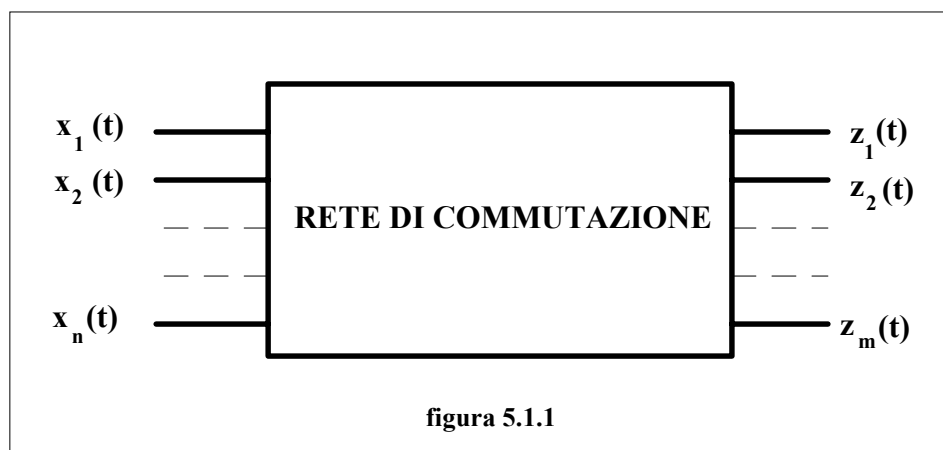


CAPITOLO V I CIRCUITI SEQUENZIALI

5.1) Introduzione ai circuiti sequenziali.

I circuiti considerati fino a questo punto sono stati i circuiti combinatori. Si consideri ora il modello generale di un circuito di commutazione, illustrato in fig. 5.1.1.



Si hanno n ingressi $x(t)$ e m uscite $z(t)$, funzioni del tempo; se in ogni istante il valore attuale delle uscite z è determinato unicamente dal valore degli ingressi si dice che il circuito è di tipo combinatorio. Esso, come si è già visto nei capitoli precedenti, può essere totalmente descritto da un set di equazioni booleane del tipo:

$$z_j = F_j(x_1, x_2, \dots, x_n)$$

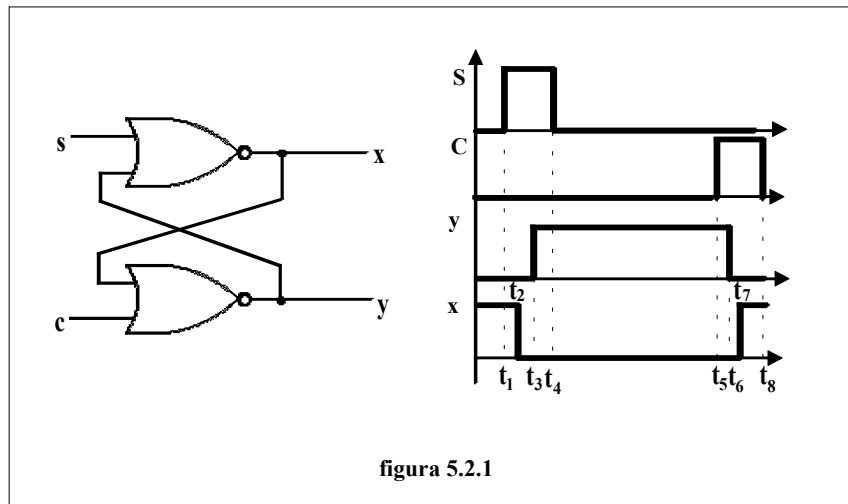
dove la dipendenza dal tempo non compare esplicitamente in quanto si sottointende che i valori presi in considerazione per le n variabili di ingresso si riferiscano allo stesso istante.

Quando invece il valore delle uscite non dipende solamente dal valore presente delle variabili di ingresso, ma anche dalla storia passata del circuito, si parla di circuito sequenziale.

Il sistema telefonico è un tipico esempio di sistema sequenziale; si supponga infatti di essere in procinto di comporre l'ultima cifra di un numero telefonico. Questa ultima cifra è l'ingresso attuale del sistema e l'uscita sarà il segnale che effettuerà il collegamento. Ovviamente l'ingresso attuale non è il solo fattore che determina il collegamento, ma anche le cifre composte precedentemente sono ugualmente importanti. Un altro esempio di sistema sequenziale è il calcolatore elettronico; di solito in esso vengono usati in maniera sequenziale diversi sottoinsiemi che possono essere di volta in volta sequenziali o combinatori.

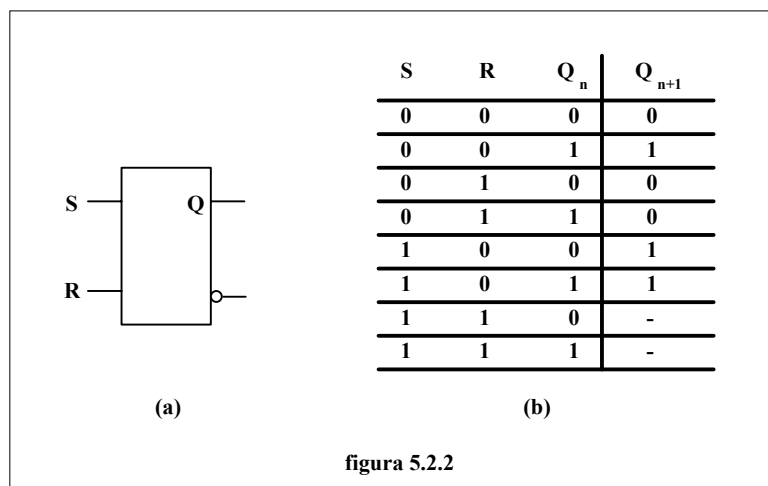
5.2) I flip-flop.

È abbastanza evidente che nei sistemi sequenziali, introdotti al paragrafo precedente, le informazioni sulla storia passata del circuito devono venir memorizzate. Si possono avere diversi tipi di dispositivi di memorizzazione, ma uno dei più comuni e frequentemente usati è il flip-flop. In fig. 5.2.1 è illustrato un flip-flop costruito con due porte NOR ed è riportato anche il diagramma di temporizzazione per una tipica sequenza operativa.



All'istante iniziale ambedue gli ingressi siano 0, l'uscita Y sia 0, quella X sia 1; poiche' le uscite sono riportate agli ingressi delle porte NOR, la condizione ipotizzata e' una condizione stabile. Infatti il NOR 2 ha come ingressi $C=0$ e $X=1$ e quindi la sua uscita e' $Y=0$, mentre il NOR 1 avendo ambedue gli ingressi nulli ha come uscita $X=1$. All'istante t_1 l'ingresso S vada a 1. Gli ingressi del NOR 1 passano percio' da 00 a 01 e dopo un ritardo pari a $t_2 - t_1$ la sua uscita X passa a 0. Questo fatto fa cambiare gli ingressi del NOR 2 da 01 a 00 in modo che dopo un ritardo pari a $t_3 - t_2$ l'uscita Y passa a 1. Non si ha alcun ulteriore effetto, ne' per la commutazione a 11 degli ingressi del NOR 1, ne' per il passaggio di S a 0 all'istante t_4 . La nuova condizione raggiunta e' stabile. Una sequenza analoga, ma inversa, ha luogo quando l'ingresso C passa a 1, riportando il circuito nella condizione iniziale.

Un flip-flop che operi nel modo descritto prende il nome di flip-flop set-reset, viene indicato con il simbolo RS ed e' rappresentato con il simbolo di fig. 5.2.2 (a).



Gli ingressi S e R corrispondono a quelli S e C di fig. 5.2.1, mentre l'uscita Q corrisponde a Y, quella \bar{Q} a X.

Il funzionamento del circuito e' riassunto in modo completo nella tabella di fig. 5.2.2 (b). S, R e Q_n indicano il valore degli ingressi e dell'uscita all'arbitrario istante t_n , mentre Q_{n+1} indica il valore assunto dall'uscita nell'istante immediatamente successivo per effetto degli ingressi applicati all'istante t_n .

Le condizioni d.c.c delle ultime due righe indicano che nel normale funzionamento non e' permesso che ambedue gli ingressi diventino 1 allo stesso

E' immediato verificare che la tabella di fig. 5.2.2 e' retta dall'equazione:

$$Q_{n+1} = S + Q_n \cdot \bar{R} \quad (5.2.1)$$

ottenuta sfruttando le condizioni d.c.c. e imponendo il vincolo

$$S \cdot R = 0 \quad (5.2.2)$$

La ragione per cui i due ingressi non possono essere contemporaneamente 1 e' duplice; in primo luogo perche' essi porterebbero ambedue le uscite a 0 violando la condizione base di funzionamento di un flip-flop secondo la quale le due uscite devono essere sempre complementari; in secondo luogo, se ambedue gli ingressi tornassero a 0 al medesimo istante lo stato in cui il flip-flop si porterebbe non sarebbe prevedibile e al limite potrebbe realizzarsi una condizione di oscillazione.

Con il vincolo (5.2.2) il flip-flop RS realizza invece un dispositivo di memorizzazione affidabile, in cui lo stato delle uscite indica quale dei due ingressi si e' trovato per ultimo al livello 1.

E' abbastanza evidente da questa analisi qualitativa del funzionamento del flip-flop RS che un impulso di ingresso deve avere una durata minima se si vuole che la commutazione avvenga con sicurezza; si supponga infatti che S ritorni a zero prima dell'istante t_3 ; ambedue gli ingressi del NOR 1 si troverebbero allora, almeno momentaneamente, a 0 e tenderebbero a far tornare X a 1. Il verificarsi di tale condizione renderebbe evidentemente incerto il funzionamento del circuito.

La minima durata dell'impulso di ingresso e' il primo di tre vincoli sulle temporizzazioni dei circuiti sequenziali contenenti flip-flop, che verranno esaminati nel seguito.

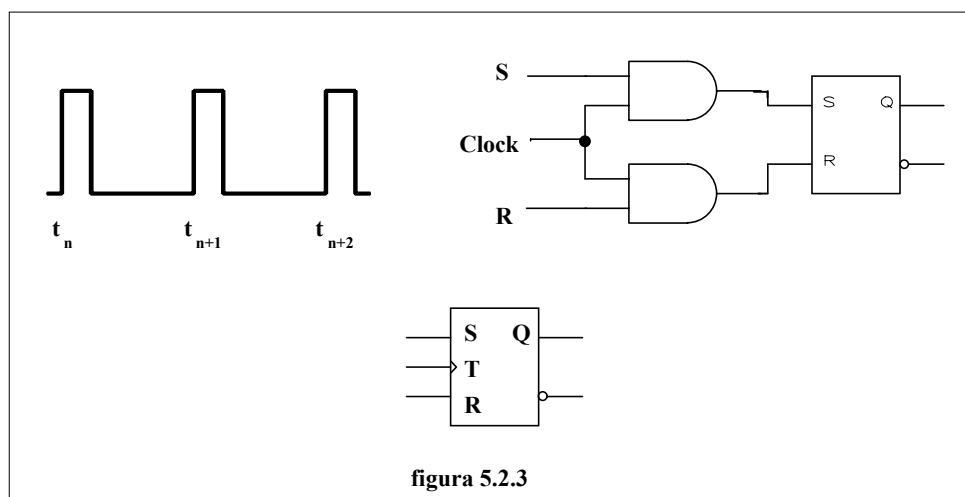
Come si vedra' nei capitoli successivi, il progetto di un circuito sequenziale puo' essere notevolmente semplificato se le commutazioni possono avvenire solo periodicamente in istanti equintervallati. Questo tipo di funzionamento puo' venir assicurato se tutti i cambiamenti di stato vengono sincronizzati da opportuni impulsi di orologio (clock). E' bene chiarire che per impulso si intende un segnale che normalmente si mantiene a un livello, usualmente 0, e va all'altro livello solamente per intervalli di tempo estremamente brevi. In contrapposizione viene chiamato a livelli un segnale che puo' rimanere sia a 0 che a 1 per periodi di tempo indefiniti e comunque molto lunghi se paragonati alla durata di un impulso. La dizione estremamente breve va evidentemente rapportata alla velocita' del circuito, ma normalmente indica una durata dello stesso ordine di grandezza del tempo di ritardo del flip-flop.

Cio' stabilito, si puo' affermare che:

Un circuito sequenziale sincronizzato da un impulso di clock puo' cambiare stato solo in corrispondenza a tale impulso e cambiera' stato non piu' di una volta per ciascun impulso di clock.

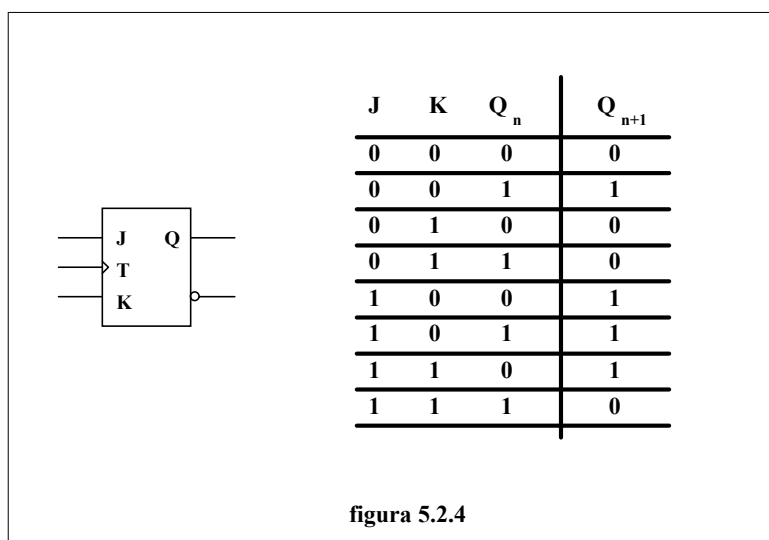
Un circuito che operi nel modo descritto viene detto **circuito sincrono o circuito clock mode**. In fig. 5.2.3 e' illustrata una tipica forma d'onda di clock, lo schema di un flip-flop RS sincrono e il relativo simbolo.

L'unico vincolo necessario per il corretto funzionamento del circuito e' che i segnali S e R non cambino durante l'intera durata dell'impulso di clock.



La tavola di verita' e le equazioni che reggono il funzionamento del flip-flop sono le stesse del flip-flop RS non sincronizzato (detto anche asincrono); in questo caso tuttavia S, R e Q_n indicano i valori presenti durante l'impulso di clock, mentre Q_{n+1} si riferisce al valore assunto dall'uscita immediatamente dopo tale impulso.

Malgrado che il flip-flop RS sincrono sia assolutamente adatto a qualsiasi realizzazione circuitale, esso e' raramente usato in pratica, essendo stato praticamente soppiantato dal tipo JK. In quest'ultimo il vincolo che ambedue gli ingressi non possano essere 1 allo stesso istante viene a cadere. Il relativo simbolo e la tavola di verita' sono riportati in fig. 5.2.4.



Si noti che il flip-flop JK e' identico a quello RS con J corrispondente a S e K a R eccetto quando ambedue gli ingressi valgono 1, caso in cui il flip-flop comunque cambia stato.

L'equazione del flip-flop JK e':

$$Q_{n+1} = \overline{K} \cdot Q_n + J \cdot \overline{Q_n}$$

Il flip-flop JK e' sempre sincronizzato.

Un ulteriore tipo di flip-flop e' il flip-flop T in cui il trigger (clock) e' il solo segnale di ingresso. Ogni volta che un impulso si presenta all'ingresso T il flip-flop cambia stato, in modo che l'equazione che lo regge e' semplicemente:

$$Q_{n+1} = \overline{Q_n}$$

In realta', in termini generali, il tipo T ha due ingressi separati; l'ingresso T e l'ingresso di clock e la definizione del suo funzionamento dovrebbe essere: "all'arrivo dell'impulso di clock il flip-flop commuta solo se l'ingresso T vale 1". Da questo punto di vista e' facile verificare che l'equazione di funzionamento e' $Q_{n+1} = Q_n \cdot \overline{T_n} + \overline{Q_n} \cdot T_n = Q_n \oplus T_n$. Tuttavia molto spesso l'ingresso di clock ed quello T sono connessi assieme e l'equazione di funzionamento si riduce di conseguenza a quella citata piu' sopra.

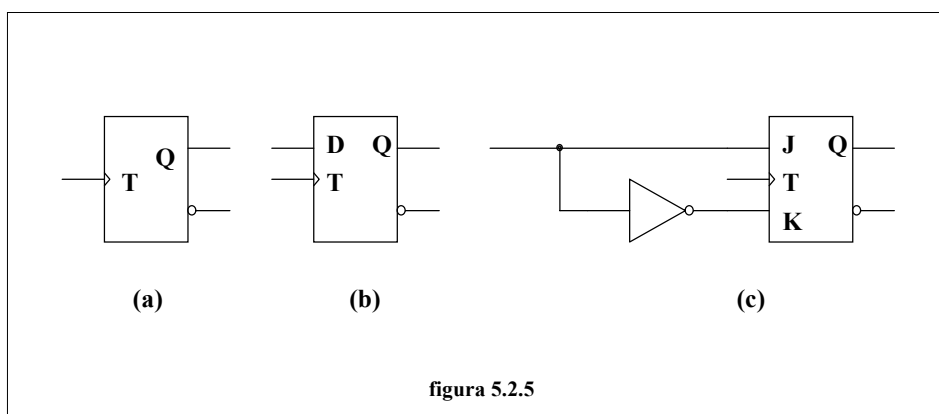
Si noti che un flip-flop JK diviene un flip-flop T se ambedue gli ingressi J e K vengono mantenuti costantemente assieme e utilizzati come ingresso T.

Infine un quarto tipo di flip-flop e' quello di tipo D. In esso l'uscita dopo un impulso di clock e' uguale al valore presente all'ingresso D all'istante di clock.

L'equazione che lo regge e':

$$Q_{n+1} = D_n$$

In fig. 5.2.5 (a) e (b) sono riportati i simboli del flip-flop T e D. In fig. 5.2.5(c) e' invece illustrato in che modo un flip-flop JK possa funzionare come un tipo D.



5.3) Motivi che consigliano la scelta dei circuiti sequenziali.

Nei paragrafi precedenti sono stati presi in esame alcuni sistemi sequenziali, ed e' abbastanza evidente che parecchi sistemi fisici si comportano in tal modo; tra l'altro gli stessi flip-flop appena esaminati sono chiaramente sistemi sequenziali.

Ora quando un progettista si trova a dover affrontare un problema di natura logica la prima domanda che gli si presenta con riferimento all'implementazione e' se usare un circuito di tipo combinatorio o di tipo sequenziale.

A titolo di esempio si consideri il progetto di una macchina per il gioco di quello che comunemente viene chiamato "filetto".

Tale gioco consiste in nove riquadri che i giocatori riempiono alternativamente con un loro simbolo fino a realizzare una sequenza di tre simboli dello stesso tipo allineati. Ogni riquadro quindi e' vuoto oppure contiene l'uno o l'altro simbolo. Le diverse configurazioni che la tabella puo' assumere sono quindi $3^9 = 16683$.

Il gioco debba svolgersi tra uomo e macchina; la macchina dovra' evidentemente consistere in un sistema logico in grado di decidere quale mossa debba essere fatta in risposta alla mossa del giocatore umano.

E' evidente che la progressione del gioco e' sequenziale, ma non e' affatto necessario che il giocatore macchina sia un dispositivo sequenziale. Essa infatti puo' essere un dispositivo combinatorio che riceve come ingressi la situazione della tabella a un dato istante e decide la mossa da fare solamente sulla base di tale informazione. Le specifiche di tale circuito saranno pertanto:

Data qualsiasi tra le 16683 possibili combinazioni di ingresso, scegliere l'uscita piu' opportuna tra le 9 possibili.

Per soddisfare tali specifiche il circuito dovra' evidentemente essere di tipo combinatorio e di grosse dimensioni.

In alternativa si puo' pensare di far si' che il giocatore macchina alteri la sua struttura man mano che il gioco prosegue, tenendo conto del fatto che ad un particolare istante del gioco non si possono piu' presentare tutte le combinazioni .

Si supponga infatti che il giocatore uomo muova per primo; poiche' egli ha solamente nove possibili mosse la macchina puo' partire da una configurazione progettata per rispondere a ciascuna di esse nel modo piu' opportuno. Dopo la mossa della macchina all'uomo sono disponibili solo sette risposte e la configurazione della macchina puo' essere modificata in modo da tenerne conto. Una sequenza di tale tipo si ripete poi ad ogni mossa successiva.

L'approccio descritto, di natura evidentemente sequenziale, consiste in definitiva nello spezzare un unico grosso problema in un certo numero di problemi piu' piccoli. Da un punto di vista circuitale cio' significa costruire un certo numero di circuiti di ridotte dimensioni, ciascuno capace di gestire un numero limitato di situazioni a ciascun passo del gioco, anziche' costruire un unico grosso circuito che possa interpretare tutte le possibili situazioni che si possono presentare. La macchina usera' in sequenza tali circuiti via via che il gioco progredisce.

E' evidente che nell'esempio preso in considerazione il problema e' di per se stesso di natura squisitamente sequenziale. In casi simili un approccio di tipo sequenziale e' di solito piu' economico che non un approccio puramente combinatorio.

Quale altro esempio si consideri un dispositivo che esegua il test di parita' su parole di 8 bit. Se gli 8 bit sono disponibili tutti simultaneamente il problema puo' venir risolto con un circuito combinatorio a 6 livelli la cui funzione di commutazione sia:

$$Z = [(x_1 \oplus x_2) \oplus (x_3 \oplus x_4)] \oplus [(x_5 \oplus x_6) \oplus (x_7 \oplus x_8)]$$

che in una realizzazione di tipo NAND richiede 21 gates.

Se pero' gli 8 bit fanno parte di una parola da trasmettere a grande distanza, l'usare 8 linee, una per bit, e' notevolmente costoso. Si preferisce usare una sola linea su cui inviare in successione gli 8 bit e eventualmente una seconda linea su cui vengono inviati opportuni impulsi di sincronizzazione.

Il circuito di controllo di parita' potrebbe allora essere realizzato memorizzando i bit finche' tutti e 8 non fossero presenti. Questo sarebbe tuttavia un modo di affrontare il problema poco efficiente, richiedendo oltre ai 21 gates gia' visti anche 8 flip-flop.

Ricordando pero' che l'OR esclusivo e' associativo, si puo' scrivere:

$$Z = [((((x_1 \oplus x_2) \oplus x_3) \oplus x_4) \oplus x_5) \oplus x_6] \oplus x_7 \oplus x_8]$$

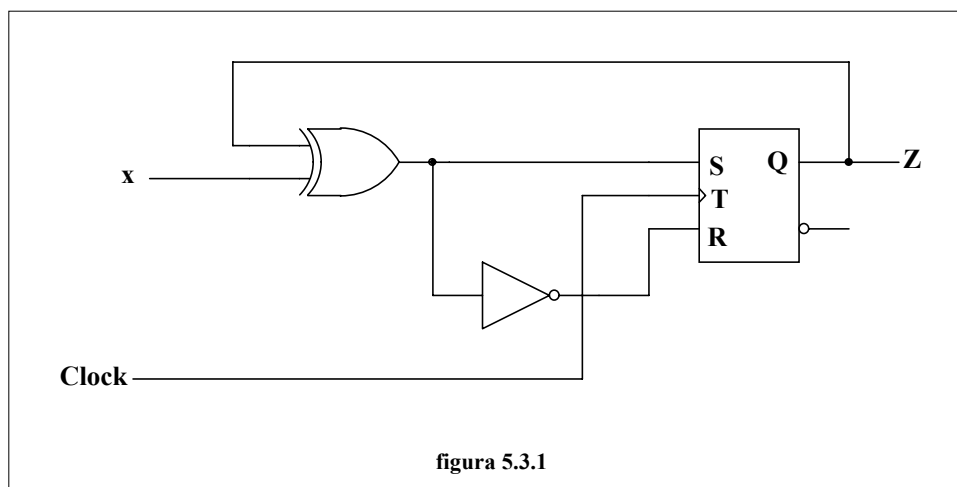
Si possono cioe' testare per la parita' i primi due bit, poi con il risultato il terzo e cosi' via. Queste considerazioni suggeriscono una realizzazione sequenziale. I primi due bit vengono esaminati da una rete che realizza l'OR esclusivo e il risultato memorizzato finche' non arriva il bit successivo. A tale istante il risultato precedente viene confrontato con il bit in arrivo sullo stesso OR esclusivo, il risultato viene memorizzato e cosi' via. In fig. 5.3.1 e' riportato un semplice circuito che realizza la sequenza descritta.

Il flip-flop deve inizialmente essere posizionato a zero in modo che all'arrivo del primo bit x si abbia:

$$x \oplus Z = x \oplus 0 = x$$

Dopo l'arrivo dell'ottavo bit l'uscita Z dara' il desiderato test di parita'. Al limite lo stesso circuito permette di verificare la parita' di parole di qualsivoglia lunghezza.

I due esempi presi in considerazione illustrano un tipo di situazione in cui l'approccio sequenziale e' nettamente preferibile. In ambedue i casi, e soprattutto nel secondo, risulta evidente che il circuito sequenziale e' piu' semplice e di conseguenza piu' economico che non il corrispondente circuito combinatorio.



5.4) Considerazioni sul grado di serializzazione, costo e velocita'.

Situazioni quali quelle descritte al paragrafo precedente, in cui le informazioni da elaborare giungono in maniera sequenziale, conducono spontaneamente alla scelta di circuiti sequenziali. Poiche' essi di solito usano le stesse parti circuitali su porzioni successive di informazione e' possibile in genere ottenere una notevole economia per quanto riguarda la realizzazione circuitale.

Se invece tutte le informazioni necessarie sono disponibili simultaneamente, e' abbastanza naturale che anche la loro elaborazione sia condotta in parallelo.

Anche in questo caso d'altra parte, se la natura del problema e' tale da poterlo spezzare in parti il piu' possibili simili tra loro, puo' essere vantaggioso serializzare deliberatamente il problema per ottenere vantaggi di ordine economico. Il grado di serializzazione dipendera' ovviamente da un compromesso tra costo e velocita' operativa.

Vi e' in effetti un gran numero di fattori complessi che determinano il grado di serializzazione piu' opportuno, alcuni dei quali possono essere illustrati dall'esempio che segue.

Si supponga di dover leggere una parola di memoria in un registro da 32 bit e di dover eseguire su di essa il controllo di parita'. Poiche' tutti i bit sono presenti al medesimo istante, sembrerebbe logico usare un circuito combinatorio. Tuttavia il piu' economico circuito realizzabile richiederebbe una struttura a 10 livelli con 93 gates.

Il circuito di fig. 5.3.1 puo' invece eseguire il test di parita' su un qualsiasi numero di bit, e quindi anche su 32, purché essi siano presentati in forma sequenziale. Un circuito in grado di soddisfare i requisiti richiesti e' quello di fig. 5.4.1. Gli impulsi di orologio pilotano un contatore binario da 5 bit, la cui uscita viene decodificata con un selettore. Il decodificatore a sua volta produce un impulso, uno per ciascun periodo di clock, che mediante delle porte AND presenta uno alla volta i 32 bit all'ingresso del circuito di verifica della parita'.

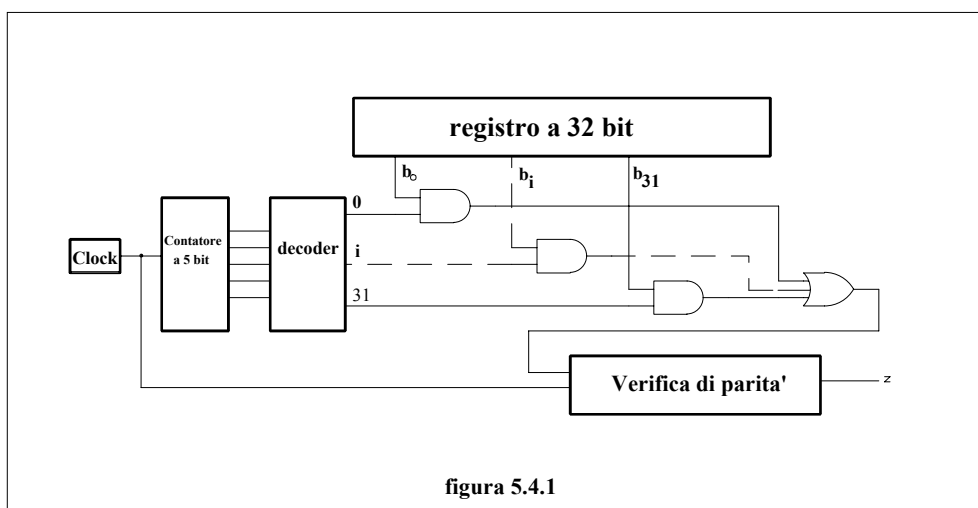


figura 5.4.1

Usando il circuito piu' semplice possibile per contatore e decodificatore il circuito descritto richiede 6 flip-flop e 72 gates. La realizzazione combinatoria e quella sequenziale sono quindi di costo comparabile. Tuttavia in un calcolatore il contatore e il decodificatore possono essere usati per altri scopi e quindi il loro costo grava solo in parte sul circuito per il test di parita'.

Ne discende l'ovvio principio che la valutazione di qualsiasi circuito va fatta in riferimento all'intero sistema. Una valutazione dei costi deve evidentemente tener conto di ogni

possibile utilizzo del circuito in rapporto al sistema; allo stesso modo ogni considerazione sulla velocita' operativa deve essere messa in relazione con la velocita' del sistema nel suo complesso.

Il circuito combinatorio dell'esempio preso in considerazione e' approssimativamente 32 volte piu' veloce del corrispondente circuito sequenziale, ma poiche' in generale il controllo di parita' viene eseguito contemporaneamente a molte altre operazioni, la velocita' complessiva non viene di molto aumentata anche se si sceglie la soluzione totalmente combinatoria.

Si consideri allora in generale un dispositivo o un sistema che fornisca n unita' di informazione allo stesso istante. Queste unita' siano bit, parole, caratteri o qualsiasi altro raggruppamento opportuno.

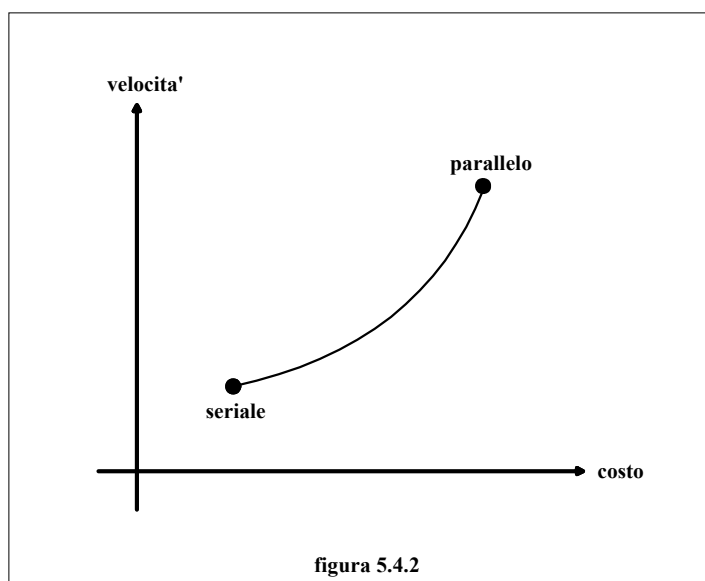
In un sistema completamente parallelo vi e' un circuito (o sottosistema) per ciascuna delle n unita' di informazione e tutte le n unita' vengono applicate agli n circuiti simultaneamente. Si noti che causa delle differenze intrinseche dei vari circuiti o della dipendenza logica uno dall'altro, questi non terminano le loro operazioni contemporaneamente, sebbene le inizino nel medesimo istante.

In un sistema completamente seriale le n unita' di informazione vengono applicate sempre allo stesso circuito, ma in tempi successivi. Supponendo che tutti gli elementi circuitali siano della stessa velocita' e dello stesso costo, il sistema completamente parallelo rappresenta la soluzione piu' veloce, ma anche la piu' costosa, mentre quello completamente seriale e' la soluzione piu' economica, ma anche la piu' lenta. Generalmente e' possibile scegliere una soluzione che si piazzia tra questi due estremi.

E' possibile, ad esempio, dividere le n unita' di informazione in due gruppi uguali, elaborare il primo gruppo in un primo intervallo di tempo e solo successivamente il secondo gruppo.

Si otterra' in tal modo una realizzazione generalmente meno costosa che non quella parallela, ma anche una realizzazione operante a meta' velocita'.

Diagrammando il costo in funzione della velocita' si otterra' una curva del tipo di quella illustrata in fig. 5.4.2.

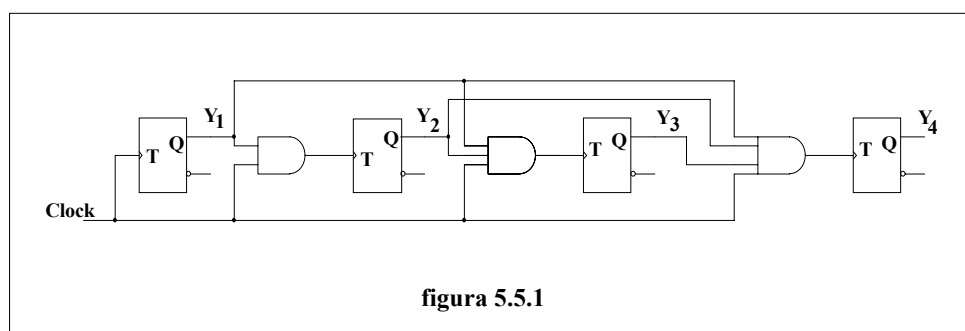


A partire da una realizzazione totalmente parallela si avra' una continua diminuzione di costo e di velocita' via via che si procede verso quella totalmente seriale. Molto spesso tuttavia

la curva raggiunge il suo minimo prima di essere nella condizione di serializzazione totale, in quanto il costo della serializzazione stessa annulla in pratica i benefici ottenuti con la diminuzione del numero di circuiti necessari.

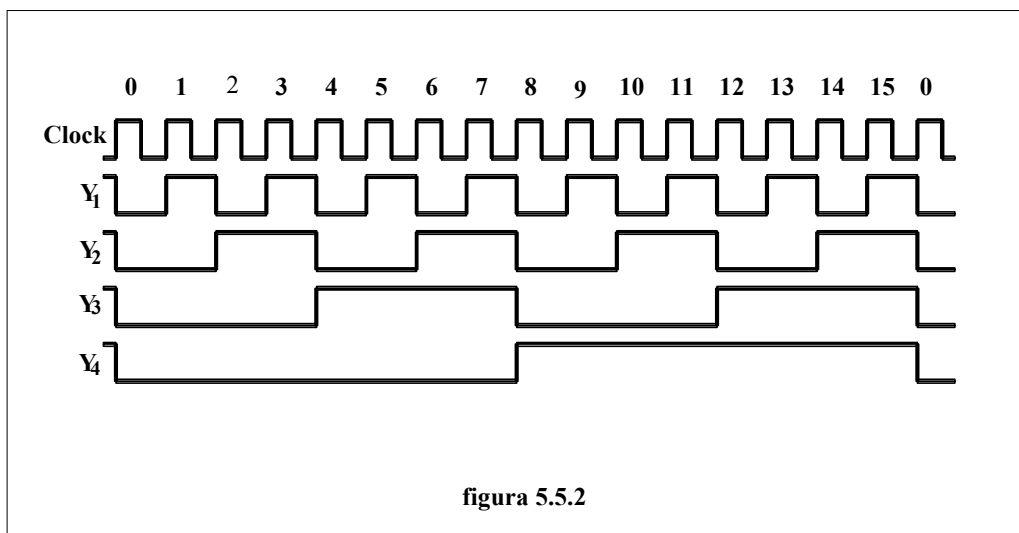
5.5) Registri a scorrimento e contatori.

In fig. 5.3.1 e' stato illustrato un circuito sequenziale che impiega un unico flip-flop. Nelle applicazioni pratiche avviene spesso che vi sia la necessita' di memorizzare piu' di un bit di informazione; in generale la quantita' di informazione che un circuito puo' memorizzare e' direttamente legata al numero di flip-flop presenti nel circuito stesso. In particolare i registri a scorrimento e i contatori presentano molto chiaramente una corrispondenza uno a uno tra capacita' di memorizzazione dell'informazione e il numero di flip-flop.



Sebbene qualsiasi tipo di flip-flop possa essere usato per realizzare un contatore, in pratica sono usati solo contatori costruiti con flip-flop JK o D.

In fig. 5.5.1 e' riportato lo schema di un contatore modulo 16 mentre in fig. 5.5.2 si ha il relativo diagramma di temporizzazione.



Con la dizione "**modulo 16**" si intende che il contatore puo' contare da 0 a 15, mentre con l'impulso successivo di clock ritorna a 0.

Il funzionamento del circuito e' intuitivo. Si noti comunque che il contatore, oltre ad effettuare la funzione di conteggio, memorizza il risultato tra due successivi impulsi di clock.

In generale qualsiasi insieme di flip-flop usato per memorizzare un dato numero di bit di informazione viene chiamato "registro". Un contatore quindi puo' essere chiamato anche registro di conteggio.

I registri vengono spesso utilizzati per memorizzare e ricostruire in forma parallela messaggi in arrivo da una sorgente seriale.

In fig. 5.5.3 e' illustrato un registro a scorrimento, di funzionamento intuitivo.

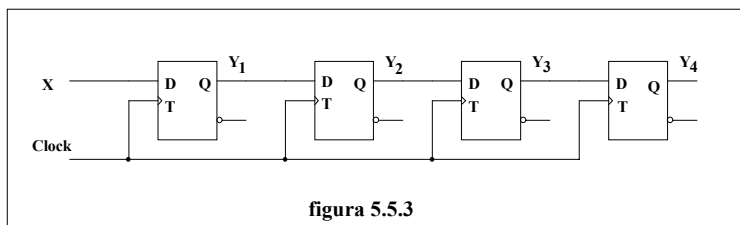


figura 5.5.3

5.6) Un modello generale per i circuiti sequenziali.

Da quanto visto, anche attraverso gli esempi, nei paragrafi precedenti, un circuito sequenziale funziona correttamente se e' in grado di conservare in qualche organo di memoria le informazioni relative agli ingressi passati. Si definisca pertanto stato di un circuito sequenziale all'istante t la configurazione della sua memoria in quell'istante. Due identiche configurazioni di ingresso potranno dare uscite diverse se coesisteranno con due diversi stati del circuito. Le uscite sono cioe' funzione sia dell'ingresso che dello stato. Si puo' allora realizzare un modello generale di circuito sequenziale secondo quanto illustrato in fig. 5.6.1

Le informazioni sugli eventi passati sono memorizzate nelle r variabili y, che prendono il nome di variabili di stato o variabili interne. Ciascuna delle 2^r combinazioni delle variabili y definisce uno stato della memoria e in generale ciascun stato corrisponde ad una particolare combinazione di eventi passati. La sezione logica combinatoria riceve quali ingressi le r variabili di stato y e gli n ingressi del circuito x. Essa genera m uscite z e p variabili di eccitazione y'. Le variabili di eccitazione sono spesso in numero uguale alle variabili di stato, ma in generale il loro numero dipende dal tipo di memoria usato.

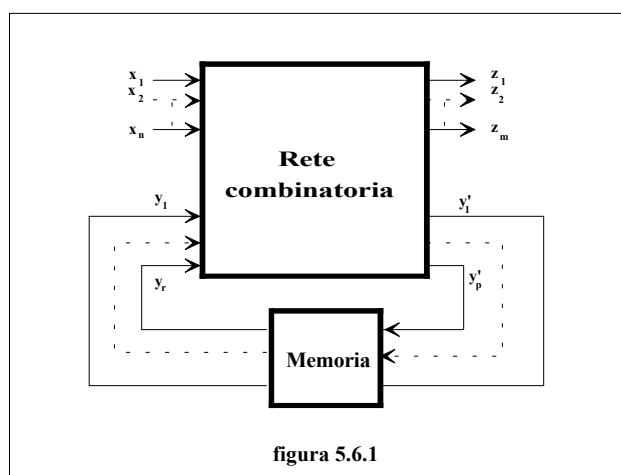


figura 5.6.1

La rete combinatoria puo' evidentemente essere definita dalle equazioni booleane

$$z_i = f_i(x_1, x_2, \dots, x_n, y_1, y_2, \dots, y_r) \quad i = 1, 2, \dots, m \quad (5.6.1)$$

$$y'_i = g_i(x_1, x_2, \dots, x_n, y_1, y_2, \dots, y_r) \quad i = 1, 2, \dots, p$$

Le prime vengono chiamate **equazioni di uscita**, mentre le seconde sono le **equazioni di stato o equazioni interne**. Esse sono evidentemente indipendenti dal tempo, cioè valgono per tutto il tempo in cui gli ingressi e le uscite si mantengono stabili.

Non è possibile specificare le equazioni che legano le y' alle y in quanto dipendono dal tipo di memoria usato, che può essere qualsiasi, come ad esempio una matrice di elementi di memoria a ferrite, dei flip-flop o ancora semplicemente un insieme di anelli di reazione in cui siano inseriti degli elementi di ritardo.

In quest'ultimo caso evidentemente $p=r$ e vale:

$$y(t+\Delta) = y'(t)$$

con Δ pari al tempo di ritardo.

Comunque sia, la combinazione delle variabili y che si avrà al tempo $t+\Delta$ per effetto delle y' presenti all'ingresso della memoria all'istante t , prende il nome di stato futuro.

5.7) Funzionamento sincrono e a impulsi del modello fondamentale.

Un modello per il funzionamento sincrono di un circuito sequenziale si ottiene quando si assume che la memoria del modello fondamentale permetta il trasferimento dell'informazione dall'ingresso all'uscita solamente in corrispondenza ad un segnale di sincronizzazione. Un circuito di tal tipo viene detto "**sincrono**" o **clock mode**.

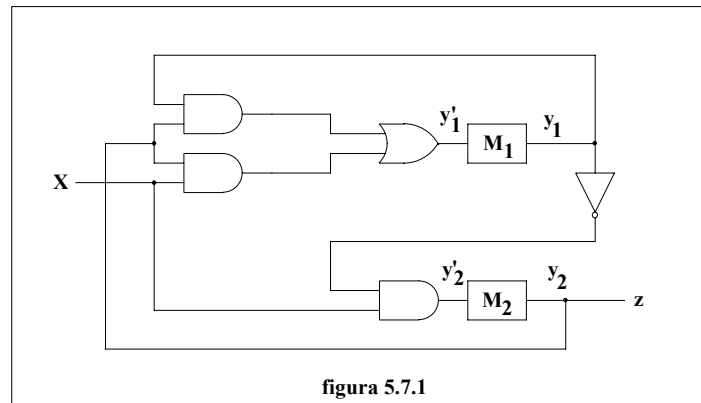
In pratica i circuiti sequenziali sincroni si realizzano utilizzando quali elementi di memoria i flip-flop, di qualunque tipo essi siano, purché sincronizzati da un clock. Al limite, per quanto riguarda il modello fondamentale di circuito sequenziale sincrono, la realizzazione della rete di memoria non ha importanza. È sufficiente che lo stato e gli ingressi possano variare solo in istanti equintervallati.

Tutti i segnali e le uscite della rete combinatoria sono segnali a livelli, segnali cioè che possono assumere uno dei due livelli logici per periodi di tempo arbitrari. Le uscite e le variabili di eccitazione, essendo la rete combinatoria indipendente dal tempo, possono variare in qualsiasi istante in risposta ad una variazione dell'ingresso. Le variabili di stato tuttavia possono variare solamente quando la memoria è abilitata dal segnale di sincronizzazione.

È evidente che il segnale di sincronizzazione non contiene informazione su come debba variare lo stato, ma semplicemente temporizza il cambiamento. Tutta l'informazione è contenuta nelle variabili di eccitazione.

Gli istanti di clock vengono indicati con t_n ($n = 1, 2, 3, \dots$). All'istante generico t_n le variabili di eccitazione devono essere stabili e dipendono dai valori correnti delle variabili x_i e y_i . L'impulso di clock, che arriva all'istante t_n , causa la transizione delle variabili di stato da y^n a $y^{n+1} = f(y^n)$ ponendo il circuito in un nuovo stato. Le variabili di uscita e di eccitazione a loro volta commutano, sia per la variazione dello stato che per un'eventuale variazione di ingresso e devono raggiungere dei nuovi livelli stabili prima che pervenga il prossimo impulso di clock.

Per chiarire quanto illustrato si esamini il circuito di fig. 5.7.1, in cui gli elementi di memoria siano sincronizzati da impulsi distanti tra di loro Δ .



Le equazioni di uscita e di stato sono rispettivamente

$$\begin{aligned} z &= y_2 \\ y_1' &= y_1 \cdot y_2 + x \cdot y_2 \\ y_2' &= x \cdot \overline{y_1} \end{aligned} \quad (5.7.1)$$

Ponendo $y_1(t_0) = y_2(t_0) = 0$ si ricava dalle (5.7.1) che in corrispondenza a $x = 0$ si ha:

$$y_1'(t_0) = y_2'(t_0) = 0$$

che sono i valori che le variabili di stato assumono all'istante $t_1 = t_0 + \Delta$.

Se in t_1 x passa da 0 a 1 si ha:

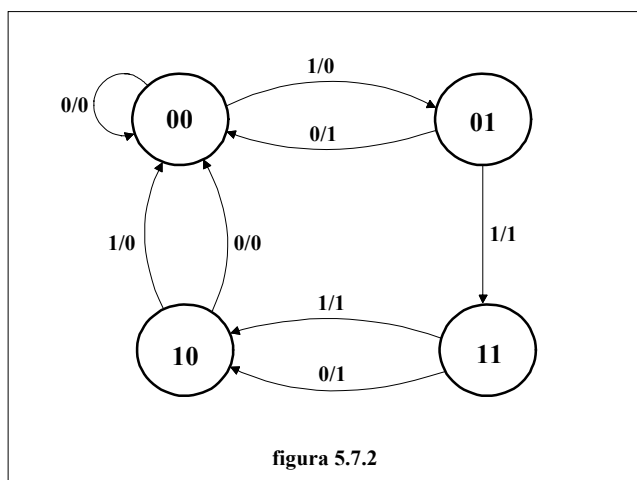
$$y_1'(t_1) = 0 \qquad y_2'(t_1) = 1$$

Ragionando in maniera analoga si ottengono i risultati raccolti nella tabella 1, dalla quale e' immediato ricavare l'evoluzione del circuito in corrispondenza di una determinata sequenza di ingresso applicata ad uno stato iniziale prefissato.

TABELLA 1

ingresso	stato a t	stato a t + Δ	Uscita
0	00	00	0
0	01	00	1
0	10	00	0
0	11	10	1
1	00	01	0
1	01	11	1
1	10	00	0
1	11	10	1

Gli stessi risultati possono essere chiaramente visualizzati nel diagramma di stato di fig. 5.7.2, con tanti nodi quanti sono i possibili stati del circuito.



E' evidente che il diagramma di stato realizzato e' formalmente identico a quello usato per rappresentare le macchine sequenziali nel modello di Mealy.

Estremamente simile al funzionamento dei circuiti sequenziali sincroni e' quello dei circuiti sequenziali ad impulsi. Essi si distinguono dai precedenti per l'assenza di una linea di sincronizzazione separata, ma somigliano ad essi per il fatto che ciascun cambiamento di stato puo' avvenire solamente in corrispondenza ad un impulso di ingresso. Gli impulsi devono essere di sufficiente durata e il circuito deve avere il tempo di stabilizzarsi prima che un secondo impulso possa presentarsi all'ingresso.

5.8) Problemi di temporizzazione e flip-flop master-slave.

Al paragrafo 5.2 e' stato fatto notare a proposito dei flip-flop che il clock deve avere una durata minima per assicurare un corretto funzionamento. Questa tuttavia non e' la sola specifica da rispettare. Una prima ovvia considerazione, che si applica sia ai circuiti logici combinatori che a quelli sequenziali, e' che l'intervallo tra due successive variazioni della stessa variabile deve essere non inferiore al tempo di ritardo del circuito logico. Se, ad esempio, il tempo di ritardo di una porta logica fosse di 10 nsec., essa non potrebbe certamente rispondere a variazioni del segnale che distino 5 nsec.

Esiste inoltre tutta una serie di altri problemi che si presenta anche nei flip-flop sincroni.

Con riferimento alla fig. 5.2.1 ed ai relativi diagrammi di temporizzazione, si supponga che il segnale S ritorni a 0 prima dell'istante t_3 , in cui commuta l'uscita Y. E' evidente che il flip-flop in questo caso non e' correttamente pilotato. Cio' porta a concludere che l'impulso di comando deve avere una durata minima, di solito specificata dal costruttore del dispositivo, determinata principalmente anche se non unicamente dal ritardo dei gates.

Per lo stesso motivo due impulsi non possono succedersi troppo rapidamente imponendo in tal modo un limite superiore alla massima frequenza di clock.

A causa di fattori, troppo complessi e di scarso interesse per venir esaminati in questa sede, il minimo intervallo tra due impulsi successivi e' generalmente molto maggiore che non la durata minima del singolo impulso.

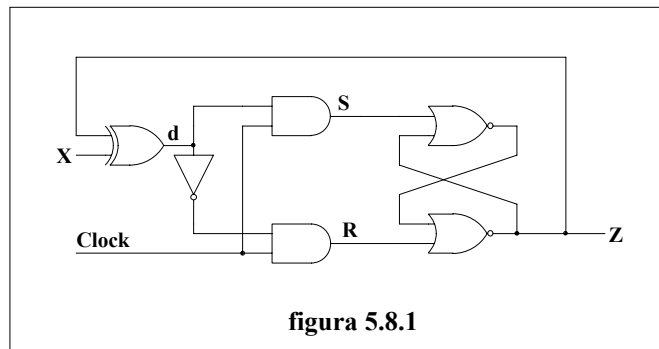
I vincoli sulla durata dell'impulso e sulla massima frequenza di ripetizione del clock si applicano a tutti i flip-flop senza alcun riferimento a come essi vengano usati.

Quando pero' il flip-flop viene usato in un circuito sincrono sorgono altri problemi. Si e' gia' detto che l'arrivo di un impulso di clock fa commutare lo stato in accordo con i valori delle variabili di eccitazione. Si supponga ora che la commutazione delle uscite dei flip-flop

della memoria si propaghi attraverso la rete combinatoria in modo da causare una nuova variazione delle variabili di eccitazione mentre l'impulso di clock e' ancora presente. I flip-flop possono allora commutare nuovamente, portando il circuito in uno stato diverso e la cosa puo' continuare finche' dura l'impulso di clock.

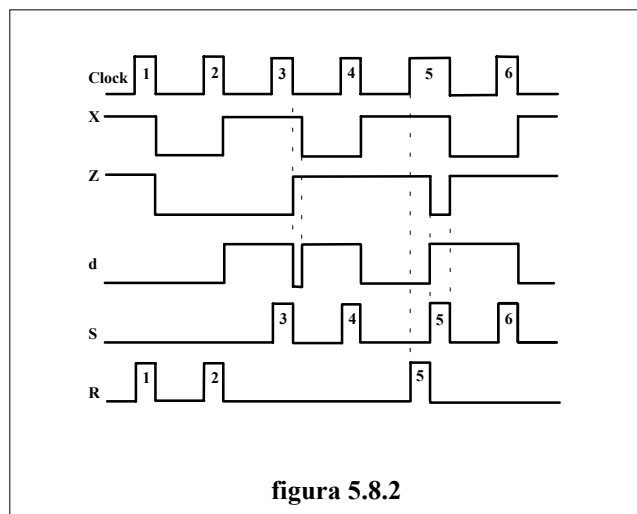
Un tal modo di funzionare sarebbe in contrasto con l'ipotesi fatta inizialmente per i circuiti sincroni, cioe' che ciascun impulso di clock dia luogo ad un'unica variazione dello stato.

Problema del tutto analogo si ha nei circuiti ad impulsi, quando l'impulso di ingresso e' eccessivamente lungo. A titolo di esempio si consideri il circuito di divisione di frequenza illustrato in fig. 5.8.1.



L'ingresso x sia un'onda quadra con frequenza pari alla meta' di quella di clock. L'uscita z cambiera' valore ogni volta che l'ingresso e' 1 in corrispondenza all'impulso di clock. Si ottiene in tal modo un'uscita con frequenza pari alla meta' della frequenza di ingresso e a un quarto di quella di clock.

Si consideri ora la risposta del circuito al terzo impulso di clock. All'inizio di tale impulso (fig. 5.8.2) x e z sono rimasti stabili per un sufficiente periodo di tempo affinche' d si sia stabilizzato a 1.



L'impulso di clock viene quindi trasferito alla linea S e fa si' che dopo un certo ritardo l'uscita z passi a 1.

In tale istante x sia ancora a 1 e quindi d passa a 0 fintantoche' anche x non va a 0. Si ha quindi un breve impulso verso il valore logico 0 sulla linea d, ma tale fatto non causa alcun

problema in quanto l'impulso di clock e' terminato prima che la variazione di d possa propagarsi.

Si consideri ora la situazione in corrispondenza dell'impulso 5, che si suppone essere considerevolmente piu' lungo degli altri.

All'inizio di questo impulso d e' 0; l'impulso di clock viene trasferito alla linea R e con un certo ritardo z passa a 0.

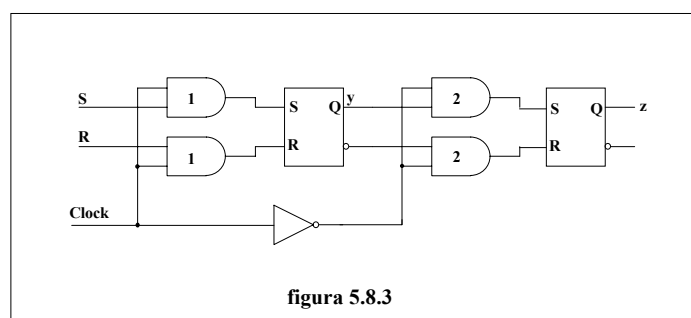
Poiche' x e' ancora a 1 d commuta a 1 e poiche' in tale istante il clock e' ancora presente, R va a 0 e S a 1 causando un'ulteriore commutazione di z. Si hanno cioe' due commutazioni dell'uscita, di cui la seconda indesiderata.

Come in questo esempio, in tutti i circuiti che usino flip-flop sincroni esiste quindi un vincolo non solo sulla durata minima dell'impulso di clock, ma anche su quella massima.

Disgraziatamente la tecnologia costruttiva dei circuiti integrati porta ad una notevole dispersione del valore dei ritardi dei singoli elementi e non e' quindi possibile basarsi su un'accurata scelta della durata dell'impulso di clock per assicurare un corretto funzionamento al circuito.

Gli inconvenienti esaminati possono essere evitati ricorrendo a dei flip-flop dal circuito piu' raffinato, detti flip-flop master-slave.

In fig. 5.8.3 e' illustrato lo schema di principio di un flip-flop RS master-slave.



In assenza di segnali ai terminali di ingresso, l'uscita del flip-flop 1 e di quello 2 sono uguali. Quando viceversa e' presente un segnale ai terminali di ingresso, durante l'intervallo in cui il clock e' 1 il flip-flop 1 si porta nello stato che gli compete in funzione degli ingressi, mentre il flip-flop 2 non modifica il suo stato essendo i gates 2 chiusi.

Successivamente, durante l'intervallo in cui il clock e' 0, il flip-flop 1 viene isolato dagli ingressi, chiudendo le porte 1, mentre l'apertura delle porte 2 trasferisce l'informazione all'uscita z.

Considerando ora un circuito sequenziale sincrono che contenga solo flip-flop master-slave e' evidente che la variazione di stato e quindi l'eventuale modifica delle variabili di eccitazione avviene quando le porte 1 di ingresso del flip-flop sono chiuse. Quindi per ciascun impulso di clock si puo' avere un'unica variazione di stato, indipendentemente dalla durata dell'impulso di clock stesso. I cambiamenti di stato si hanno evidentemente in corrispondenza del fronte di discesa dell'impulso di clock.

Un secondo tipo di flip-flop che permette il cambiamento di stato in corrispondenza al fronte di salita dell'impulso di clock, e che permette anch'esso di evitare gli inconvenienti messi in luce in questo paragrafo, e' il flip-flop "edge triggered" che si basa su altri principi di funzionamento.

Si noti tuttavia che, anche usando tali tipi di flip-flop, quando il clock di uno o piu' di essi e' spostato rispetto al clock degli altri flip-flop, puo' ugualmente verificarsi una tran-

sizione multipla di stato; tale fenomeno e' conosciuto sotto il nome di "clock skew" e puo' verificarsi con una certa facilità nei sistemi estremamente veloci, in cui i ritardi di trasmissione nelle interconnessioni tra gli elementi circuitali cominciano ad assumere notevole importanza. Tale fenomeno verra' preso in considerazione con maggior dettaglio nel seguito.

5.9) Funzionamento asincrono del modello fondamentale.

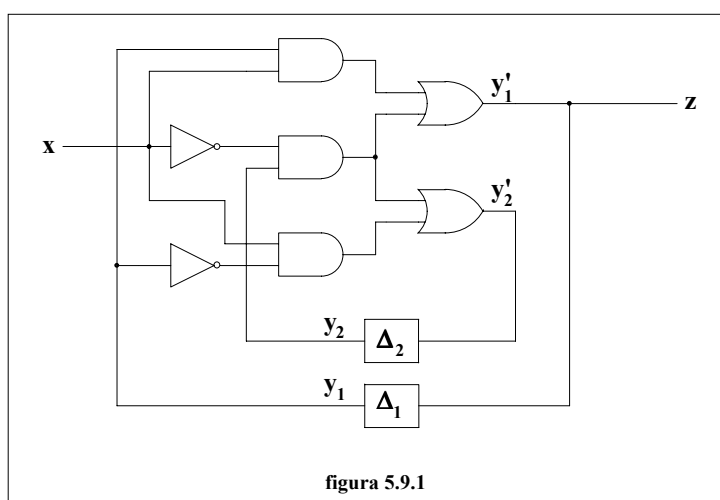
Sia il funzionamento sincrono a livelli che quello ad impulsi, già descritti ai paragrafi precedenti, richiedono che la transizione di stato avvenga solo in coincidenza di ben precisi istanti, cioè tramite un'opportuna temporizzazione. Vengono quindi imposti dei vincoli alla struttura del circuito e alla velocità cui esso puo' funzionare.

Si puo' pero' pensare di far funzionare il modello fondamentale eliminando tali vincoli; il funzionamento cioè avviene sempre secondo le (5.6.1), ma senza alcuna sincronizzazione. La memoria in tal caso e' data da semplici elementi di ritardo sulla catena di reazione.

Un circuito di tal tipo viene detto "**asincrono**" o **level mode**. Se si impone inoltre che i segnali di ingresso varino uno solo alla volta e che dette variazioni non possano avvenire se non quando i ritardi sono inattivi si hanno notevoli semplificazioni circuitali; si parla in tal caso di funzionamento in **modo fondamentale**. E' intuitivo che la variazione simultanea di piu' di un ingresso puo' causare problemi, poiché in realtà gli eventi non possono mai essere contemporanei. L'effettiva risposta del circuito dipende quindi dalla sequenza, generalmente non conosciuta, con cui le variazioni si presentano.

Si puo' notare che il circuito asincrono e' il piu' generale tipo di circuito sequenziale, nel senso che puo' rispondere a qualsiasi tipo di segnale, sia esso a impulsi o a livelli. Infatti la distinzione tra impulso e livello e' una pura questione di definizione; fisicamente un impulso puo' essere considerato costituito da due successive variazioni di un segnale a livelli che avvengono a breve distanza di tempo. Differente e' invece l'interpretazione; in un circuito a impulsi l'ingresso impulsivo viene visto come un singolo evento che da' luogo ad un unico cambiamento di stato. In un circuito asincrono un impulso viene considerato come due eventi distinti, ciascuno dei quali puo' causare una variazione di stato.

Per illustrare il funzionamento asincrono del modello fondamentale si consideri il circuito di fig. 5.9.1.



Esso corrisponde perfettamente al modello fondamentale, quando in Δ_1 e Δ_2 si considerino concentrati tutti i ritardi della parte combinatoria del circuito.

Si supponga che a partire dalla situazione iniziale

$$x = y_1 = y_2 = y'_1 = y'_2 = 0$$

tutte le variazioni dell'ingresso avvengano a intervalli Δ , con

$$\Delta = \max(\Delta_1, \Delta_2)$$

e che x rimanga costante ogni volta che $y \neq y'$. Si supponga cioe' che il circuito lavori in modo fondamentale. Le equazioni del circuito sono:

$$\begin{aligned} z &= y'_1 \\ y'_1 &= x \cdot y_1 + \bar{x} \cdot y_2 \\ y'_2 &= x \cdot \bar{y}_1 + \bar{x} \cdot y_2 \end{aligned}$$

L'evoluzione del circuito e' riportata nella tabella 2.

Le righe di tale tabella possono essere divise in due categorie; quelle in cui $y_i \neq y'_i$, che corrispondono ai periodi transitori, e quelle segnate con l'asterisco in cui $y_i = y'_i$, corrispondenti agli stati in cui il circuito permane finche' non viene cambiato l'ingresso.

TABELLA 2

	Ingresso	stato a t	stato a t + Δ	Uscita
*	0	00	00	0
	1	00	01	0
*	1	01	01	0
	0	01	11	1
*	0	11	11	1
	1	11	10	1
*	1	10	10	1
	0	10	00	0

Anche in questo caso si puo' costruire il diagramma degli stati, ma e' piu' conveniente usare il modello di Moore, compatibile con le ipotesi fatte per il funzionamento in modo fondamentale. Tutti gli stati devono evidentemente essere stabili e con riferimento al circuito di fig. 5.9.1 si ha il diagramma degli stati di fig. 5.9.2.

