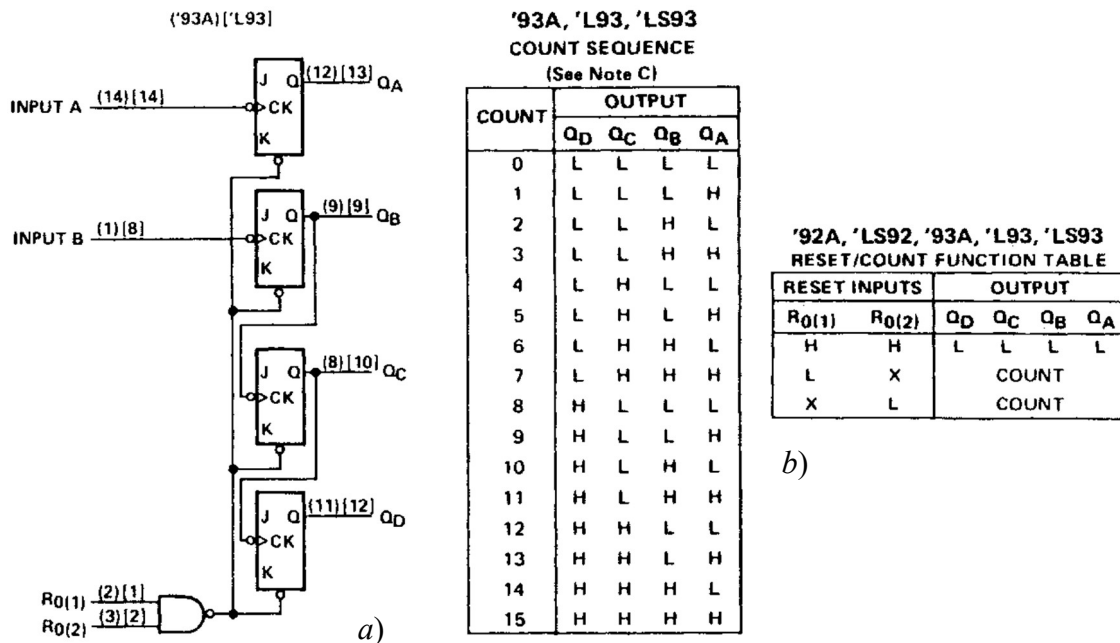


Fig. 2.32 – a) Schema interno del contatore 7493 e b) relative tabelle di verità

Il 7492 è anch'esso un contatore del tutto simile al 7490 con la differenza che i flip-flop *D*, *C* e *B* sono collegati tra loro come contatore modulo 6 e non presenta ingressi di preassegnazione. Lo schema a blocchi funzionale è identico a quello del 7493 mostrato in Fig. 2.32a. Applicando gli impulsi di clock sull'ingresso INPUT A e collegando l'uscita Q_A all'ingresso INPUT B si ottiene il contatore modulo 12.

In Fig. 2.33 si mostra lo schema interno del 7492 insieme alla tabella di funzionamento.

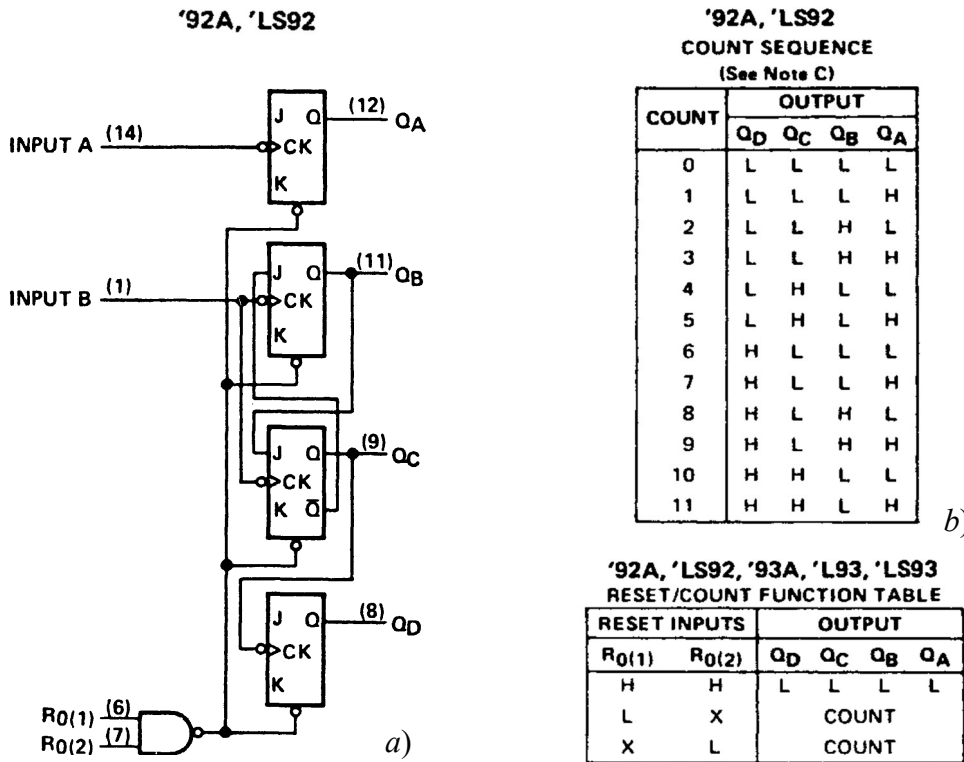


Fig. 2.33 – a) Schema interno del contatore 7492 e b) relative tabelle di verità

I contatori 7490 e 7493 sono commercialmente disponibili anche nelle serie 74C, 74HC CMOS e sono identici a quelli TTL esaminati precedentemente per quanto riguarda la piedinatura e il modo di funzionamento. Le caratteristiche elettriche, al solito, sono quelle tipiche della famiglia logica CMOS: tensione di alimentazione da 3 V a 15 V, massima frequenza di funzionamento 10 MHz, potenza dissipata trascurabile.

Della serie CMOS 4000 annoveriamo i contatori 4020, 4040 e 4024; essi sono, rispettivamente, a 14, 12 e 7 bit. Presentano l'ingresso di clock Φ attivo sul fronte negativo e l'ingresso di RESET attivo sul livello logico alto.

Fra i contatori sincroni integrati, i più diffusi sono a 4 bit con caricamento parallelo, conteggio unidirezionale e bidirezionale, di tipo binario e BCD. I tipi più noti sono i contatori TTL 74192 e

74193, esistenti anche in versione CMOS. Entrambi sono di tipo up/down a 4 bit e presentano la stessa piedinatura, come riportato nello schema funzionale di Fig. 2.34.

La differenza tra i due contatori è che il 74192 è di tipo BCD, mentre il 74193 è binario. Sono provvisti di ingressi UP e DOWN attivi sul fronte positivo. Per contare in avanti si pone DOWN = 1 (stato di riposo) e si applicano gli impulsi di clock all'ingresso UP; viceversa per il conteggio DOWN.

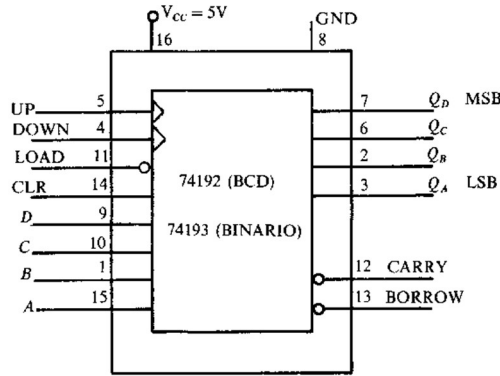


Fig. 2.34 – Schema a blocchi funzionale dei contatori sincroni 74192 e 74193

Le uscite CARRY e BORROW vengono utilizzate per effettuare il collegamento in cascata per aumentare la capacità di conteggio. In tal caso l'uscita CARRY (BORROW) del contatore meno significativo è collegato all'ingresso UP (DOWN) del successivo.

Ponendo l'ingresso CLR = 1 si resetta il contatore.

Infine, portando la linea LOAD = 0 si memorizza nei 4 flip-flop il dato presente sulle linee D, C, B e A indipendentemente dagli impulsi di clock.

Per concludere vediamo lo schema applicativo di un circuito a contatori riportato in Fig. 2.35.

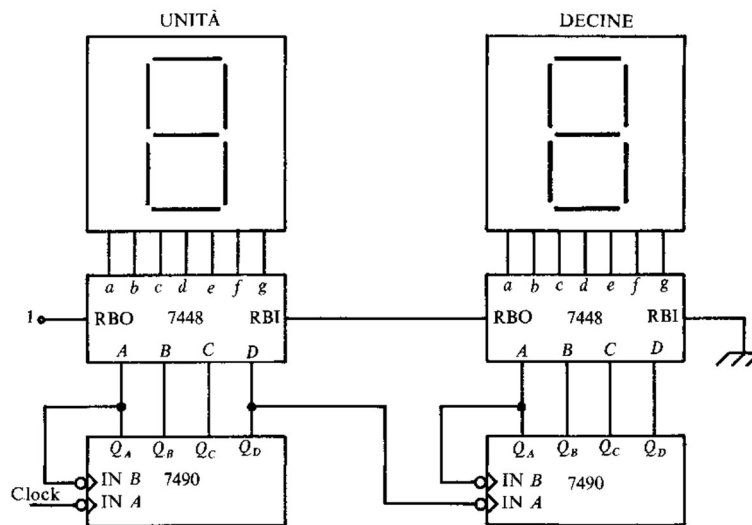


Fig. 2.35 – Contatore modulo 100 con visualizzazione su display

Esso è un contatore BCD modulo 100 che permette la visualizzazione degli impulsi contati tramite due display a sette segmenti. Per la visualizzazione delle cifre si collegano le uscite di ciascun contatore agli ingressi di un decoder (7448) in grado di pilotare i display.

Un quadro sinottico è riportato nella Tab. 2.2 nella quale sono evidenziate le sigle e le caratteristiche dei contatori integrati più comuni.

Tab. 2.2 – Rassegna dei principali contatori integrati

Descrizione	Sigla	Numero di pin	Caratteristiche			N. bit	Tecnologia								Note
			U _i /D	LOAD	CLEAR		TTL	LS	S	ALS	AS	CMOS	HC	HCT	
A decade asincroni	'90	14		Set a 9	H	4	*	*				*			Contatore per 2 e per 5. Può funzionare come registro. Equivalente al '90 ma con diverso pin-out. Versione duale del '290 (senza Set a 9). Versione duale del '290.
	'196	14		Asin	L	4	*	*	*						
	'290	14		Set a 9	H	4	*	*							
	'390	16		—	H	4	*	*					*	*	
	'490	16		Set a 9	H	4	*	*		*					
Binari asincroni	'93	14		—	H	4	*	*				*	*	*	Può funzionare come registro. Equivalente al '93 ma con diverso pin-out. Versione duale del '293. Registro interno con uscite tri-state. Linee di I/O bidirezionali 3-state.
	'197	14		Asin	L	4	*	*	*						
	'293	14		—	H	4	*	*							
	'393	14		—	H	4	*	*				*	*	*	
	4020B	16		—	H	14					*	*	*		
	'590	16		—	L	8						*	*	*	
Altri asincroni	'92	14		--	H	4	A	A							Contatore-divisore per 12. Divisore di frequenza per 5, 10, 50. Divisore a 14 stadi con oscillatore interno.
	'56	8		--	H	1		*							
	4060B	16		--	H	10					*	*	*		
A decade sincroni	'160	16		Sin	L-Asin	4	*	A		B	*	*	*	*	Ingresso di modo D/ \bar{U} . 2 ingressi di clock (Ck up e Ck down).
	'162	16		Sin	L-Sin	4	*	A	*	B	*	*	*	*	
	'190	16	U/D	Asin	—	4	*	*		*	*	*	*	*	
	'192	16	U/D	Asin	H-Asin	4	*	*		*	*	*	*	*	
Binari sincroni	'161	16		Sin	L-Asin	4	*	A		B	*	*	*	*	Ingresso di modo D/ \bar{U} . 2 ingressi di clock (Ck up e Ck down). 2 ingressi di modo (S1, S0)
	'163	16		Sin	L-Sin	4	*	A	*	B	*	*	*	*	
	'191	16	U/D	Asin	—	4	*	*		*	*	*	*	*	
	'193	16	U/D	Asin	H-Asin	4	*	*		*	*	*	*	*	
Altri sincroni	'869	24	U/D	Sin	Sin	8				*					
	'97	16		—	—	1	*								Moltiplicatore di frequenza: $f_o = MF_i/64$. Contatore a decade o binario. Contatore Johnson, uscite decodificate.
	4029B	16	U/D	Asin	—	4				*					
4017B	16		—	H	10				*		*				

* Disponibile nella tecnologia indicata.
 A, B Disponibile nella tecnologia indicata nella versione A o B.

