

7.5 I registri

Def. Nei sistemi digitali, l'unità d'informazione non è generalmente costituita dal singolo *bit* (*Binary Digit*), ma da un gruppo di bit che vengono trattati insieme e che insieme assumono significato, secondo un determinato codice; tale gruppo di bit costituisce la **parola** (*word*) del sistema.

Frequentemente la lunghezza della parola coincide con il **nibble** (4 bit) o con il **byte** (8 bit) o con multipli del **byte**.

Viene definito **registro** un gruppo di latch o di flip-flop organizzato per memorizzare una parola; naturalmente saranno necessari tanti bistabili quanti sono i bit della parola da memorizzare.

L'operazione di inserimento della parola nel registro è detta **scrittura** (*write*), mentre quella di estrazione della parola memorizzata è detta **lettura** (*read*).

Scrittura e lettura del registro sono effettuate in forma **parallela** se gli n bit della parola vengono trasferiti contemporaneamente sfruttando n linee; in alternativa si parla di scrittura e lettura **seriale**, in cui i dati sono trasferiti uno alla volta tramite un'unica linea, ma in n istanti successivi.

7.5.1 Il registro parallelo

Def. Il **registro parallelo** è costituito da n latch D (fig. 7.36a) o da n flip-flop D (fig. 7.36b), tutti comandati dallo stesso segnale di enable o di clock. In questo modo è possibile memorizzare una parola di n bit posta sugli n ingressi e mantenerla disponibile sulle n uscite fino al successivo livello attivo dell'enable o fronte attivo del clock.

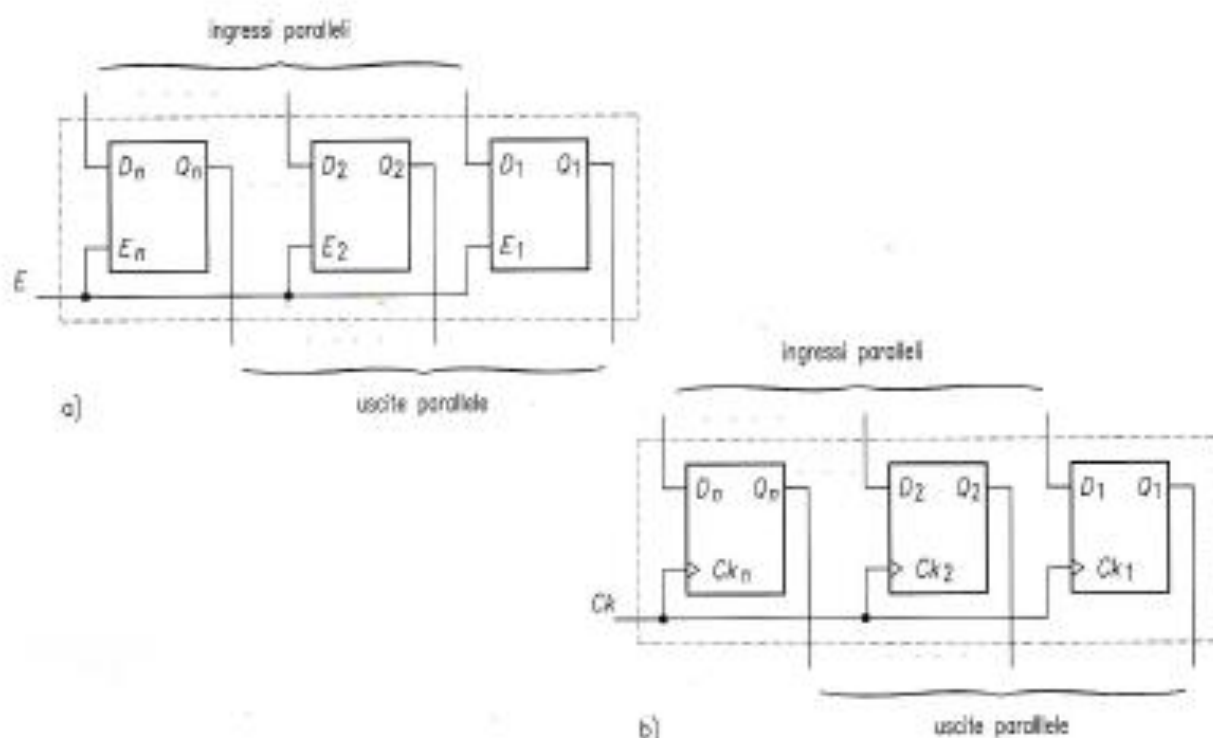
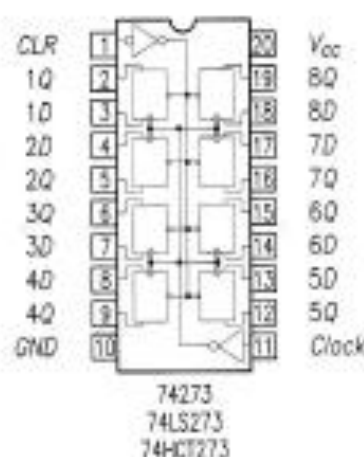


Fig. 7.36 Registro parallelo: a) con latch D; b) con flip-flop D.



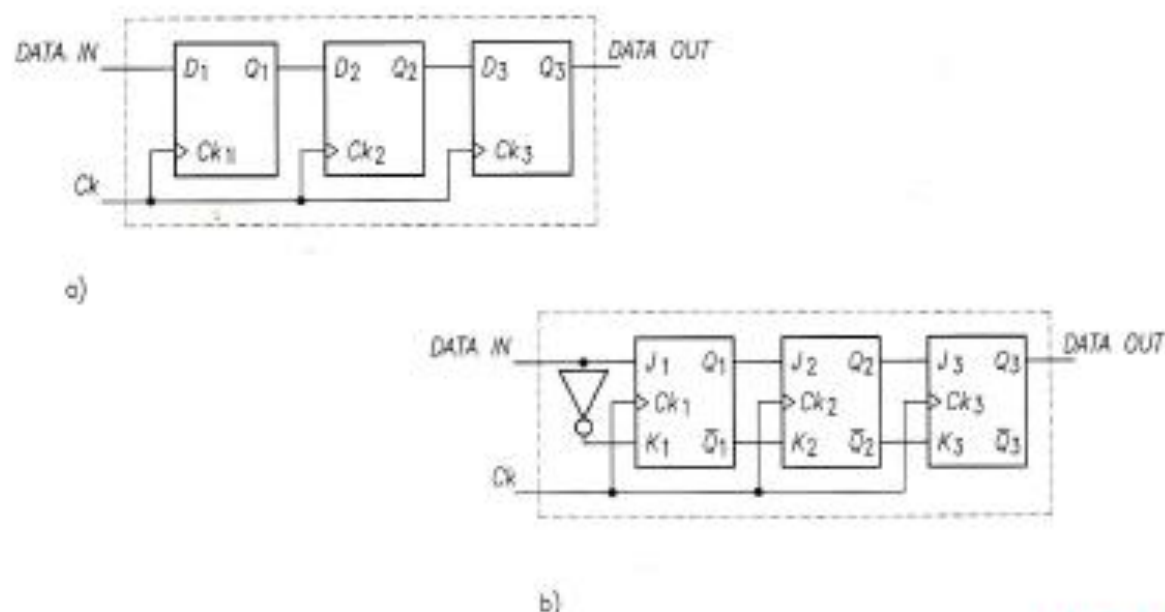
Per la modalità di funzionamento appena descritta questo registro è detto anche **PIPO** (*Parallel Input-Parallel Output*).

Alcuni registri paralleli integrati sono il 74373 (asincrono) ed il 74374 (sincrono) i cui pin-out sono già stati rappresentati nelle fig. 7.30 e 7.31. Un altro registro parallelo sincrono è il 74273 (fig. 7.37) che dispone di un comando asincrono di *clear* (CLR) che, se attivato, azzerava le uscite di tutti i flip-flop.

Fig. 7.37 Pin-out del registro parallelo integrato 74373.

7.5.2 Il registro a scorrimento

Def. Il **registro a scorrimento** (*shift register*) è costituito da n flip-flop D o JK collegati in cascata, come indicato nella fig. 7.38; si noti che l'uscita di ogni flip-flop è inviata all'ingresso del successivo e che tutti i bistabili sono comandati dal medesimo segnale di clock.



Realizzazione di uno shift register: a) con flip-flop D; b) con flip-flop JK. **Fig. 7.38**

Il funzionamento è il seguente:

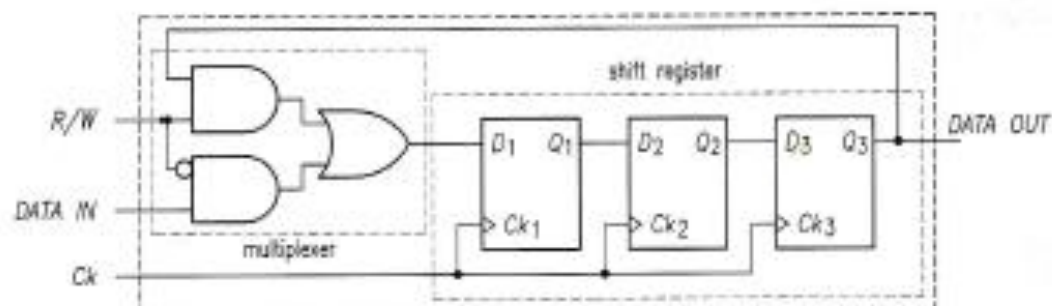
ad ogni fronte attivo di clock i dati presenti agli ingressi dei flip-flop passano sulle uscite, traslando (*shift*) di una posizione verso destra, cioè verso l'uscita del registro.

Il caricamento seriale di n flip-flop necessita quindi di n colpi di clock. Ora il primo dato acquisito può essere letto sull'uscita seriale e con altri $n - 1$ fronti attivi di clock tutti i dati transitano in uscita.

Si osservi che questo registro non può essere realizzato con latch, poiché nell'intervallo di abilitazione tutti i bistabili diventerebbero trasparenti e quindi tutte le uscite assumerebbero il valore presente in ingresso.

Il registro a scorrimento, utilizzato con scrittura e lettura seriale, è detto anche SISO (*Serial Input-Serial Output*).

L'operazione di lettura provoca la perdita dei dati memorizzati ma, se è necessario evitare questo inconveniente, si possono reinserire in ingresso i dati letti. Per ottenere ciò si utilizza un multiplexer a due ingressi (fig. 7.39) comandato dall'ingresso di selezione R/\overline{W} con la seguente logica: se $R/\overline{W} = 1$ i dati letti sull'uscita sono reinseriti all'ingresso dello shift register; se $R/\overline{W} = 0$ vengono introdotti nuovi dati dall'ingresso DATA IN e quelli letti in uscita si perdono.



Shift register con reinserimento in ingresso dei dati letti in uscita. **Fig. 7.39**

È possibile realizzare un **registro a scorrimento bidirezionale** (fig. 7.40), ponendo all'ingresso di ogni flip-flop un multiplexer che carica il dato proveniente dal bistabile di destra o di sinistra, a seconda del segnale di controllo comune *LEFT / RIGHT*. In questo modo si ottiene una traslazione verso destra o verso sinistra di tutti i dati contenuti nel registro. Ora è possibile inserire i dati da destra, tramite l'ingresso D_L , o da sinistra, tramite l'ingresso D_R .

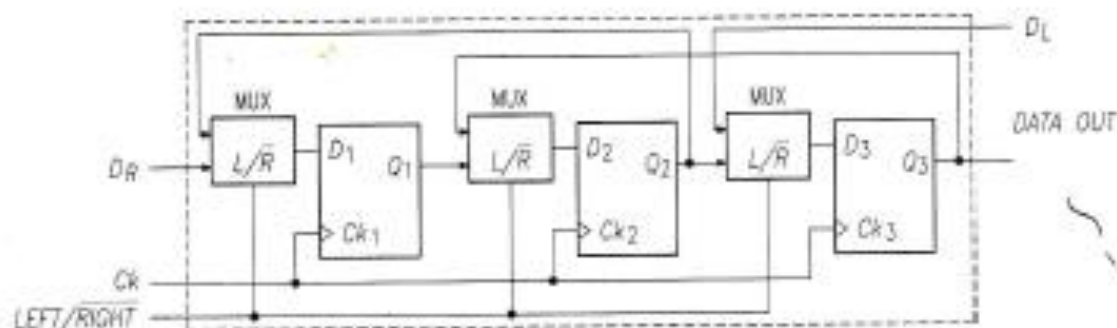


Fig. 7.40 Shift register bidirezionale.

Con un registro bidirezionale, una volta inseriti i dati attraverso l'ingresso del flip-flop più a sinistra, è possibile invertire il verso di scorrimento leggendo i dati all'uscita del medesimo flip-flop; in questo modo l'ultimo dato inserito in ingresso risulta il primo letto in uscita.

Una gestione di questo tipo viene detta **LIFO** (*Last In-First Out*); i dati possono essere pensati come posti uno sopra l'altro in una pila (*stack*), dove l'unico accessibile è quello inserito per ultimo.

La gestione seriale unidirezionale viene invece detta **FIFO** (*First In-First Out*), poiché il primo dato posto in ingresso al registro è anche il primo che si presenta in uscita, dopo n colpi di clock.

I registri a scorrimento sono utilizzati per memorizzare una sequenza di bit in modo seriale o per ritardarla di un tempo pari al periodo del clock moltiplicato per il numero di flip-flop impiegati.

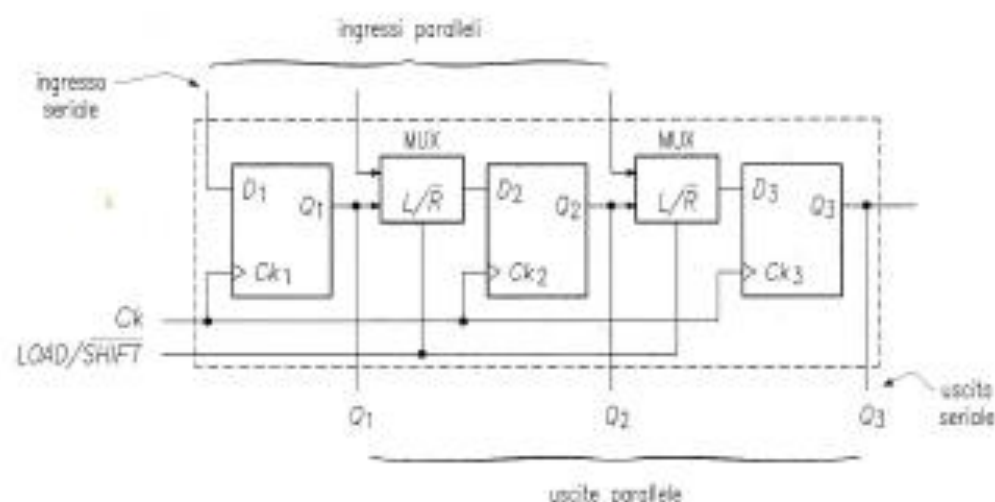
Altre importanti applicazioni dei registri a scorrimento sono le conversioni parallelo-seriale e seriale-parallelo.

La **conversione seriale-parallelo** si effettua caricando in maniera seriale i flip-flop di un registro a scorrimento e poi leggendo in parallelo sulle uscite Q la parola memorizzata.

La **conversione parallelo-seriale** si realizza caricando in parallelo tutti i flip-flop e, mediante successivi impulsi di clock, trasferendo all'uscita seriale i dati memorizzati.

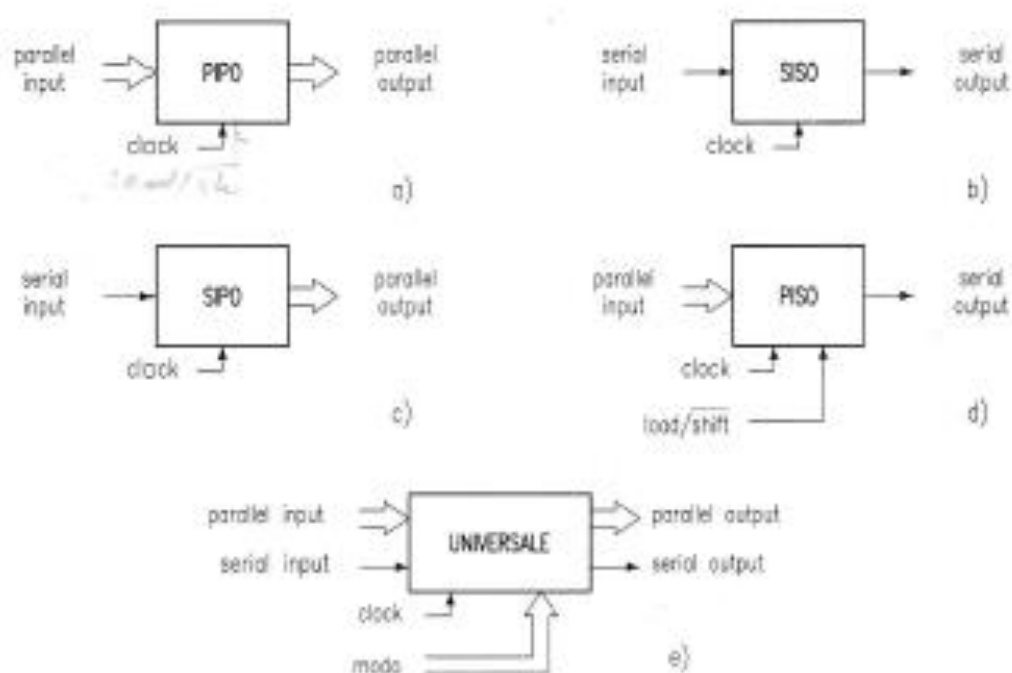
Il registro a scorrimento che realizza queste conversioni, rappresentato nella fig. 7.41, presenta dei multiplexer con i quali è possibile inserire nel flip-flop i dati provenienti dagli ingressi paralleli (*Load*) o dai flip-flop adiacenti (*Shift*), in base al segnale di controllo comune *LOAD / SHIFT*.

Def. Un registro a scorrimento che consente la conversione seriale-parallelo è detto **SIPO** (*Serial Input-Parallel Output*), mentre se effettua la conversione parallelo-seriale è detto **PISO** (*Parallel Input-Serial Output*). Viene definito **registro universale** un registro a scorrimento bidirezionale con ingressi ed uscite seriali e parallele.



Conversione parallelo-seriale e seriale-parallelo con *shift register*. **Fig. 7.41**

Si riportano nella fig. 7.42 gli schemi a blocchi dei registri appena descritti.



Schemi a blocchi dei vari registri. **Fig. 7.42**



Confrontando le modalità di trasferimento dei dati, parallela e seriale, si deduce che quella parallela necessita di un numero maggiore di linee ma risulta più veloce, poiché la trasmissione di un'intera parola avviene in un unico istante; al contrario quella seriale è più lenta, perché sono necessari n periodi di clock, ma è sufficiente una sola linea dati.

In pratica la trasmissione parallela è utilizzata nei collegamenti brevi e veloci, come ad esempio tra un personal computer ed una stampante; la trasmissione seriale viene sfruttata per i collegamenti lunghi, come ad esempio quelli che utilizzano la rete telefonica, in cui pur di risparmiare sul costo della linea si accettano velocità di trasmissione inferiori.

Poiché il computer internamente trasferisce i dati in maniera parallela, il collegamento ad una linea telefonica seriale necessita, fra l'altro, di un convertitore parallelo-seriale in trasmissione e di un convertitore seriale-parallelo in ricezione.

ESEMPIO 7.9

Nel registro a scorrimento rappresentato in fig. 7.38, si determini l'andamento delle uscite dei tre flip-flop, quando all'ingresso D_1 si presenta il segnale di fig. 7.43. Si supponga inizialmente $Q_1 = Q_2 = Q_3 = 0$.

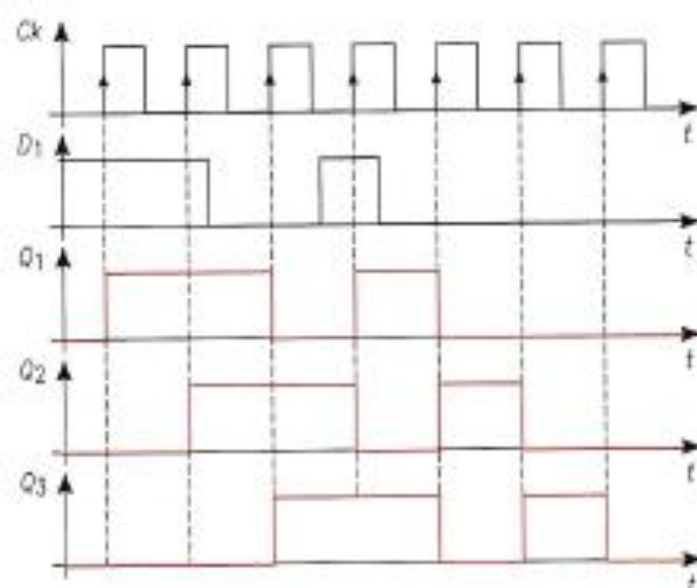


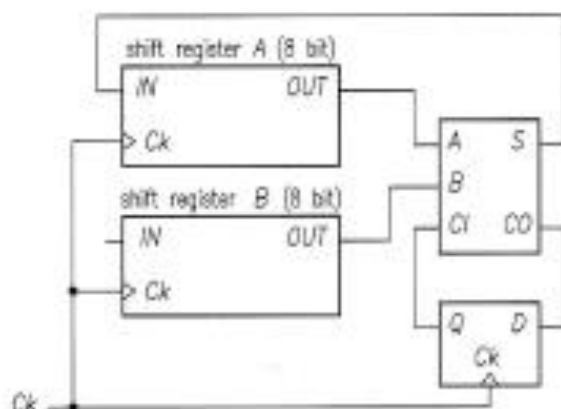
Fig. 7.43

Soluzione

Come si nota nei diagrammi temporali in figura, i segnali sulle tre uscite hanno il medesimo andamento, ritardato di uno, due e tre periodi di clock rispetto al segnale in ingresso al registro.

ESEMPIO 7.10

Si realizzi una rete in grado sommare due numeri binari di otto bit l'uno memorizzati su due registri, utilizzando un unico full-adder.



Sommatore seriale. Fig. 7.44

L'operazione dovrà svolgersi sommando a due a due, in tempi successivi, i bit degli addendi di uguale peso, aggiungendo l'eventuale riporto generato dai bit di peso inferiore.

Il circuito proposto in fig. 7.44 esegue la funzione richiesta nel seguente modo:

- gli addendi sono caricati in parallelo su due shift register.
- Sull'uscita *S* del full-adder è presente la somma dei bit A_0 e B_0 e dell'uscita del flip-flop *D*, preventivamente resettato.
- Sull'uscita *CO* del full-adder è presente il riporto generato dalla somma.
- In corrispondenza del fronte attivo di clock, i bit degli addendi traslano verso destra di una posizione, presentando all'uscita dei registri i bit A_1 e B_1 . Contemporaneamente il risultato della somma di A_0 e B_0 viene immesso nella posizione di sinistra del registro *A*, rimasta ora inutilizzata; inoltre il riporto in uscita (*CO*) viene acquisito dal flip-flop *D* e inviato all'ingresso *Ci* del full-adder, per essere sommato ai bit A_1 e B_1 .
- Dopo otto impulsi di clock la somma è completata ed il risultato è memorizzato nel registro *A*. L'uscita del flip-flop *D* costituisce il bit più significativo del risultato; infatti per rappresentare la somma di due numeri di otto bit sono necessari nove bit.

Questo circuito è noto con il nome di *sommatore seriale*.

ESEMPIO 7.11

Il circuito di fig. 7.45a è detto *scrambler* (mescolatore) e funziona nel modo seguente: la sequenza di bit ricevuta in ingresso ne produce una in uscita modificata secondo una determinata logica. Il ripristino della successione di bit originaria viene effettuato mediante un circuito, avente una struttura analoga allo *scrambler*, detto *descrambler*.

Lo *scrambler* è utilizzato nelle telecomunicazioni con i seguenti scopi:

- eliminare lunghe sequenze di 0 o di 1, che in alcuni casi creano problemi nei circuiti ricevitori;
- criptare il segnale, cioè renderlo indecifrabile a chi non possiede un apposito circuito di decodifica.
- generare sequenze pseudocasuali, cioè senza una logica apparente, in cui il 50% dei bit abbia valore 0 ed il 50% valore 1.

Supponendo il registro caricato inizialmente con i valori $Q_1 = 0$, $Q_2 = 0$, $Q_3 = 1$ e $Q_4 = 0$, ricavare la successione dei bit d'uscita quando all'ingresso dati viene applicata la sequenza: 00001101111.

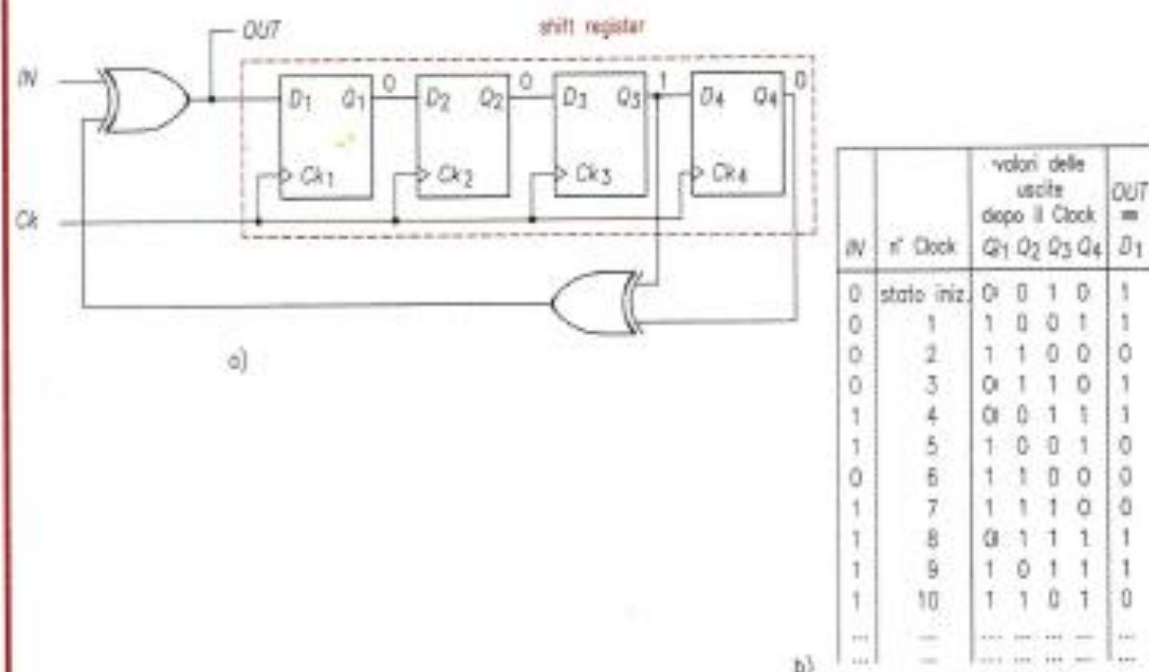


Fig. 7.45 Scrambler.

Soluzione

Il dato in ingresso allo scrambler viene combinato da una porta EXOR con il risultato dell'EXOR tra le uscite 3 e 4 dello shift register. L'uscita dello scrambler coincide con l'ingresso dello shift register.

La tabella nella fig. 7.45b mostra l'andamento delle uscite del registro e dello scrambler in corrispondenza dei successivi fronti di salita del clock, in risposta alla successione dei bit in ingresso.

Si noti che la successione dei bit d'uscita non presenta, come quella d'ingresso, lunghe sequenze di bit uguali.

7.5.3 I registri a scorrimento integrati

Si riportano ora i pin-out e le descrizioni sintetiche di alcuni registri a scorrimento integrati, rimandando al *functional index* nell'indice C per un quadro più completo.

Registro SISO 4006:

questo integrato CMOS contiene due registri a scorrimento a quattro stati e due a cinque stati, il cui collegamento interno è rappresentato nella fig. 7.46a.

Connetteno opportunamente ingressi ed uscite dell'integrato è possibile realizzare shift register con sezioni multiple di 4, 5, 8, o 9 stadi oppure un unico registro a scorrimento costituito da 10, 12, 13, 14, 16, 17 o 18 stadi.

Esistono sul mercato numerosi integrati che vanno sotto il nome di "FIFO Memories"

costituite da un grande numero di celle a cui si accede solo in modo seriale; uno di questi è il 74ACT2235 contenente 1024 parole di 9 bit.

Registro SIPO 74164:

Il registro (fig. 7.46b) possiede due ingressi seriali (*A* e *B*) posti internamente in AND, un ingresso di *CLEAR* asincrono ed un ingresso di *CLOCK* per l'avanzamento verso destra dei dati. Le uscite degli otto flip-flop interni sono tutte disponibili all'esterno ($Q_A + Q_H$).

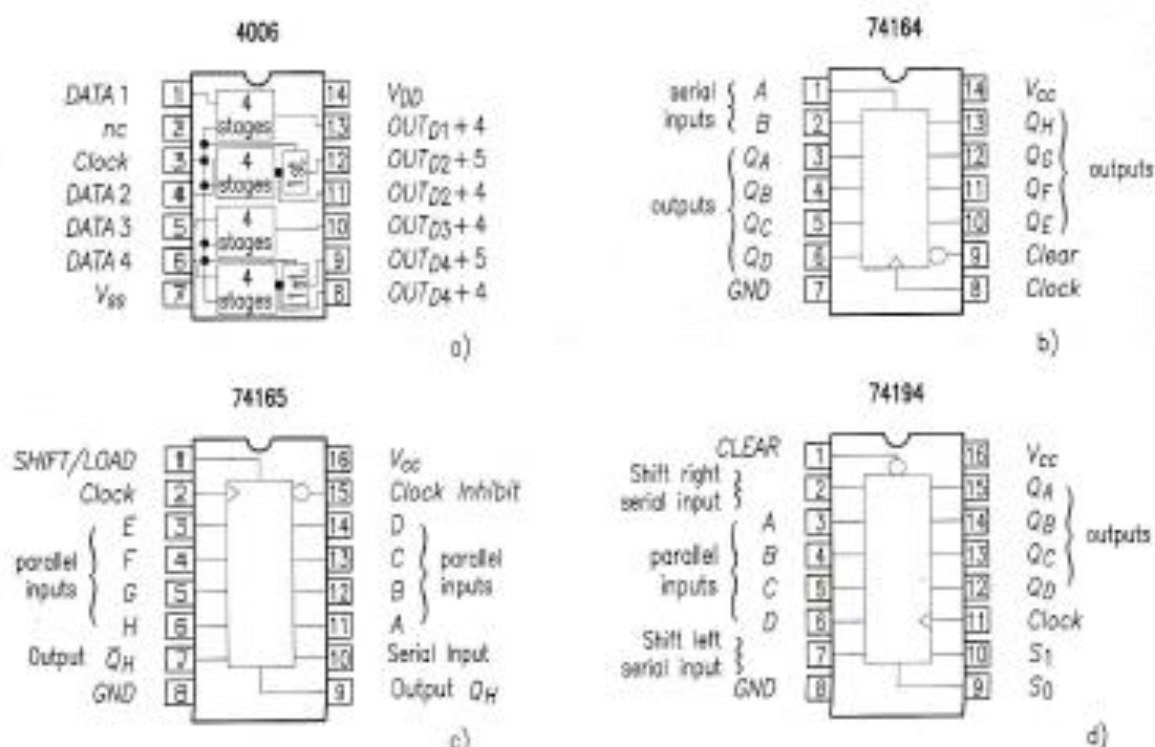
Registro PISO 74165:

questo registro (fig. 7.46c) presenta otto ingressi paralleli (*A+H*) per il caricamento dei dati sui flip-flop, che avviene in maniera asincrona non appena il pin *SHIFT/LOAD* viene portato a livello BASSO. Gli impulsi sul pin *CLOCK*, che possono eventualmente essere inibiti portando BASSO il pin *CLOCK INHIBIT*, provocano lo shift verso destra dei dati, se il pin *SHIFT/LOAD* si trova a livello ALTO. L'uscita seriale è rappresentata dal pin Q_H . È disponibile anche l'ingresso *SERIAL INPUT* per l'inserimento seriale dei dati.

L'integrato 74166 possiede funzioni identiche al 74165, ma con la differenza che il caricamento parallelo avviene in modo sincrono.

Registro universale 74194:

il registro (fig. 7.46d) presenta all'esterno tutti gli ingressi (*A+D*) e le uscite ($Q_A + Q_D$) dei quattro flip-flop interni; è possibile resettare in modo asincrono tutte le uscite portando BASSO il pin *CLEAR*.



Pin-out di shift register integrati: a) 4006; b) 74164; c) 74165; d) 74194.

Fig. 7.46

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 e 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.