

RETI SEQUENZIALI

5.1 - Modelli di reti sequenziali

Una **rete sequenziale** è la realizzazione fisica di un automa a stati finiti.

All'esterno una rete sequenziale è rappresentata, come qualunque rete logica, con n ingressi ed m uscite, a cui sono associate altrettante variabili logiche.

All'interno, per poter realizzare gli stati dell'automa, la rete deve possedere **elementi di memoria**. Poichè come si è visto qualunque rete si ottiene esclusivamente collegando tra loro blocchi logici, gli elementi di memoria in una rete sequenziale devono poter essere ottenuti mediante opportune connessioni tra blocchi logici. Precisamente le connessioni richieste sono realizzate in modo che si formino uno o più **cicli** diretti o indiretti tra l'uscita di uno (o più) blocchi logici e l'ingresso dello stesso (o degli stessi).

Consideriamo una rete R contenente uno o più cicli. In R è sempre possibile trovare dei blocchi tali che, sconnettendone le uscite dal resto delle rete, si interrompono tutti i cicli. Siano Y_0, Y_1, \dots, Y_{k-1} le variabili associate alle uscite di tali blocchi e y_0, y_1, \dots, y_{k-1} le variabili associate ai loro ingressi. In questo modo si individua nella rete una porzione priva di cicli, quindi combinatoria, ed una, esterna alla precedente, costituita dai percorsi che portano le variabili Y_i ai punti di entrata delle variabili y_i , $0 \leq i \leq k-1$. La Fig. 5.1 esemplifica la scomposizione.

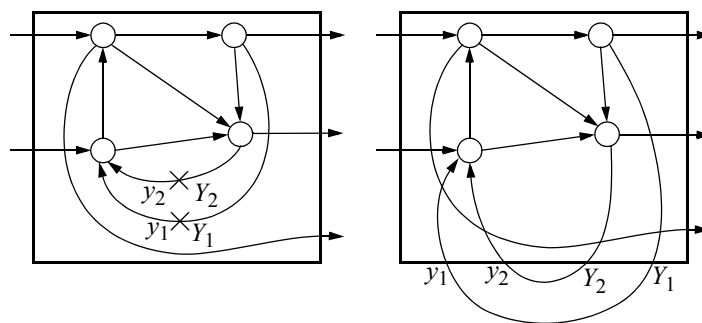


Fig. 5.1

Le connessioni esterne alla parte combinatoria si chiamano **anelli di reazione** e l'insieme $\{Y\}$ o $\{y\}$ costituisce l'insieme delle **variabili di anello**. Esse sono gli elementi logici mediante i quali viene codificata l'informazione relativa agli stati interni dell'automa e vengono quindi dette correntemente **variabili di stato**. Una rete con k anelli possiede k variabili di anello che permettono di codificare 2^k stati interni. Per esempio nell'addizionatore seriale, nel quale esistono due stati, c'è un solo anello (la richiusura del segnale di riporto dall'uscita all'ingresso) ed una sola variabile di anello. In particolare una rete combinatoria ha un solo stato e nessuna variabile di anello, in quanto priva di anelli di reazione. È da notare che il numero e la disposizione dei tagli per eliminare i cicli possono essere diversi in una stessa rete e ciò porta a diverse realizzazioni di uno stesso automa.

Generalizzando queste considerazioni, una rete sequenziale può essere strutturata secondo la Fig. 5.2 (modello di Huffman).

In questo schema la rete combinatoria genera l'insieme $\{Z\}$ delle variabili che codificano lo stato di uscita e l'insieme $\{Y\}$ delle variabili che codificano lo stato successivo e ha come ingressi l'insieme $\{x\}$ delle variabili che codificano lo stato di ingresso e l'insieme $\{y\}$ delle variabili che codificano lo stato attuale. Tra le variabili degli insiemi $\{Y\}$ e $\{y\}$ esiste la correlazione temporale $Y_i \rightarrow y_i$, $0 \leq i \leq k-1$, per cui lo stato successivo diventa stato attuale e questo concorre con gli ingressi a generare il nuovo stato successivo e così via in una sequenza temporale $t_0, \dots, t_{i-1}, t_i, t_{i+1}, \dots$.



Fig. 5.2

Il modello di Huffman, dal momento che assume implicitamente la parte combinatoria priva di ritardi, è in grado di descrivere il funzionamento della rete sequenziale solo considerando la variabile tempo *discretizzata*. In altre parole il tempo viene fatto avanzare a passi unitari e si ammette che le transizioni di stato siano possibili solo tra la fine di un passo e l'inizio del successivo e che la risposta dell'intera rete sia definita in corrispondenza di essi: si tratta di un modello puramente ideale che opera in modo **sincrono**.

In realtà, come sappiamo, internamente alla parte combinatoria esistono ritardi, diversi sui vari percorsi, dei quali occorre tenere conto nel modo di specificare le variabili di uscita e le variabili dello stato successivo in funzione delle variabili di ingresso e delle variabili dello stato attuale. Ai fini del modello possiamo pensare che questi ritardi siano concentrati sugli anelli di reazione e sulle linee di uscita e che la parte combinatoria ne sia priva; in tal modo i valori delle variabili $\{Y\}$ sono generati istantaneamente, mentre l'aggiornamento delle variabili $\{y\}$ e delle

variabili di uscita avviene con i ritardi propri delle rispettive linee. Con questa ipotesi lo schema della rete sequenziale si modifica in quello della Fig. 5.3, nella quale è immediato fare rientrare i modelli strutturali secondo Moore e secondo Mealy, definendo in modo opportuno il comportamento della rete combinatoria per quello che riguarda gli insiemi $\{Y\}$ e $\{Z\}$ in funzione degli insiemi $\{x\}$ ed $\{y\}$:

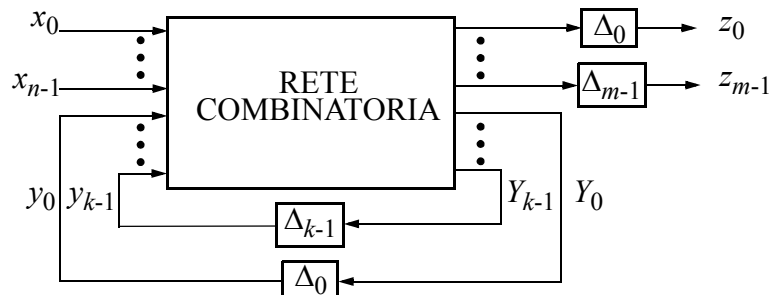


Fig. 5.3

corrispondentemente le equazioni caratteristiche assumono la forma:

$$Y_i(t+\Delta_i) = f_i(x_1(t), \dots, x_n(t), y_1(t), \dots, y_k(t)), \quad 0 \leq i \leq k-1$$

$$z_j(t+\Delta_j) = g_j(x_1(t), \dots, x_n(t), y_1(t), \dots, y_k(t)), \quad 0 \leq j \leq m-1$$

per il modello di Mealy e la forma

$$Y_i(t+\Delta_i) = f'_i(x_1(t), \dots, x_n(t), y_1(t), \dots, y_k(t)), \quad 0 \leq i \leq k-1$$

$$z_j(t+\Delta_j) = g'_j(y_1(t), \dots, y_k(t)), \quad 0 \leq j \leq m-1$$

per il modello di Moore.

Poichè i ritardi $\{\Delta_i\}$ e $\{\Delta_j\}$ sono differenti e in generale sconosciuti, le variabili di stato si propagano sui rispettivi cammini con rapporti di velocità diversi e sconosciuti. Reti che rispondono a questo modello sono dette **asincrone**. Si tratta di reti soggette ad una serie di vincoli che ne limitano il pratico impiego. Infatti, di norma, a) solo una variabile di ingresso x_i per volta può cambiare valore e b) non è consentito alcun cambiamento in ingresso prima che lo stato conseguente l'ultimo cambiamento si sia stabilizzato (proprietà delle **transizioni di ingresso a stati stabili**).

La ragione di queste limitazioni sta nel fatto che le reti asincrone sono soggette per loro natura a **transizioni multiple**, ovvero sono in grado di tenere traccia di tutti i cambiamenti delle variabili di stato che si verificano in conseguenza di un cambiamento dello stato di ingresso. Infatti per una data combinazione di ingresso $\{x_0, x_1, \dots, x_{n-1}, y_0, y_1, \dots, y_{k-1}\}$, viene generata una combinazione di valori delle variabili dello stato successivo $\{Y'_0, Y'_1, \dots, Y'_{k-1}\}$ la quale è riportata in ingresso con velocità diverse sui vari percorsi di reazione determinandone un nuovo assetto, diverso tuttavia da quello finale; in conseguenza viene generata una nuova combinazione $\{Y''_0, Y''_1, \dots, Y''_{k-1}\}$, pur essendo rimaste le variabili $\{x_i\}$ costanti, e così via fino

al raggiungimento dello stato finale previsto.

La rete passa attraverso una sequenza di stati interni instabili, prima di arrivare a quello finale stabile (il quale peraltro potrebbe anche non essere mai raggiunto, potendosi innescare un comportamento oscillatorio oppure potendo diventare stabile uno stato diverso da quello previsto), in conseguenza di un'unica variazione dell'ingresso. Se dunque qualche variabile di ingresso x_i cambiasse durante la fase di stabilizzazione, il comportamento della rete diventerebbe del tutto imprevedibile, non essendo possibile conoscere lo stato interno in cui si trova la rete al momento di tale variazione.

Una conseguenza di quanto detto è il fatto che le reti asincrone possono funzionare correttamente solo con **segnali a livelli**, ossia con segnali nei quali l'intervallo minimo tra due transizioni di livello consecutive non è inferiore al massimo ritardo di stabilizzazione degli stati interni: si dice anche che le reti asincrone lavorano in **modo fondamentale**. Reti asincrone, reti con segnali a livelli, reti in modo fondamentale sono quindi sinonimi, anche se esistono reti in modo fondamentale che possiedono uno o più ingressi che accettano segnali ad impulsi, come sarà visto nel seguito.

Al fine di eliminare l'incertezza sulla durata del tempo di stabilizzazione di una rete asincrona, che può essere difficile da valutare e che comunque può essere diverso da un esemplare all'altro della stessa rete sequenziale, è necessario modificare il modello, sottraendolo all'influenza dei ritardi interni.

La soluzione consiste nel creare una separazione tra lo stato successivo e lo stato attuale, inserendo sui percorsi di reazione elementi di memoria (**elementi di registro**) non temporanei, come lo sono i ritardi dello schema di Fig. 5.3, bensì in grado di conservare per un tempo arbitrariamente lungo i valori delle variabili $\{Y\}$ e di presentarli agli ingressi $\{y\}$ solo in istanti ben precisi e controllabili, sincronizzati da un segnale ad impulsi C . Il nuovo modello che ne deriva è quello di una **rete sequenziale sincronizzata**, schematizzato nella Fig. 5.4.

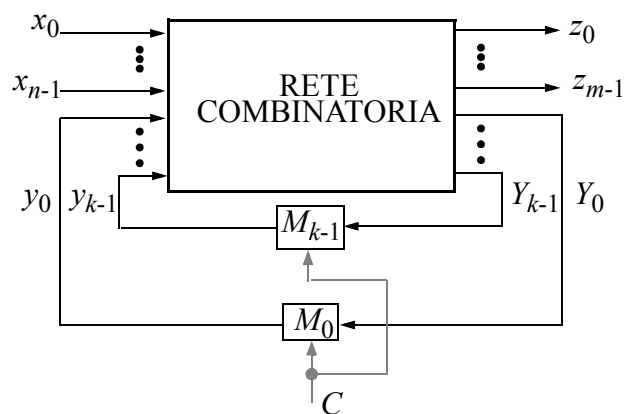


Fig. 5.4

Il modo di funzionamento del modello è il seguente, nell'ipotesi di una rete di Mealy. Supponiamo che gli elementi di registro contengano ad un istante t_0 , assunto come iniziale, lo stato S_h e che in quell'istante sia applicato agli ingressi principali lo stato di ingresso X_i . Sia

Δ_R il massimo ritardo con cui l'informazione si propaga attraverso la parte combinatoria. Dopo tale intervallo lo stato di uscita Z_p e lo stato successivo S_k saranno stabilizzati all'uscita principale della rete e all'ingresso dei registri ed il punto di operazione si troverà nella casella di coordinate (S_h, X_i) della tabella di flusso, contrassegnata con α (Fig. 5.5).

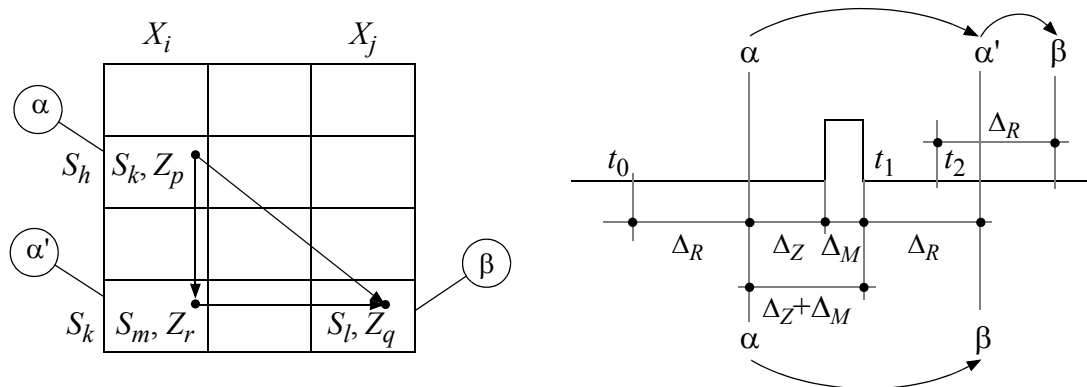


Fig. 5.5

Sia $\Delta_Z \geq 0$ l'intervallo di tempo durante il quale le uscite della rete combinatoria rimangono stabili e supponiamo di applicare un impulso all'ingresso C dei registri dopo questo intervallo. Ciò determina la memorizzazione dello stato successivo S_k presente agli ingressi a livelli dei registri e, dopo il ritardo di propagazione interno Δ_M di questi, S_k diventa stato attuale e si presenta agli ingressi secondari $\{y\}$ della parte combinatoria. Se all'istante t_1 in cui cessa l'impulso si applica un nuovo stato di ingresso X_j , con un ritardo Δ_R il punto di operazione si sposterà sulla tabella di flusso nella posizione β a cui corrisponde il nuovo stato interno successivo S_l ed il nuovo stato di uscita Z_q . A questo punto il ciclo può ricominciare. Se invece il nuovo stato di ingresso X_j viene applicato ad un istante t_2 successivo a t_1 , all'istante $t_1 + \Delta_R$ l'uscita diventa Z_r e tale rimane fino all'istante $t_2 + \Delta_R$ in cui acquista il valore definitivo Z_q , mentre viene previsto lo stato successivo S_m , che a sua volta sarà sostituito dallo stato successivo definitivo S_l all'istante $t_2 + \Delta_R$: sulla tabella di flusso questa situazione si manifesta con lo spostamento del punto di operazione prima in α' e successivamente in β .

Per una rete di Moore, la situazione è rispecchiata nella Fig. 5.6 e mostra che dopo Δ_R unità di tempo a partire dall'istante t_1 , in cui cessa l'impulso di sincronizzazione, compare l'uscita Z_p associata al nuovo stato S_k , mentre gli effetti del nuovo stato di ingresso X_j , indipendentemente dal fatto che sia applicato all'istante t_1 o ad un istante successivo t_2 , si faranno sentire dopo l'ulteriore impulso di sincronizzazione, che al più presto può essere applicato all'istante $t_1 + \Delta_R + \Delta_Z$. A seconda dell'istante in cui lo stato di ingresso X_j viene applicato, il punto di operazione potrà spostarsi direttamente da α a β , oppure transitare prima da α' , ma in ogni caso l'uscita non ne risente.

In ogni caso il comportamento del modello corrisponde esattamente a quello ideale sincrono e quindi può essere sostituito ad esso.

Sul funzionamento di una rete sequenziale che rispecchia questo modello fisico si possono fare alcune osservazioni rispetto alla rete sincrona ideale.

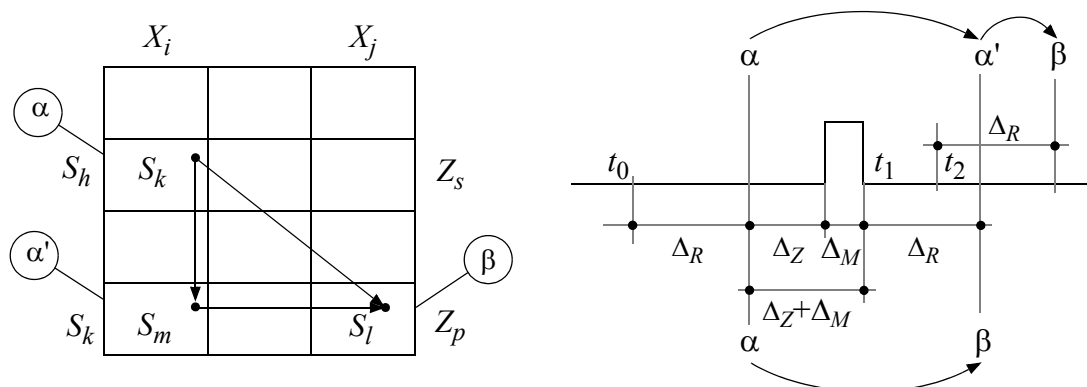


Fig. 5.6

- Nel modello ideale, se lo stato di ingresso e quello interno cambiano insieme, lo spostamento del punto operativo sulla tabella di flusso avviene secondo tale doppia transizione (linea obliqua). Nel modello fisico la presenza del registro consente la sequenzializzazione del cambiamento dello stato interno e di quello di ingresso.
- L'intervallo di operazione nel modello ideale è rigidamente fissato dall'intervallo della sequenza temporale. Invece nel modello reale l'intervallo $\Delta_Z + \Delta_M + \Delta_R$ tra due punti operativi successivi può variare da un minimo pari a $\Delta_M + \Delta_R$, potendo far durare per un tempo nullo la disponibilità dello stato di uscita, ad un massimo qualsiasi.
- Il modo di operare della rete fisica, al contrario di quella asincrona, è di tipo **ad impulsi** per la presenza del segnale C . La durata dell'impulso di questo segnale non deve superare il ritardo interno degli elementi di registro. Infatti se Δ_M fosse arbitrariamente lungo, si verificherebbe una variazione delle variabili $\{y\}$ determinata dalla propagazione delle variabili $\{Y\}$ attraverso gli elementi di memoria, ed una eventuale risposta della parte combinatoria con conseguente variazione indesiderata delle variabili $\{Y\}$ prima del termine dell'impulso. Per garantire il rispetto della stabilità degli ingressi agli elementi di memoria durante l'impulso, occorre modificare il modello inserendo in cascata ad essi opportuni ritardi aggiuntivi; tuttavia vedremo nel seguito che, utilizzando particolari elementi di memoria, questo vincolo può essere rimosso senza danno per il corretto funzionamento della rete.

Le reti reali rispondono oltre che al modello di Fig. 5.4, in cui tutti gli elementi di memoria hanno un unico ingresso di sincronizzazione ad impulsi e costituiscono un unico **registro** di lunghezza pari al numero delle variabili di stato presenti nella rete, anche ad un modello in cui l'impulso di sincronismo viene inviato separatamente a ciascun elemento di memoria.

Esistono inoltre due categorie principali di reti sincronizzate: *a)* quelle nelle quali ai terminali di ingresso e di uscita della parte combinatoria sono applicati solo segnali a livelli e *b)* quelle in cui esiste almeno un ingresso su cui si trova un segnale ad impulsi.

a) Lo schema generale delle reti di questa categoria è mostrato in Fig. 5.7; rispetto allo schema di Fig. 5.4, ogni elemento di memoria ha uno o più ingressi (**variabili di eccitazione**) che sono in relazione ad una variabile di stato successivo, nel senso che sono argomenti delle funzioni di stato realizzate dall'elemento di memoria stesso. Il modo di funzionamento tuttavia non differisce da quello del modello generale e queste reti possono essere realizzate sia secondo il modello di Mealy che secondo quello di Moore.

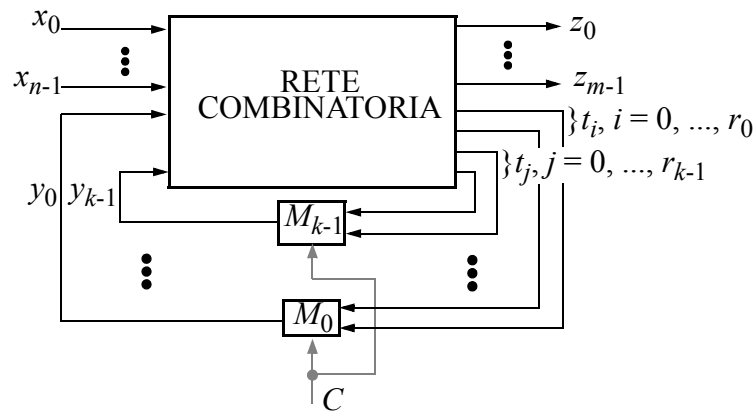


Fig. 5.7

b) Alla seconda categoria appartengono le reti nelle quali il segnale ad impulsi è applicato, oltre che agli elementi di memoria, anche ad almeno un ingresso primario della rete combinatoria, o meglio della parte di essa che genera i segnali di uscita; la temporizzazione delle uscite principali è quindi modificata rispetto al modello generale, in quanto la loro disponibilità è condizionata alla presenza del segnale di sincronismo; si tratta di reti modellate secondo Mealy ritardato. Anche in questo schema gli elementi di memoria possiedono uno o più ingressi a cui sono applicate le variabili di eccitazione generate dalla parte combinatoria (Fig. 5.8).

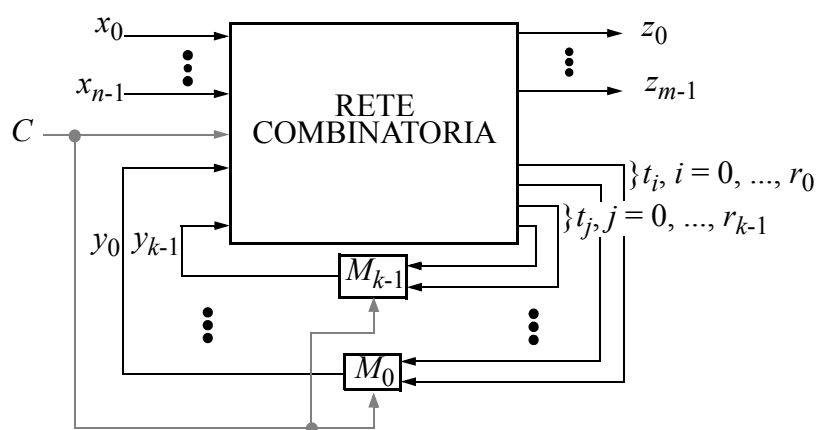


Fig. 5.8

5.2 - Elementi di memoria

I più semplici elementi di memoria sono reti sequenziali asincrone o sincronizzate, dette **latch** e **flip-flop**, le quali possiedono due stati interni, codificati come stato 0 e stato 1.

5.2.1 - Latch

5.2.1.1 - Latch SR

Il più noto componente della famiglia, anche per ragioni storiche, è il **latch SR**. Si tratta di una rete, facilmente modellabile come macchina di Moore, con due ingressi denominati S (**set**) ed R (**reset**) ed una uscita Q la quale coincide con l'unica variabile di stato y , come è indicato nel seguente schema a blocchi (Fig. 5.9).

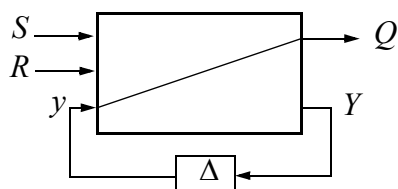


Fig. 5.9

Per questa rete le equazioni caratteristiche hanno la seguente forma:

$$Y(t+\Delta) = f(S(t), R(t), y(t))$$
$$Q(t) = g(y(t)) = y(t)$$

L'espressione esplicita della funzione f può essere determinata a partire dalla descrizione a parole del funzionamento del latch attraverso le seguenti frasi, dove \times sta per "non specificato":

$S = 1$ e $R = 0 \Rightarrow$	$Y = 1$	(operazione di set)
$S = 0$ e $R = 1 \Rightarrow$	$Y = 0$	(operazione di reset)
$S = 0$ e $R = 0 \Rightarrow$	$Y = y$	(operazione di latch)
$S = 1$ e $R = 1 \Rightarrow$	$Y = \times$	(configurazione proibita)

Nell'operazione di latch lo stato successivo è uguale allo stato attuale, sia esso lo stato stabile 1 oppure lo stato stabile 0; lo stato 1 è raggiunto con una operazione di set, lo stato 0 con una operazione di reset; infine la combinazione di ingresso $S = R = 1$ non è consentita poichè, come vedremo, può non essere determinato lo stato assunto dalla rete quando tale configurazione cambia.

Dalla descrizione a parole si ricava la seguente mappa di Karnaugh e quindi l'espressione di f , ad esempio in forma PS (Fig. 5.10). La rete corrispondente è illustrata nella figura in due forme alternative, di cui la seconda evidenzia anche il segnale di uscita della prima porta NOR; questo segnale Q' in genere, ma non sempre, è il complemento dell'uscita Q .

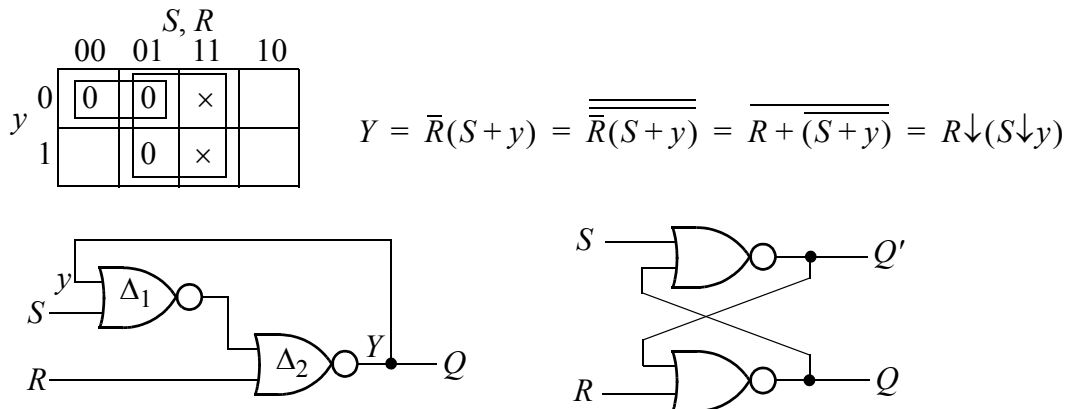


Fig. 5.10

La mappa di Karnaugh sulla quale viene effettuata la sintesi altro non è che la tabella di flusso del latch e da essa deriva l'equazione caratteristica, imponendo per Y la condizione sotto la quale il latch assume o rimane nello stato stabile 0 (o stato di reset). Vale la pena notare che, rispetto alla forma della tabella di flusso degli automi a stati finiti esaminati nel capitolo 4, nella quale gli stati venivano espressi con un nome simbolico, in una rete sequenziale gli stati sono codificati dalle variabili di stato e quindi sono indicati con le combinazioni di valori binari che esse possono assumere.

Imponendo la condizione sotto la quale il latch assume o rimane nello stato stabile 1 (o stato di set), si ottiene la seguente espressione NAND:

$$Y = S + \bar{R}y = \bar{S} | (\bar{R} | y)$$

nella quale i segni di complementazione attribuiti alle variabili S ed R indicano che il funzionamento risponde ancora alla definizione a parole, purchè si intenda invertito il valore da assegnare alle variabili di ingresso per ogni operazione del latch. In altri termini l'operazione di set, ad esempio, avrà luogo quando $S = 0$ ed $R = 1$. La figura 8.10 mostra il circuito corrispondente:

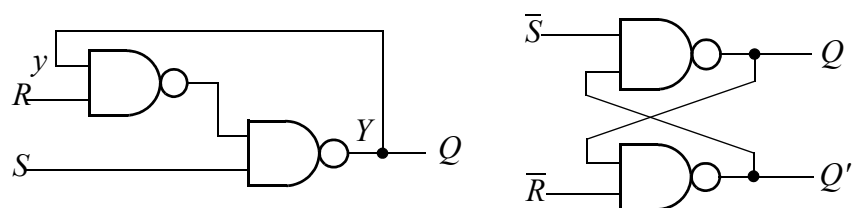


Fig. 5.11

L'informazione contenuta nella tabella di flusso può essere rappresentata anche in un'altra forma detta **tabella di eccitazione** (trigger table), la quale specifica quali devono essere i valori di ingresso per ogni possibile transizione dallo stato presente allo stato successivo:

y	Y	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

L'interpretazione della tabella è la seguente. Se il latch deve rimanere nello stato 0 è necessario che l'ingresso di set S sia 0, indipendentemente dal valore dell'ingresso di reset R . Infatti se $R = 0$, viene effettuata un'operazione di latch e lo stato rimane inalterato, nella fattispecie 0; se $R = 1$, il latch compie una operazione di reset e lo stato successivo sarà certamente 0. Analogamente si interpretano le altre righe della tabella.

Il latch SR, come ogni rete asincrona, funziona a **livelli**, ovvero i segnali di ingresso devono mantenere il proprio valore costante per un periodo di tempo pari almeno al ritardo dell'intera rete; tale intervallo è chiamato **tempo di eccitazione** o **trigger time**. Più precisamente, se indichiamo con T_S e T_R i tempi di eccitazione dell'ingresso S e dell'ingresso R e con $\Delta = \Delta_1 + \Delta_2$ il ritardo del latch, espresso dalla somma dei ritardi delle due porte NOR, deve essere:

$$T_S, T_R \geq \Delta = \Delta_1 + \Delta_2$$

come è visualizzato nel seguente diagramma temporale:

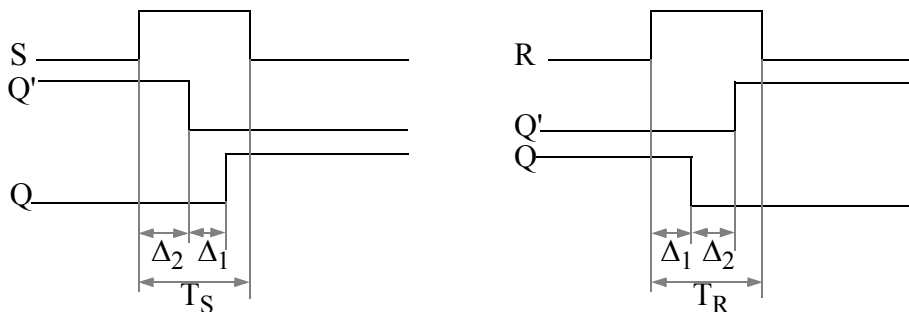


Fig. 5.12

Se viceversa i segnali di ingresso S od R rimangono al valore 1 per un intervallo di tempo inferiore al periodo di eccitazione, il latch non riesce a seguire la variazione dello stato di ingresso e può effettuare un'operazione scorretta i cui effetti si manifestano con un impulso spurio (spike) in uscita (Fig. 5.13).

Vediamo ora il motivo per il quale non è consentita la configurazione di ingresso $S = R = 1$, riferendoci allo schema di Fig. 5.11, e tenendo presente che ragionamenti duali valgono anche per lo schema di Fig. 5.12. Se entrambi gli ingressi S ed R hanno valore 1, sia Q che Q' diventano 0, poichè entrambe le porte NOR hanno un ingresso che vale 1. Se a questo punto simultaneamente S ed R assumono valore 0, non è possibile sapere con certezza quale stato assume il latch, in quanto ciò dipende dalla velocità relativa con cui rispondono le porte NOR

e dal ritardo con cui i nuovi valori di Q e di Q' si propagano ciascuno all'ingresso dell'altra porta NOR.

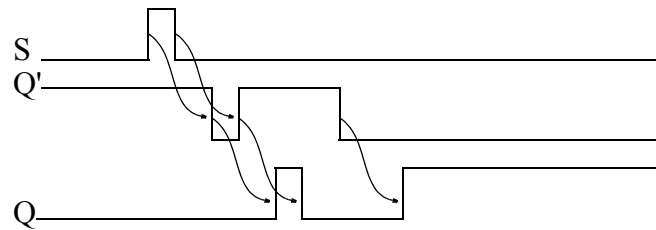


Fig. 5.13

Infatti dall'esame del circuito è possibile vedere che se il ritardo Δ_1 è minore del ritardo Δ_2 , l'uscita Q conserva valore 0 e Q' diventa 1, ossia il latch assume lo stato 0. Viceversa se Δ_1 è maggiore di Δ_2 , l'uscita Q assume valore 1 e Q' il valore 0, corrispondenti allo stato 1. Questa incertezza viene risolta quando uno dei due ingressi assume di nuovo valore 1 (Fig. 5.14).

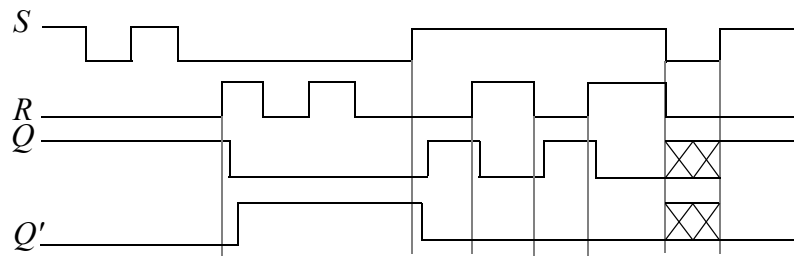


Fig. 5.14

È da osservare che, come è stato accennato in precedenza e come indica la Fig. 5.14, non sempre i segnali di uscita delle porte NOR hanno valori opposti e ciò accade quando S ed R hanno lo stesso valore 1.

Il latch SR può essere dotato di un ingresso ad impulsi C con funzione di ingresso di controllo e prende allora il nome di **gated latch** o anche, seppure impropriamente, di latch sincronizzato (Fig. 5.15). In questa versione le operazioni del latch sono possibili solo quando C è attivo, ossia i segnali S ed R sono di fatto sostituiti dai segnali CS e CR rispettivamente, per cui gli ingressi S ed R non hanno effetto finché C non diventa 1.

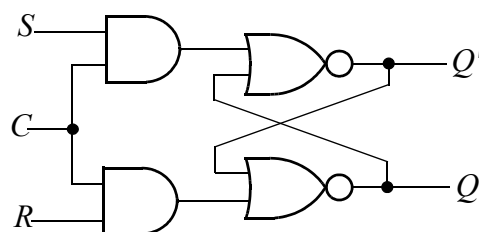


Fig. 5.15

Gli ingressi condizionati dal segnale di controllo si dicono **ingressi sincroni**, a significare che essi hanno effetto quando il latch è controllato dall'attività dell'ingresso C , mentre quelli

dei latch delle figure 5.10 e 5.11 sono detti **asincroni**; esistono latch con ingressi sia sincroni che asincroni, questi ultimi indicati con termini quali **preset** e **clear** e utilizzati per effettuare operazioni di set e di reset in modo asincrono rispetto al segnale C e quindi rispetto all'operazione corrente.

Le forme d'onda dei segnali sono mostrate nella Fig. 5.16, la quale indica in corrispondenza degli istanti t_a e t_b che il dispositivo trasmette direttamente in uscita, a meno del ritardo di propagazione proprio, gli effetti di una variazione sugli ingressi S o R , in presenza del segnale C attivo. Questo comportamento è una caratteristica dei latch che viene indicata con il termine di **trasparenza** e i circuiti sequenziali con due stati vengono distinti in **latch** se hanno la caratteristica della trasparenza e in **flip-flop** altrimenti.

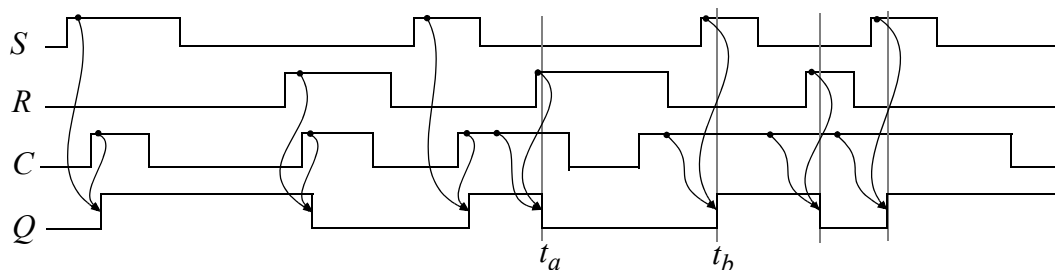


Fig. 5.16

Il fatto che il segnale C debba essere attivo affinché le variazioni di ingresso abbiano effetto sull'uscita indica che il latch SR è sensibile al livello di C . Questo può determinare il fenomeno, indesiderato in una rete sincronizzata dalla quale ci si aspetta che le variazioni di stato e di uscita si verifichino a istanti ben definiti, di avere transizioni dell'uscita anche dopo che l'ingresso C è passato da 0 ad 1 (istante t_a), o che addirittura l'uscita possa cambiare più volte durante uno stesso impulso di sincronizzazione (intervallo a partire da t_b). La soluzione del problema richiede che la durata dell'impulso di sincronizzazione (**control pulse width**) sia ridotta al minimo, compatibilmente con la necessità di assicurare che i segnali di ingresso rimangano invariati per un tempo sufficiente per il raggiungimento del nuovo stato; ma questo rende comunque impossibile definire con certezza l'istante della sincronizzazione e per questo motivo i latch sincronizzati tendono ad essere sostituiti nelle applicazioni da altri elementi di memoria sincronizzabili effettivamente sul fronte di un segnale di controllo.

Un altro punto da sottolineare è che l'indeterminatezza di funzionamento del latch SR quando gli ingressi passano dal valore 11 al valore 00 contemporaneamente, più teorica che pratica in un latch asincrono a causa della scarsa probabilità che due ingressi possano realmente cambiare nello stesso istante, diventa invece reale in un latch sincronizzato, se S ed R sono 1 quando C passa da 1 a 0.

5.2.1.2 - Latch D

Se in un latch SR, sia del tipo asincrono, sia del tipo sincronizzato, si fa in modo che agli ingressi a livelli S ed R siano applicati segnali di valore opposto l'uno rispetto all'altro, si

ottiene un nuovo tipo di latch denominato **latch D**, il cui schema è illustrato nella Fig. 5.17. Questo latch possiede solo un ingresso a livelli D e internamente si differenzia da un SR unicamente per la presenza di un invertitore che complementa tale segnale all'ingresso di reset R . Il vantaggio di questa struttura è duplice: 1) il segnale Q' è sempre il complemento di Q e 2) non esiste il rischio di una combinazione di ingresso proibita. L'equazione caratteristica si ottiene semplicemente considerando che è $R = \bar{S} = \bar{D}$, ossia:

$$Y = D \cdot (D + y) = D$$

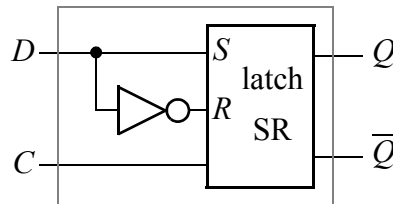


Fig. 5.17

valida per la versione asincrona; essa indica che l'uscita è uguale all'ingresso, ovviamente ritardata rispetto ad esso del tempo di propagazione. La tabella di eccitazione specifica tale fatto nella seguente forma:

y	Y	D
0	0	0
0	1	1
1	0	0
1	1	1

ovvero affinché lo stato successivo sia nullo, l'ingresso deve essere 0, mentre un valore dell'ingresso uguale ad 1 provoca l'aggiornamento dello stato pure al valore 1. Per la versione sincronizzata, l'equazione caratteristica assume la seguente espressione, ottenuta applicando i teoremi di De Morgan ed il teorema del consenso e ricordando ancora che $R = \bar{D}$:

$$Y = \overline{\bar{C}\bar{D}} \cdot (CD + y) = (\bar{C} + D) \cdot (CD + y) = CD + y\bar{C} + yD = CD + y\bar{C}$$

Si noti che se $C = 0$, risulta $Y = y$, ossia il latch è in grado di conservare stabilmente lo stato, cosa non possibile nella versione asincrona.

La Fig. 5.18 illustra le forme d'onda tipiche del latch D, evidenziando in t_a , t_b , t_c la trasparenza del dispositivo che è ancora del tipo sensibile al livello di C .

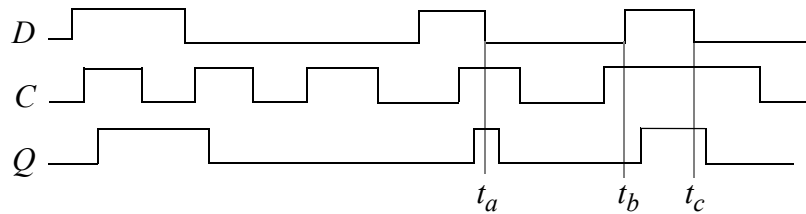


Fig. 5.18

5.2.2 - Flip-flop

Un flip-flop è un elemento di memoria a due stati con un ingresso di controllo, il quale non ha la proprietà della trasparenza, ossia nel quale i cambiamenti dell'uscita non sono mai una risposta diretta di cambiamenti degli ingressi sincroni. Viceversa un flip-flop risponde solo a transizioni dell'ingresso di controllo o degli ingressi asincroni, se presenti. La differenza di comportamento di un flip-flop rispetto ad un latch è messa in evidenza dal seguente grafico che mostra le forme d'onda del segnale di uscita Q_{ff} di un flip-flop D, che esamineremo tra breve, e del segnale di uscita Q_l del latch omonimo:

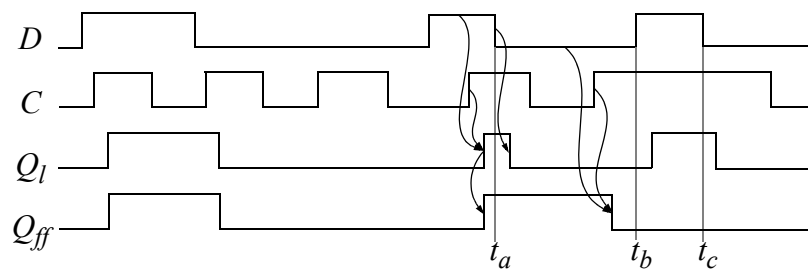


Fig. 5.19

Come si vede il comportamento dei due dispositivi è lo stesso finché il segnale D cambia mentre C è 0; invece all'istante t_a , in cui si verifica la transizione $1 \rightarrow 0$ di D essendo $C = 1$, l'uscita Q_l del latch cambia con il proprio ritardo interno dal valore 1, che aveva assunto in conseguenza della transizione $0 \rightarrow 1$ di C , al valore 0, mentre quella del flip-flop Q_{ff} rimane insensibile a tale transizione di ingresso, conservando il valore 1 assunto allo stesso modo di Q_l ; l'uscita Q_{ff} subisce la transizione $1 \rightarrow 0$ in conseguenza della successiva transizione $0 \rightarrow 1$ di C con $D = 0$. In maniera analoga, all'istante t_b si verifica la transizione $0 \rightarrow 1$ dell'ingresso sincrono D , con $C = 1$; ad essa il latch risponde con la transizione $0 \rightarrow 1$ dell'uscita, mentre l'uscita del flip-flop rimane a 0. Infine all'istante t_c la caduta di D con $C = 1$ provoca la caduta di Q_l , ma non ha effetto su Q_{ff} .

La non trasparenza dei flip-flop viene ottenuta attraverso una configurazione di tipo master-slave: due dispositivi trasparenti sono collegati in cascata, in modo che l'uscita del primo (**master**) sia ingresso del secondo (**slave**), e sono controllati con i segnali \overline{C} e C

rispettivamente (Fig. 5.20), cosicchè la transizione $1 \rightarrow 0$ di C determina l'aggiornamento dello stato del master, lasciando immutato lo stato dello slave e la transizione $0 \rightarrow 1$ provoca l'aggiornamento dello stato dello slave, concordemente all'uscita del master, mentre lo stato di questo non è influenzato dal valore degli ingressi sincroni. In sostanza il latch master ha il compito di elaborare l'informazione modificando il proprio stato in risposta a variazioni di ingresso, mentre il latch slave serve per ritardare la propagazione del risultato in uscita.

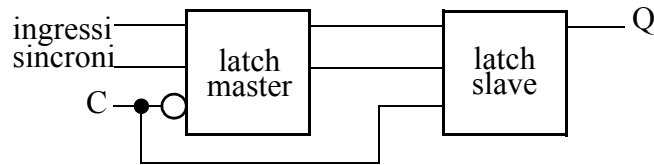


Fig. 5.20

5.2.2.1 - Flip-flop D

Utilizzando nella Fig. 5.20 latch D come master e slave, si ottiene un flip-flop D master-slave, il cui schema è mostrato in Fig. 5.21; con la notazione della figura è immediato anche ricavarne le equazioni caratteristiche:

$$Y_1 = \bar{C}D + Cy_1 \quad Y_2 = Cy_1 + \bar{C}y_2 \quad Q = y_2$$

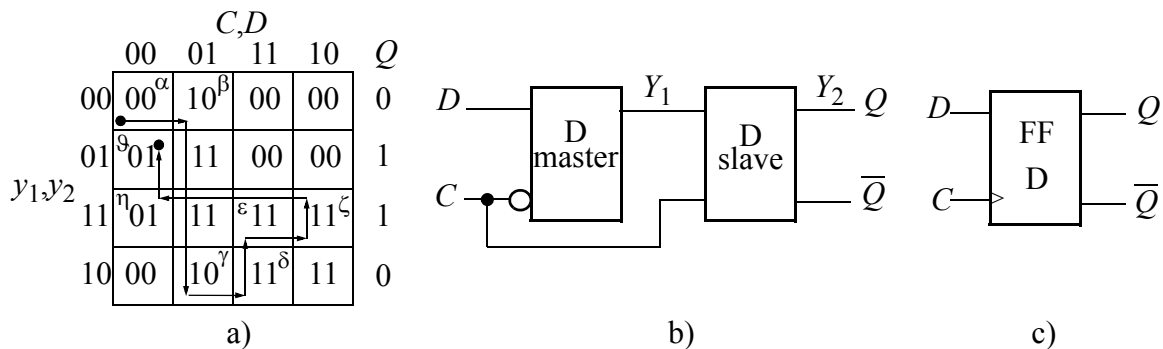


Fig. 5.21

Da esse, come pure dalla tabella di flusso mostrata in Fig. 5.21a, si vede che quando il segnale C è 0, lo stato Y_1 del master segue le variazioni dell'ingresso D , ma lo stato dello slave e di conseguenza l'uscita Q si mantengono stabili; viceversa quando C è 1, il master non può modificare il proprio stato comunque vari l'ingresso D , mentre lo stato Y_2 e l'uscita Q diventano uguali all'ingresso Y_1 .

Pertanto l'uscita Q cambia diventando uguale a D a causa della transizione $0 \rightarrow 1$ di C ; per renderci meglio conto di questo fatto, supponiamo che il flip-flop si trovi nello stato stabile 00 e che C e D siano 0 (casella α della tabella di flusso); se a questo punto cambia l'ingresso D da 0 a 1, lo stato diventa 10, pure stabile: il punto di operazione si sposta sulla tabella da α in β e

quindi in γ : come si vede la variazione di D non si risente in uscita, che continua ad essere 0. Se ora si verifica la transizione $0 \rightarrow 1$ di C , lo stato diventa 11 ed il punto di operazione si sposta in δ e quindi in ϵ , che rappresenta una condizione di stato stabile, caratterizzata dal fatto che l'uscita Q vale 1, ossia è diventata uguale all'ingresso; se D cambia di nuovo da 1 a 0 con C ancora ad 1, lo stato rimane immutato (casella ζ) e l'uscita pure. Una transizione $1 \rightarrow 0$ di C con D stabile a 0 porta il flip-flop nello stato 01 (caselle η e θ) e l'uscita rimane stabile ad 1. Dunque qualsiasi variazione di D con C uguale a 0 viene memorizzata nel master, mentre qualunque sua variazione con $C = 1$ non ha alcun effetto sul circuito. Viceversa una transizione $0 \rightarrow 1$ di C provoca l'aggiornamento dello stato di uscita e una transizione opposta è senza influenza. La tabella di eccitazione del flip-flop D è la seguente:

Il flip-flop descritto, per il fatto di possedere un ingresso C di controllo dinamico, nel senso che l'ingresso D viene acquisito sul fronte in salita di tale controllo e l'uscita varia seguendo immediatamente tale fronte, viene qualificato come **flip-flop D positive edge triggered** (eccitato sul fronte in salita). Le sue forme d'onda tipiche sono quelle già mostrate nella Fig. 5.19. L'ingresso dinamico viene riconosciuto sullo schema a blocchi della Fig. 5.21c con il simbolo ">". Cambiando l'ordine con cui il segnale C ed il suo complemento sono applicati al latch master e allo slave, si ottiene un flip-flop D eccitato sul fronte in discesa (negative edge triggered).

5.2.2.2 - Flip-flop JK

Anche il latch SR può essere trasformato in un flip-flop non trasparente di tipo master slave, allo stesso modo con cui è stato realizzato il flip-flop D; tuttavia, a causa del non determinismo del funzionamento quando S ed R sono 1, si preferisce al suo posto un flip-flop, detto **flip-flop JK**, che si ottiene riportando le uscite Q e Q' rispettivamente sull'ingresso R e sull'ingresso S del latch master attraverso due porte AND insieme a due ingressi esterni denominati J e K rispettivamente (Fig. 5.22).

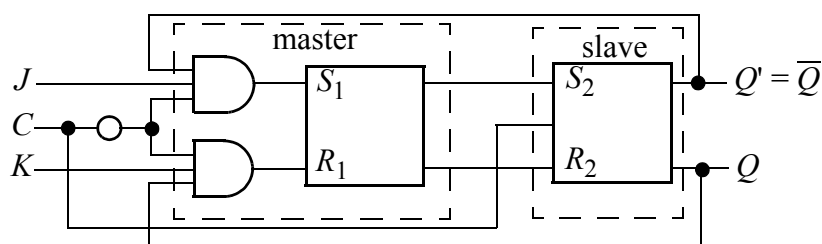


Fig. 5.22

Il flip-flop JK ha il vantaggio di consentire tutte e quattro le combinazioni di J e di K ed il suo funzionamento rispecchia quello del flip-flop SR (o del latch SR), con gli ingressi J e K che corrispondono agli ingressi S ed R rispettivamente: quando $C = 0$, qualsiasi variazione degli ingressi J e K non viene avvertita all'uscita, che rimane stabile. Se $J = 0$ e $K = 0$ quando C cambia da 0 ad 1, l'uscita rimane inalterata; se $J = 0$ e $K = 1$ quando avviene la transizione di C , viene effettuata un'operazione di reset e l'uscita assume valore 0; se $J = 1$ e $K = 1$ al

momento della transizione di C , l'uscita assume il valore complementare di quello originario. Infine se $J = 1$ e $K = 0$, avviene un'operazione di set e l'uscita assume valore 1. La transizione $1 \rightarrow 0$ di C non ha alcun effetto sull'uscita.

Nella Fig. 5.23a il funzionamento del flip-flop JK è riassunto attraverso la sua tabella di flusso mentre la Fig. 5.23b riporta la tabella di eccitazione, la cui interpretazione è la seguente. Se lo stato attuale (y) e quello successivo (Y) sono uguali a 0, ciò può essere dovuto a due motivi: 1) il flip flop sta effettuando un'operazione di latch ($J=0$ e $K=0$), oppure un'operazione di reset di uno stato che è già 0 ($J=0$ e $K=1$). Se lo stato attuale è 0 ed il successivo 1, si può essere in presenza di un'operazione di set ($J=1$, $K=0$), oppure di un'operazione di complementazione dello stato ($J=1$, $K=1$). Nel caso opposto si può avere un'operazione di reset ($J=0$ e $K=1$) oppure di complementazione dello stato ($J=1$ e $K=1$). Infine se lo stato attuale e quello successivo sono entrambi 1, si può avere un'operazione di latch ($J=0$ e $K=0$) oppure di set ($J=1$ e $K=0$).

		$C=0$				$C=1$	Q
		J, K					
		00	01	11	10		
y_1, y_2	00	00	00	10	10	00	0
	01	11	01	01	11	00	1
	11	11	01	01	11	11	1
	10	00	00	10	10	11	0

a)

y	Y	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

b)

C	J	K	Q
0	X	X	Q
\uparrow	0	0	Q
\uparrow	0	1	0
\uparrow	1	1	\overline{Q}
\uparrow	1	0	1

c)

Fig. 5.23

La Fig. 5.23c infine riporta una tabella correntemente fornita dai costruttori di componenti digitali al posto della tabella di flusso per riassumere il funzionamento del flip-flop, in quanto esprime l'uscita in funzione dell'ingresso di sincronizzazione C e degli ingressi sincroni. il simbolo \uparrow indica la transizione positiva del segnale C .

Oltre alla versione di Fig. 5.22, per il flip-flop JK si usa anche lo schema di Fig.5.24 che fa uso di un flip-flop D master-slave e realizza un dispositivo del tipo **positive edge-triggered**.

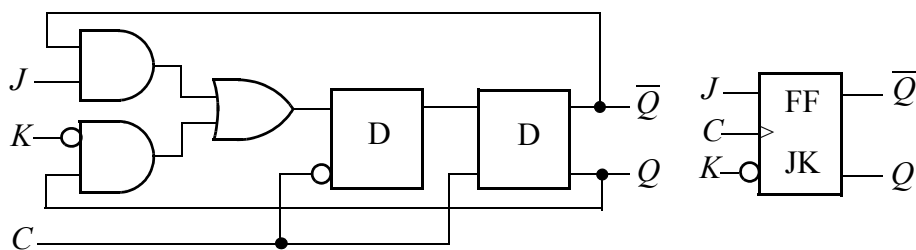


Fig. 5.24

Le forme d'onda tipiche di un flip-flop JK sono mostrate in Fig. 5.25:

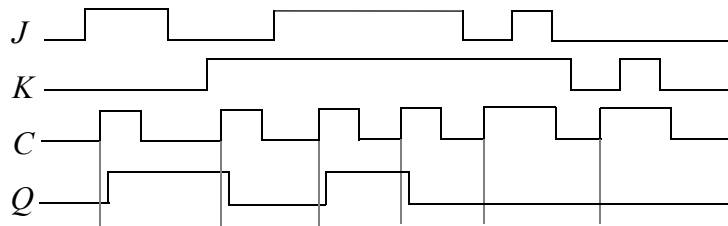


Fig. 5.25

5.2.2.3 - Flip-flop T

Un terzo tipo di flip-flop edge-triggered è il flip-flop T, detto talvolta **toggle**, schematizzato nella Fig. 5.25a. Il modo di operazione è il seguente; se T è 0, lo stato rimane stabile; se T è 1, lo stato cambia quando si verifica la transizione $0 \rightarrow 1$ di C . Questo flip flop si può derivare dal flip-flop JK edge triggered collegando tra di loro gli ingressi J e K (Fig. 5.25b), oppure si può realizzare con un flip-flop D ed una porta XOR, secondo lo schema di Fig. 5.26c.

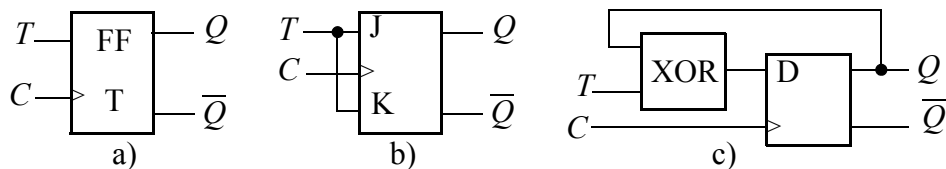


Fig. 5.26

Nella Fig. 5.27a è riportata la tabella analoga a quella di Fig. 5.23c per il flip-flop JK e la tabella di eccitazione; inoltre è mostrato anche il diagramma temporale.

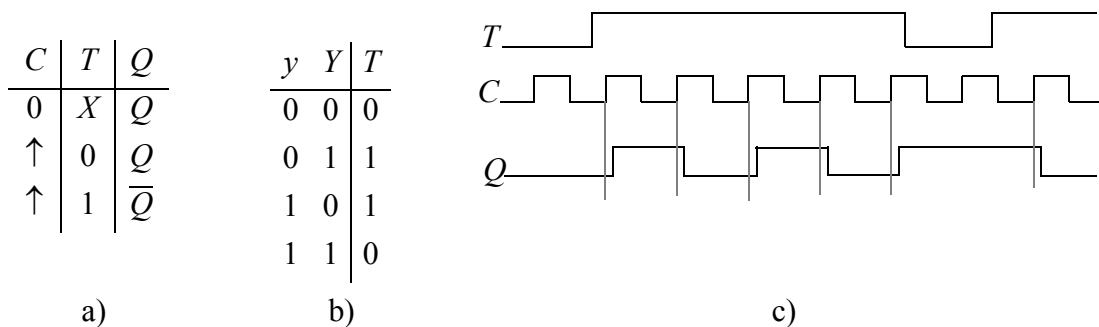


Fig. 5.27

5.2.2.4 - Flip-flop asincroni

Esistono infine due tipi di flip-flop master-slave asincroni: essi sono un **flip-flop SR** ed un **flip-flop T**, entrambi caratterizzati dal fatto che rispondono ad impulsi sui loro ingressi e sono in genere utilizzati non come singoli componenti, ma nel progetto di strutture più complesse.

In particolare il flip-flop SR master-slave asincrono trova applicazione quando in ingresso vi possono essere più impulsi dei quali solo il primo deve essere avvertito. Infatti questo

flip-flop effettua un'operazione di set sul fronte in discesa di un impulso sull'ingresso S , mentre impulsi successivi su S sono ignorati; analogamente viene effettuata un'operazione di reset sul fronte in discesa del primo impulso sull'ingresso R . Lo schema logico è mostrato in Fig. 5.28 insieme alle forme d'onda:

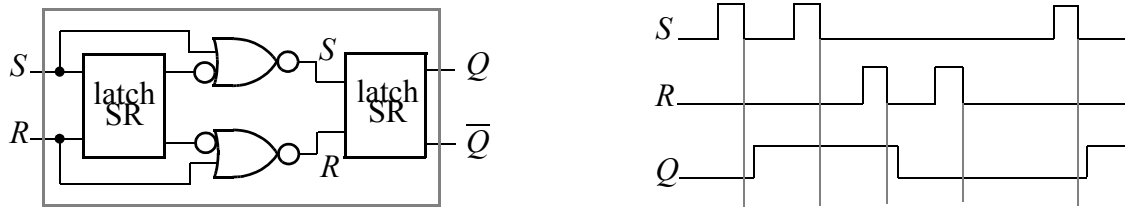


Fig. 5.28

Il flip-flop T master-slave asincrono ha un solo ingresso T e cambia stato ogni volta che un impulso si presenta sull'ingresso, per cui è particolarmente adatto a realizzare una funzione di conteggio binario. Si può derivare da un flip-flop JK master-slave mantenendone gli ingressi J e K ad 1 e applicando il segnale T all'ingresso di sincronizzazione (Fig. 5.29a) oppure da un flip-flop D richiudendone l'uscita \bar{Q} all'ingresso D e applicando il segnale T come nel caso precedente (Fig. 5.29b). Le forme d'onda sono mostrate in Fig. 5.29c.

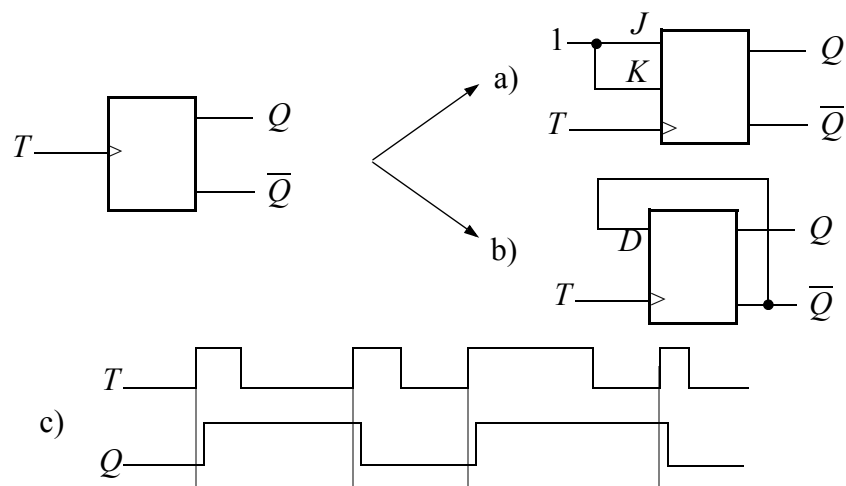


Fig. 5.29

5.3 - Sintesi delle reti sequenziali sincronizzate

Il procedimento per la sintesi delle reti sequenziali sincronizzate può essere suddiviso in due passi.

- a) Costruzione della tabella di flusso dell'automa che descrive il comportamento, definito a parole, della rete che soddisfa certe relazioni ingresso-uscita prefissate, che realizza determinate operazioni o che risolve un dato problema.

b) Costruzione della rete a partire dalla tabella di flusso secondo un prefissato modello strutturale.

a) La costruzione della tabella di flusso rappresenta l'operazione più laboriosa e difficile, specialmente quando il numero degli stati interni, di ingresso e di uscita è elevato. Per questo è utile avvalersi, ad uno stadio intermedio tra la descrizione a parole originaria e la tabella di flusso, di un grafo di stato che tenga conto del modello di Mealy o di Moore che si intende assumere.

Quindi la tabella viene costruita inserendo una riga per ogni nodo del grafo ed una colonna per ogni arco uscente da un nodo, ossia per ogni stato di ingresso. Ciascuna casella viene contrassegnata con lo stato successivo, ricavato dal nodo a cui è diretto un arco uscente e con lo stato di uscita corrispondente per la macchina di Mealy, oppure aggiungendo a margine una colonna con lo stato di uscita per la macchina di Moore.

Per esempio si voglia costruire una rete sequenziale la quale è in grado di riconoscere, in una sequenza arbitraria di caratteri alfanumerici in ingresso, una sottosequenza particolare, quale *aba*, segnalando il riconoscimento con uno stato di uscita z_2 ; lo stato z_1 segnala invece ogni altra condizione.

Indicando con x qualunque carattere diverso da a e da b , i grafi di stato secondo Mealy e secondo Moore sono i seguenti:

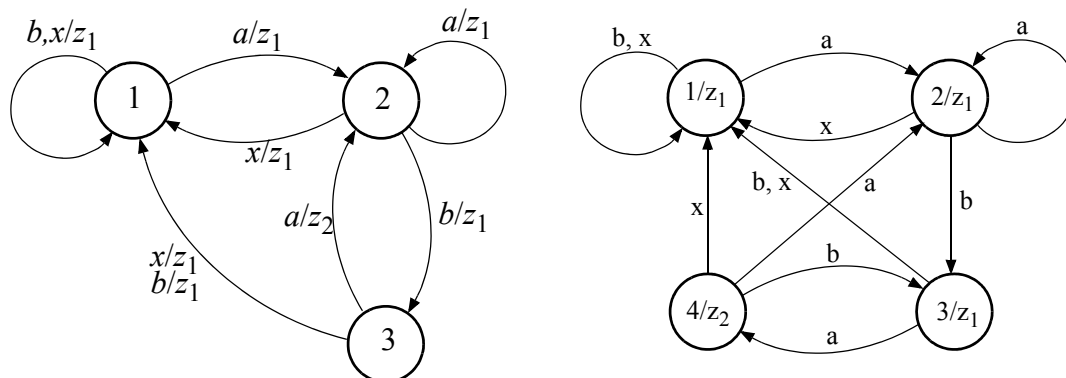


Fig. 5.30

Per il grafo secondo Mealy, supponendo di essere nello stato 1, finchè nella sequenza di ingresso non si incontra il carattere a , lo stato non cambia e l'uscita rimane z_1 ; quando si presenta a , poichè questo può essere l'inizio di una sottosequenza da riconoscere, lo stato cambia in 2 e l'uscita rimane z_1 . Nello stato 2, finchè vengono riconosciuti caratteri a , si rimane nello stesso stato, mentre qualunque carattere x fa ritornare l'automa nello stato 1; invece il riconoscimento di b fa passare nello stato 3 ancora con uscita z_1 . Infine dallo stato 3 il presentarsi in ingresso di un secondo carattere a provoca il cambiamento dell'uscita a z_2 , perchè la sottosequenza è stata riconosciuta, mentre lo stato successivo è nuovamente 2 e non 1, come ci si potrebbe aspettare, dal momento che il carattere a che termina la sottosequenza riconosciuta può essere l'inizio di una nuova sottosequenza *aba*; al contrario qualunque altro

carattere riporta l'automa nello stato 1 e lo stato di uscita torna ad essere z_1 .

Per l'automa di Moore con stato iniziale 1 e stato di uscita corrispondente z_1 , il presentarsi dello stato di ingresso a provoca ancora il passaggio in 2 con uscita z_1 , mentre qualunque altro stato di ingresso lascia immutato lo stato. L'automa rimane nello stato 2 finchè in ingresso è presente lo stato a , raggiungendo lo stato 3 se in ingresso si presenta b ; invece qualunque altro stato di ingresso x riporta in 1. L'uscita associata allo stato 3 è ancora z_1 e il riconoscimento di un ulteriore carattere a fa passare nello stato 4, dove l'uscita diventa z_2 , per l'avvenuto riconoscimento di aba ; si noti che in questo caso il presentarsi di una seconda a dopo b non riporta lo stato in 2, perchè l'uscita deve assumere il valore che indica l'esito positivo della ricerca, mentre allo stato 2 è associato il valore di uscita che indica esito negativo. Dallo stato 4 l'arrivo di b riporta l'automa in 3 con uscita z_1 , ed in questo modo viene accettata la seconda a della sottosequenza e riconosciuta come inizio di una nuova; invece un'ulteriore a riporta in 2 con uscita z_1 e infine qualunque altro carattere x riporta nello stato 1.

Una volta costruiti i grafi di stato si passa a costruire le tabelle di flusso per le due macchine nel seguente modo:

	a	b	x	
1	2, z_1	1, z_1	1, z_1	
2	2, z_1	3, z_1	1, z_1	
3	2, z_2	1, z_1	1, z_1	

	a	b	x	
1	2	1	1	z_1
2	2	3	1	z_1
3	4	1	1	z_1
4	2	3	1	z_2

Fig. 5.31

Prima di passare alla seconda fase del procedimento di sintesi, osserviamo che esistono reti sequenziali le quali devono essere portate in uno stato iniziale particolare se si vuole che funzionino correttamente, mentre per altre è del tutto indifferente ai fini del funzionamento lo stato di partenza. Per le prime il problema può essere risolto in due modi :

- Si può applicare preventivamente una sequenza di ingresso tale da pilotare la rete nello stato di partenza, qualunque sia lo stato in cui si trova; ciò tuttavia è possibile solo se il grafo di stato è *fortemente connesso*, cioè solo se ogni stato interno è raggiungibile a partire da qualunque altro.
- Se il grafo non è fortemente connesso o se lo stato iniziale deve essere raggiunto in un tempo breve e definito indipendentemente dallo stato in cui la macchina si trova, si può aggiungere un nuovo stato di ingresso, detto di **reset**, che per qualunque stato interno genera lo stato iniziale come stato successivo.

Un altro punto da sottolineare è che, come si è visto nel capitolo 4, esistono tabelle di flusso non completamente specificate, tali cioè che per alcuni stati totali non è definito lo stato successivo; ciò può essere dovuto a due ordini di cause:

- uno stato totale (X_i, S_j) non può mai verificarsi a causa di vincoli posti sulla sequenza di

ingresso, pur potendo presentarsi singolarmente sia X_i che S_j .

- Lo stato totale (X_i, S_j) può verificarsi ad un certo passo della sequenza di ingresso, ma nei passi successivi non interessa con quale sequenza di stati di uscita la macchina reagisce alla sequenza di ingresso.

b) La seconda fase parte dalla tabella di flusso per arrivare alla realizzazione della rete. Il passo iniziale consiste nella **codifica degli stati** mediante variabili binarie; spesso ci si riduce alla codifica dei soli stati interni, dal momento che sia gli ingressi che le uscite sono già rappresentate da variabili binarie. Se il numero degli stati interni è n , le variabili per la codifica devono essere in numero pari a $\lceil \log_2 n \rceil$: pertanto se n non è potenza di 2, ci saranno certe combinazioni di valori non significativi e di conseguenza funzioni non completamente specificate. La scelta della codifica non è senza conseguenze, poiché può portare ad una struttura più o meno complessa, ma in questa sede non affrontiamo il problema della scelta ottima.

Il secondo passo consiste nel sostituire ai simboli della tabella di flusso i codici scelti e nel separare stati e uscite in due tabelle dette rispettivamente **tabella delle transizioni degli stati** e **tabella delle uscite**. Da esse si derivano le mappe di Karnaugh per le variabili dello stato successivo e per le variabili dello stato di uscita, sulle quali viene condotto il processo di sintesi della parte combinatoria della rete. In alternativa si ricorre al metodo di Quine-McCluskey. Tuttavia nella sintesi della parte combinatoria occorre tenere conto anche del *modello strutturale* (v. pagg. 134-135), secondo il quale si vuole realizzare la rete e del fatto che sugli anelli di reazione devono essere inseriti *elementi di memoria impulsati*, del tipo esaminato nel capitolo precedente.

Come sappiamo infatti esistono fondamentalmente due categorie di reti sincronizzate: a) quelle in cui tutti gli ingressi e le uscite primarie sono a livelli e b) quelle in cui esiste almeno un ingresso di natura impulsiva, la cui funzione è decidere l'istante di operazione dell'intera rete. In entrambi i casi però l'informazione di ingresso da cui dipende il funzionamento delle reti è quella codificata dai segnali a livelli e su quella deve essere condotta la sintesi. Pertanto, ammesso che le caratteristiche della rete in esame rientrino in una delle due categorie sopra citate, la specifica del modello strutturale dipende:

- 1) dalla specifica dei segnali di ingresso. Per esempio se la rete deve avere un ingresso impulsivo e quindi deve essere del tipo di Mealy ritardato, il modello da scegliere è quello della categoria b).
- 2) Dal tipo di elemento di memoria che si vuole utilizzare. Da esso infatti dipende il numero e l'espressione delle variabili di eccitazione e in definitiva la struttura della parte combinatoria che le genera.

Riguardo a questo secondo punto, la scelta dell'elemento di memoria interviene a trasformare la tabella delle transizioni per ciascuna variabile di stato successivo in una o più tabelle, dette **tabelle di applicazione dei flip-flop**, a seconda che tale elemento abbia una o più variabili di eccitazione. Queste nuove tabelle, in riferimento ai modelli strutturali delle Fig. 5.7 e 5.8, specificano le funzioni:

$$t_{i,j} = f_{i,j}(\mathbf{x}, \mathbf{y}), i = 0, \dots, k-1, j = 0, \dots, r_i-1, \mathbf{x} = \{x_0, x_1, \dots, x_{n-1}\}, \mathbf{y} = \{y_0, y_1, \dots, y_{k-1}\}$$

che governano le variabili di eccitazione $t_{i,j}$, in accordo alla tabella di eccitazione dell'elemento di memoria scelto. Le tabelle di applicazione vengono quindi trasformate nelle corrispondenti mappe di Karnaugh sulle quali viene condotta la sintesi.

Riprendiamo l'esempio della rete che riconosce una sequenza di caratteri, proponendoci di realizzarne la versione di Mealy. Essendo tre gli stati interni, occorrono per la loro codifica due variabili di stato y_1 e y_2 ; facciamo i seguenti assegnamenti, consapevoli tuttavia che una scelta piuttosto che un'altra può portare a funzioni combinatorie più o meno costose:

stato	y_1	y_2
1	0	0
2	0	1
3	1	1
–	1	0

Codifichiamo inoltre gli stati di ingresso con due variabili x_1 ed x_2 , con le corrispondenze $a \equiv 00$, $b \equiv 01$, $x \equiv 11$ e lo stato di uscita con una variabile z con gli assegnamenti $z_1 \equiv 0$, $z_2 \equiv 1$. Con questa codifica la tabella delle transizioni e quella delle uscite diventano:

		x_1, x_2			
		00	01	11	10
y_1, y_2	00	01	00	00	xx
	01	01	11	00	xx
	11	01	00	00	xx
	10	xx	xx	xx	xx
		Y_1, Y_2			

		x_1, x_2			
		00	01	11	10
y_1, y_2	00	0	0	0	x
	01	0	0	0	x
	11	1	0	0	x
	10	x	x	x	x
		z			

Fig. 5.32

Supponiamo di scegliere come elementi di memoria flip-flop JK eccitati sul livello in salita del clock. La parte combinatoria deve generare le funzioni che governano le variabili di eccitazione di tali flip-flop, J_1 , K_1 , J_2 , K_2 , in accordo alla tabella delle transizioni, dalla quale perciò otteniamo le tabelle di applicazione di Fig. 5.33.

Da esse e dalla tabella delle uscite si ottengono le funzioni combinatorie:

$$\begin{aligned} J_1 &= \bar{x}_1 x_2 y_2 \\ K_1 &= 1 \\ J_2 &= \bar{x}_2 \\ K_2 &= x_2 y_1 + x_1 \\ z &= \bar{x}_2 y_1 \end{aligned}$$

con le quali si realizza la rete sequenziale richiesta (Fig. 5.34).

		x_1, x_2			
		00	01	11	10
y_1, y_2	00	0	0	0	×
	01	0	1	0	×
	11	×	×	×	×
	10	×	×	×	×
J_1					

		x_1, x_2			
		00	01	11	10
y_1, y_2	00	×	×	×	×
	01	×	×	×	×
	11	1	1	1	×
	10	×	×	×	×
K_1					

		x_1, x_2			
		00	01	11	10
y_1, y_2	00	1	0	0	×
	01	×	×	×	×
	11	×	×	×	×
	10	×	×	×	×
J_2					

		x_1, x_2			
		00	01	11	10
y_1, y_2	00	×	×	×	×
	01	0	0	1	×
	11	0	1	1	×
	10	×	×	×	×
K_2					

Fig. 5.33

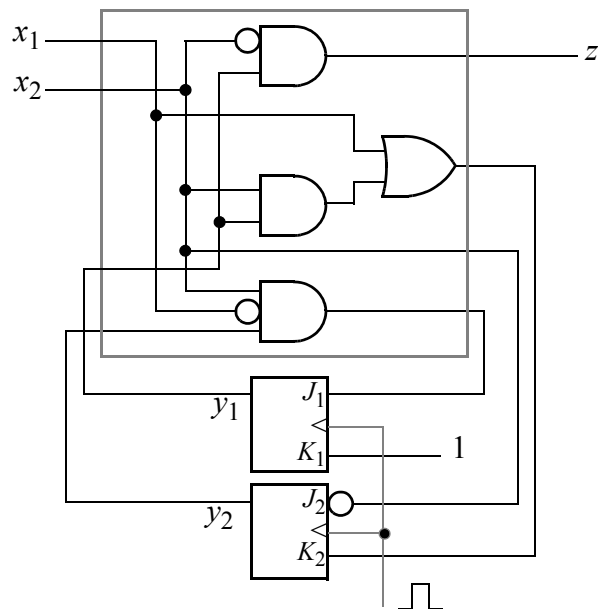


Fig. 5.34

5.4 - Sintesi delle reti sequenziali asincrone

5.4.1 - Tabelle di flusso normali

Le reti sequenziali sincronizzate presentano l'inconveniente che la sequenza di stati di ingresso effettivamente riconosciuta dipende dalla temporizzazione degli impulsi di sincronismo.

Infatti il funzionamento di queste reti è corretto solo se un impulso è applicato dopo che ogni transizione delle variabili di eccitazione degli elementi di memoria è terminata, ossia se si fa trascorrere un tempo non inferiore al ritardo Δ_R con cui la parte combinatoria propaga le variazioni dei suoi ingressi, a partire dall'istante in cui queste si verificano. Tali variazioni possono essere o cambiamenti degli ingressi secondari conseguenti all'aggiornamento dello stato interno determinato dall'ultimo impulso applicato, oppure cambiamenti dello stato di ingresso.

Di tutti i cambiamenti dello stato di ingresso che possano avvenire tra due impulsi consecutivi la rete avverte solo l'ultimo e al contrario il permanere di uno stato di ingresso per un tempo superiore all'intervallo tra due impulsi viene interpretato come una successione di due stati di ingresso identici, del tipo $X_k X_k$.

Questo comportamento nei confronti della sequenza di ingresso è messo in evidenza dal seguente esempio. Supponiamo che una rete riceva in ingresso una successione di stati appartenenti all'insieme $\{X_i, i = 1, \dots, 4\}$, codificati con due variabili x_1 ed x_2 come in tabella:

	x_1	x_2	x_3	x_4
X_1	0	0	1	1
X_2	0	1	1	0

A partire da un certo istante t_0 la sequenza applicata sia $X_1 X_4 X_3 X_2 X_1 X_2 X_3 X_4$ (Fig. 5.35 in alto):

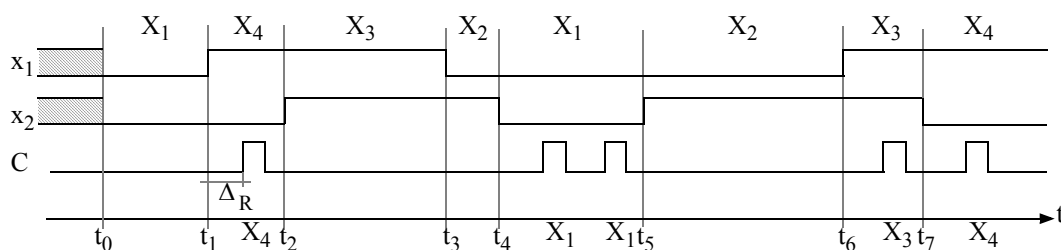


Fig. 5.35

Se gli impulsi C sono applicati secondo la sequenza di figura, la rete risulta sensibile alla successione di stati di ingresso $X_4 X_1 X_1 X_3 X_4$ e non a quella effettivamente presentata. E' evidente che cambiando la temporizzazione cambia la sequenza di ingresso riconosciuta e quindi la risposta della rete.

Si conclude che se si vuole che una rete sincronizzata si comporti nei confronti di una sequenza di stati di ingresso ritenuta significativa nel modo corretto è indispensabile la conoscenza dell'esatta temporizzazione con cui devono essere applicati gli impulsi C .

Ci sono casi in cui questa conoscenza non è disponibile oppure costituisce un vincolo troppo stringente da realizzare e quindi non è possibile utilizzare reti sincronizzate, ma reti sequenziali nelle quali i cambiamenti dello stato interno dipendano esclusivamente dai cambiamenti dello stato di ingresso: le reti adatte a questo scopo sono quelle asincrone, nelle quali non esistono elementi di memoria controllati da segnali di tipo impulsivo.

Tuttavia sappiamo che le reti asincrone, in conseguenza del fatto che tutti i cambiamenti dello stato di ingresso sono significativi, richiedono, per un corretto funzionamento, che siano soddisfatti i seguenti vincoli.

1) Ogni cambiamento di ingresso deve avvenire solo quando lo stato interno è stabile e in conseguenza del nuovo stato di ingresso deve essere raggiunto un nuovo stato stabile prima che la successiva variazione di ingresso possa avere luogo (**proprietà delle transizioni di ingresso a stati stabili**); il minimo intervallo di tempo che intercorre tra due successivi cambiamenti dello stato di ingresso non deve essere inferiore al massimo ritardo con cui lo stato si stabilizza. La proprietà delle transizioni di ingresso a stati stabili tra l'altro può portare ad una rete asincrona più lenta rispetto ad una sincronizzata equivalente dal punto di vista computazionale.

2) Sequenze di ingresso del tipo $X_k X_k$ non possono essere applicate, poichè una rete asincrona non è in grado di distinguere stati di ingresso identici generando stati di uscita diversi.

3) Poichè nel passaggio da uno stato stabile α , relativo ad uno stato di ingresso X_h , ad uno stato stabile β , conseguente ad uno stato di ingresso X_k , una rete asincrona può passare attraverso una successione di stati instabili, mantenendosi costante lo stato di ingresso X_k , è necessario definire come stato di uscita corrispondente allo stato di ingresso X_k quello associato allo stato β , ignorando quelli associati agli eventuali stati instabili intermedi.

Le considerazioni precedenti portano a riconoscere la necessità che la tabella di flusso di una rete asincrona soddisfi ad alcune condizioni affinché essa possa funzionare correttamente.

1) Prima di tutto occorre che in ogni colonna vi sia almeno uno stato stabile. Questa tuttavia è una condizione necessaria ma non sufficiente, come mostra il seguente esempio. Se la rete si trova nello stato stabile S_i con stato di ingresso X_h , in conseguenza della transizione $X_h \rightarrow X_k$ si porta nello stato stabile S_j , con stato di ingresso X_k (Fig. 5.36a). Tuttavia può accadere anche che la rete, dallo stato stabile S_i con stato di ingresso X_h , in conseguenza della stessa transizione di ingresso, si immetta nella sequenza ciclica $S_i \rightarrow S_j \rightarrow S_i \rightarrow \dots$ con stato di ingresso X_k , rimanendovi finchè non si verifica una successiva transizione $X_k \rightarrow X_l$, a seguito della quale peraltro la rete si stabilizzerà in uno stato non prevedibile, dipendendo dallo stato attuale in cui la transizione trova la rete (Fig. 5.36b):

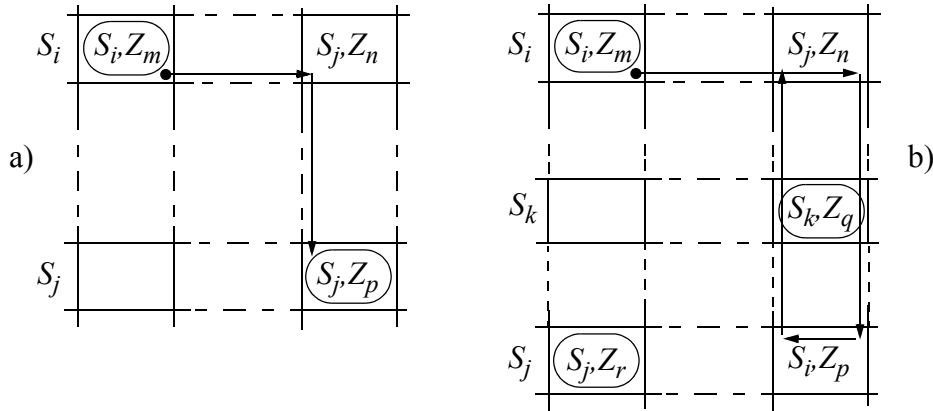


Fig. 5.36

Segue da questa condizione, e dal fatto che l'uscita di una rete asincrona è sempre quella relativa ad uno stato stabile, che nella tabella di flusso le righe che non contengono stati stabili possono essere eliminate e le uscite associate a stati instabili possono essere qualsiasi, perchè prive di significato. Inoltre se vi sono condizioni di indifferenza sulle uscite, esse possono essere utilizzate con vantaggio per la riduzione della tabella stessa (Fig. 5.37).

	X_1	X_2		X_1	X_2
S_1	(S_1, Z_2)	$S_2, -$	\Rightarrow	S_1	(S_1, Z_2) $S_3, -$
S_2	$S_5, -$	$S_3, -$		S_3	$S_4, -$ (S_3, Z_1)
S_3	$S_4, -$	(S_3, Z_1)		S_4	(S_4, Z_2) $S_5, -$
S_4	(S_4, Z_2)	$S_5, -$		S_5	(S_5, Z_1) $S_4, -$
S_5	(S_5, Z_1)	$S_4, -$			

Fig. 5.37

2) In secondo luogo, dal momento che lo stato di ingresso può essere aggiornato solo a stati interni stabili, conviene che la tabella di flusso sia strutturata in modo da ridurre per quanto possibile il tempo T_{Stab} richiesto per passare da uno stato stabile all'altro ed il tempo T_Z necessario perchè lo stato di uscita assuma il valore corrispondente al nuovo stato di ingresso. Se la rete passa attraverso n stati instabili prima di arrivare a quello stabile, è facile verificare che vale la relazione $T_{Stab} = (n+1)\Delta_R$. Il ritardo T_{Stab} è minimo quando la rete attraversa al più uno stato instabile, ossia è $T_{Stab_{min}} = 2\Delta_R$. Invece T_Z è minimo se lo stato di uscita raggiunge il valore corrispondente allo stato di ingresso con un ritardo pari a quello di propagazione della rete combinatoria, ossia $T_{Z_{min}} = \Delta_R$. Affinchè T_Z sia minimo, tutti gli stati di uscita devono essere ovviamente significativi. Per avere T_{Stab} e T_Z minimi, è necessario che in ogni colonna

ad uno stato instabile corrisponda uno stato stabile dello stesso nome e che lo stato di uscita associato allo stato instabile sia specificato e uguale a quello associato allo stato stabile, ossia:

$$f(X_i, S_j) = S_k \mid S_j \neq S_k \Rightarrow f(X_i, S_k) = S_k \text{ e } g(X_i, S_j) = g(X_i, S_k), \forall X_i$$

Le tabelle di flusso strutturate in modo da soddisfare i punti 1) e 2) e quelle che soddisfano il punto 1) e per le quali l'uscita dipende solo dallo stato interno (macchine di Moore) sono dette **tabelle normali** e le reti corrispondenti **reti asincrone normali**.

A differenza delle reti sincronizzate che possono essere realizzate sia secondo il modello di Mealy che secondo il modello di Moore, dando la preferenza al secondo quando si vuole una rete il cui lo stato di uscita sia sempre significativo, di solito ha poco senso porsi il problema del modello nel caso delle reti asincrone; infatti, dovendo realizzare reti normali, lo stato di uscita è sempre significativo, dal momento che in ogni colonna della tabella normale l'uscita di uno stato instabile è sempre uguale a quella dello stato stabile corrispondente.

5.4.2 - Sintesi delle reti asincrone normali

La sintesi delle reti asincrone passa attraverso le stesse fasi viste per le reti sincronizzate, ossia costruzione della tabella di flusso dalla descrizione a parole e realizzazione della rete secondo il modello strutturale prescelto. Tuttavia è necessario fare in modo che la tabella di flusso sia costruita rispettando i vincoli che la rendono normale. Tali vincoli si trasferiscono nel grafo degli stati nel modo seguente:

a) Per ogni arco X_k/Z_m che entra in un nodo S_i e che proviene da un nodo S_j , $i \neq j$, c'è un arco omonimo che esce da S_i e si richiude su S_i stesso (Fig. 5.38):

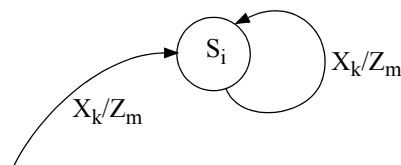


Fig. 5.38

b) Non possono esistere due archi che entrano in S_i provenendo da nodi diversi etichettati con lo stesso stato di ingresso e stati di uscita diversi; infatti in caso contrario dovrebbero esistere due archi uscenti e rientranti in S_i etichettati con i due stati di uscita e lo stesso stato di ingresso, il che non può essere, in quanto da un nodo non possono uscire due archi etichettati con lo stesso stato di ingresso e stati di uscita diversi (Fig. 5.39).

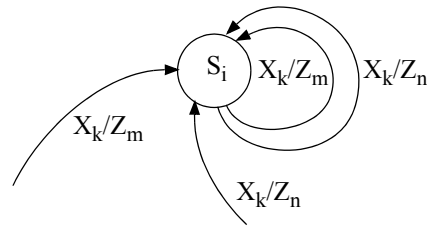


Fig. 5.39

Per esempio si voglia costruire una rete asincrona che riconosce la sottosequenza di stati di ingresso *aba* ogni volta che essa si presenta in una sequenza continua costruita con gli stati *a*, *b* e *x*; si esclude che possano presentarsi due stati di ingresso identici consecutivi.

Il grafo degli stati e la relativa tabella di flusso sono illustrati in Fig. 5.40.

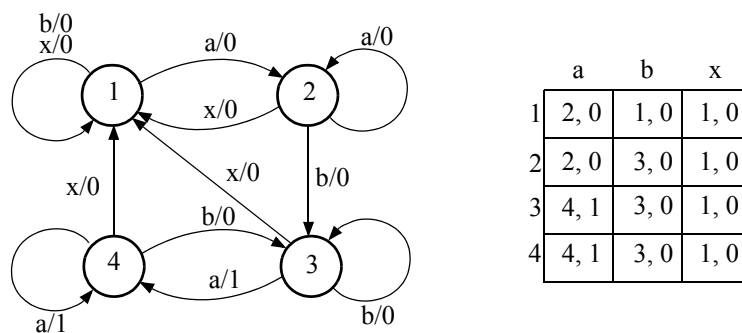


Fig. 5.40

Il procedimento di sintesi procede da questo punto come nel caso delle reti sincronizzate, salvo osservare che, non essendovi di norma latch o flip-flop sui percorsi di retroazione, viene a mancare la parte relativa alla definizione delle tabelle di applicazione dei flip-flop.

Vale la pena osservare la somiglianza tra questo grafo (di Mealy) e quello di Fig. 5.30 (di Moore) per lo stesso problema: questa osservazione rende ragione in modo esemplare a quanto affermato circa il fatto che modellare una rete asincrona secondo Mealy o secondo Moore è una questione priva di importanza.

Da ultimo facciamo alcune considerazioni sulla la velocità di risposta delle reti asincrone rispetto a quelle sincronizzate. Il tempo di ciclo minimo per una rete sincronizzata è come sappiamo $T_s = \Delta_R + \Delta_M$, mentre per una rete asincrona esso è $T_a = 2\Delta_R$, essendo Δ_R il massimo ritardo della parte combinatoria. Supponendo che Δ_R sia lo stesso per i due tipi di reti, risulta $T_a < T_s$ solo se $\Delta_R < \Delta_M$; se questa condizione non è verificata, cosa che avviene normalmente in pratica, la rete sincronizzata è più veloce di quella asincrona, contrariamente alla convinzione abbastanza diffusa che le reti asincrone siano le reti più veloci.

ESERCIZI

- 1) Progettare una rete sequenziale sincronizzata che in una sequenza di tre bit riconosce se il bit 0 è contenuto esattamente due volte.
- 2) Progettare una rete sequenziale con un ingresso ed una uscita la quale assume valore 1 se e solo se l'ingresso al tempo t è l'opposto di quello al tempo $t-1$.
- 3) Una rete sequenziale possiede un ingresso x ed una uscita z la quale vale 1 se l'ingresso al tempo t_i vale 1, mentre agli istanti t_{i-2} e t_{i-1} valeva rispettivamente 1 e 0. Progettare la rete.
- 4) Per il problema dell'esercizio 3 realizzare una rete asincrona.
- 5) Un decodificatore seriale possiede un ingresso a livelli r ed un ingresso ad impulsi x e quattro uscite z_0, z_1, z_2, z_3 . Quando $r = 0$ tutte le uscite sono 0, indipendentemente dal valore di x . Quando $r = 1$ il funzionamento del decodificatore è descritto dalla tabella seguente:

sequenza di valori di x	z_0	z_1	z_2	z_3
0 0	1	0	0	0
0 1	0	1	0	0
1 0	0	0	1	0
1 1	0	0	0	1

- 6) Una rete sequenziale nel modo fondamentale possiede due ingressi x_1 e x_2 ed una uscita z la quale cambia solo nel caso in cui gli ingressi cambiano da (0, 0) a (0, 1). Progettare la rete.
- 7) In una stanza possono trovarsi una o due persone; la luce nella stanza deve essere accesa se c'è almeno una persona, spenta se non c'è nessuno. Le due persone possono entrare o uscire solo una per volta, ma mentre una entra l'altra può uscire. Realizzare una rete sequenziale che simula la situazione descritta.
- 8) Realizzare un circuito sequenziale che rileva gli errori nel codice in eccesso di tre, trasmesso in ingresso a partire dal bit meno significativo (il più a destra), in modo che se la sequenza dei valori dei quattro bit trasmessi è diversa da una di quelle valide, l'uscita del circuito diventa 1, altrimenti rimane 0. Si suppone che la sequenza di ingresso venga interrotta non appena il circuito scopre un errore.
- 9) Un circuito sequenziale asincrono con due ingressi x_1, x_2 ed una uscita z funziona nel seguente modo. Quando $x_1 = 1$, z cambia da 0 a 1 durante la transizione di x_2 da 0 a 1; quando $x_2 = 1$, z cambia da 1 a 0 durante la transizione di x_1 da 1 a 0; z diventa 0 se x_1 e x_2 cambiano insieme. Ogni altro evento di ingresso lascia immutata l'uscita. Progettare il circuito.

- 10) Progettare un circuito sequenziale con un ingresso ed una uscita la quale vale 1 se cinque configurazioni di ingresso consecutive costituiscono una sequenza contenente esattamente tre 1 e due 0 e i primi due valori sono 1. Il circuito ritorna nello stato iniziale quando viene riconosciuta una sequenza completa oppure la sequenza si interrompe.
- 11) Realizzare un circuito sequenziale asincrono con due ingressi x_1 e x_2 e due uscite z_1 e z_2 , tale che se uno solo degli ingressi cambia, l'uscita ($z_1 z_2$) assume la configurazione di ingresso; se entrambi gli ingressi rimangono stabili o cambiano insieme, l'uscita rimane inalterata.
- 12) Un circuito sequenziale possiede due ingressi ad impulsi x_1 e x_2 ed una uscita a livelli z . Ogni volta che arriva un impulso sull'ingresso x_1 e, dopo tale impulso, due impulsi arrivano su x_2 , ma nessun altro su x_1 , l'uscita diventa 1, altrimenti rimane 0. Quando l'uscita è diventata 1, rimane tale fino al primo impulso su x_2 , che la riporta a 0. Progettare il circuito.
- 13) Realizzare una rete sequenziale con un ingresso ed una uscita, la quale vale 1 quando viene riconosciuta la sequenza di ingresso $01xx...x01$, dove $xx...x$ è una qualsiasi stringa, eventualmente di lunghezza nulla, di 0 e di 1.
- 14) L'incrocio tra una strada a grande traffico ed un attraversamento pedonale è regolato da un semaforo che può essere attuato dai pedoni che intendono attraversare la strada. In assenza di richiesta da parte di pedoni, il semaforo sulla strada principale è sempre verde e quello rivolto verso l'attraversamento rosso. Se un pedone fa richiesta di attraversare ed è trascorso un certo tempo prefissato dall'ultima richiesta, il semaforo principale deve passare al rosso, mentre quello del passaggio pedonale deve diventare verde, rimanendovi per un determinato intervallo di tempo; passato tale tempo il semaforo pedonale ritorna rosso e quello della strada principale verde. Progettare una rete sequenziale che controlla il semaforo.
- 15) Un circuito sequenziale sincronizzato ha un ingresso a livelli x ed una uscita ad impulsi z . Il valore dell'ingresso può cambiare solo in assenza dell'impulso di clock. Se x si mantiene ad 1 per tre impulsi consecutivi del clock, l'uscita diventa 1 in corrispondenza del successivo impulso di clock. In ogni altra circostanza l'uscita rimane 0.
 - a) Sintetizzare il circuito facendo uso di flip-flop JK.
 - b) Mostrare il funzionamento del circuito attraverso un diagramma temporale.
- 16) Un distributore automatico di cioccolatini accetta monete da 100 e 200 lire per un cioccolatino del costo di 300 lire e dà insieme il resto se la somma depositata è superiore a 300 lire.
Sapendo che per ogni moneta inserita vengono generati due segnali x_1, x_2 secondo il codice mostrato nella seguente tabella, si vuole progettare una rete sequenziale nel modo fondamentale con ingressi x_1 e x_2 e con due uscite z_1 e z_2 , che riconosce la cifra versata ed effettua le seguenti operazioni: 1) se la cifra è pari al costo, fornisce il cioccolatino ($z_1 z_2 = 10$); 2) se la cifra è superiore al costo, fornisce cioccolatino e resto

($z_1z_2=11$); 3) se una moneta non è valida, restituisce l'intero deposito, ovviamente senza il cioccolatino ($z_1z_2=01$); 4) altrimenti mantiene le uscite $z_1z_2=00$.

moneta	x_1	x_2
nessuna	0	0
100 lire	0	1
200 lire	1	0
falsa	1	1

a) Disegnare il grafo di stato e la tabella di flusso della rete.

b) Sintetizzare la rete utilizzando per la parte combinatoria una ROM.

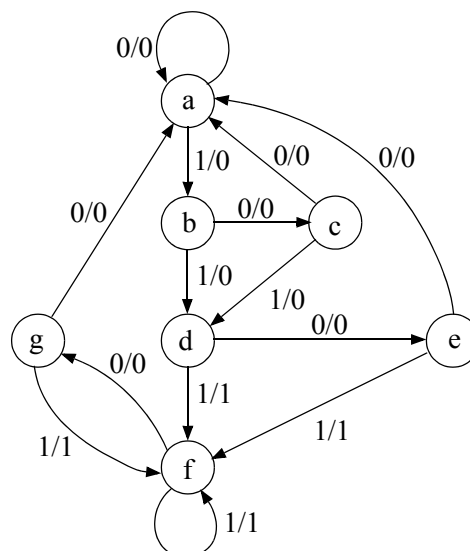
Si suppone che x_1 e x_2 ritornino automaticamente a zero prima che sia inserita una nuova moneta, ma ovviamente non prima che il circuito si sia stabilizzato in un nuovo stato. Infatti se gli ingressi avessero ancora il valore acquistato per effetto dell'ultima moneta quando una nuova viene depositata, accadrebbe che... @#*&!!!, e ciò è dovuto al fatto che una rete asincrona...

- 17) Costruire il diagramma degli stati di una macchina di Moore che ha due stati di ingresso, $I=\{0,1\}$, due stati di uscita $Z=\{0,1\}$ ed il seguente comportamento: la sequenza degli stati di ingresso è esaminata dalla macchina a gruppi di quattro stati consecutivi e l'uscita va ad 1 per un periodo di clock se si verifica una delle due sequenze 0110 o 1001.

Esempio: sequenza di ingresso 01011000011010...

sequenza di uscita 00000000000010...

- 18) Sintetizzare con flip-flop JK la seguente macchina sequenziale, minimizzandola se necessario:



- 19) Quando un canale di trasmissione binario seriale opera correttamente, tutti i blocchi di 0 consecutivi sono di lunghezza pari e tutti i blocchi di 1 consecutivi sono di lunghezza dispari.

Sintetizzare una rete sequenziale di Mealy, usando FF di tipo T, che produca una uscita z ad 1 per un periodo di clock quando si rileva un comportamento erraneo. Disegnare il circuito usando porte NAND e costruire inoltre la tabella di stato della macchina di Moore simile.

Esempio x 001000111011000...
 z 000000100010100...

- 20) Progettare una rete sequenziale sincrona che simula il gioco dell'oca con un solo giocatore, il cui tracciato è riportato nella seguente figura:

P	1 vai a 3	2	3
			4 vai a 7
			5 stai fermo un lancio
A	8 torna alla partenza	7	6 torna a 3

Il gioco consiste nell'arrivare esattamente nella casella finale A, a partire dalla casella iniziale P, lanciando successivamente un dado che riporta sulle facce solo i valori 1, 2 e 3; ad ogni lancio il giocatore avanza di un numero di caselle corrispondente al numero estratto e rispettando le condizioni associate alle varie caselle del tracciato. Ogni eccesso di punteggio rispetto alla casella di arrivo fa retrocedere di un numero corrispondente di caselle; ad esempio se il giocatore si trova nella casella 7 e il lancio fa uscire 3, si deve tornare alla casella 8.

Utilizzare come elementi di memoria flip-flop di tipo T. La rete ha come ingresso il valore estratto dal lancio del dado e come uscita una variabile che assume valore 1 quando viene raggiunta esattamente la casella di arrivo.

- 21) Ad un incrocio tra due strade A e B è situato un semaforo che regola il traffico operando ciclicamente nel seguente modo:

- mantiene lo stato S_0 (A rosso, B giallo) per 1 clock, quindi passa in S_1 ;
- mantiene S_1 (A verde, B rosso) per almeno 3 clock e comunque fino a quando un veicolo non è presente in direzione B, condizione rilevata da un sensore b , posto lungo B, che assume valore 1; quindi passa in S_2 ;
- mantiene S_2 (A giallo, B rosso) per 1 clock, quindi passa in S_3 ;

- mantiene S_3 (A rosso, B verde) per almeno 3 clock e comunque fino a quando un veicolo non è presente in direzione A, condizione rilevata da un sensore a, posto lungo A, che assume valore 1; quindi passa in S_0 .
- a) Descrivere il diagramma degli stati di un circuito sequenziale di Moore che comanda il semaforo.
- b) Sintetizzare il circuito utilizzando flip-flop JK. È sufficiente ricavare le funzioni di eccitazione di uno solo dei flip-flop e quelle delle uscite z_1 e z_2 che comandano il semaforo secondo la tabella:

stato	codice di comando
S_0	00
S_1	01
S_2	10
S_3	11

- c) In alternativa sintetizzare la rete utilizzando un contatore binario.

22) Un tale possiede due autorimesse A, B in cui parcheggia le sue auto. Egli vuole installare un dispositivo di apertura automatica che, oltre a toglierli la seccatura dell'apertura e chiusura manuale, sia anche capace di evitargli di cercare quale autorimessa contenga l'auto quando vuole uscire e quale invece è libera quando vuole parcheggiare. Il dispositivo deve disporre di due tasti P ed U; il tasto P verrà premuto ($P=1$) per parcheggiare e dovrà aprire la prima (nell'ordine A, B) autorimessa disponibile ad accogliere un'auto, il tasto U verrà premuto ($U=1$) per uscire e dovrà aprire la prima (nell'ordine A, B) autorimessa che contiene un'auto. Il dispositivo deve essere costruito in modo da impedire la pressione contemporanea di entrambi i tasti P ed U. La chiusura avverrà automaticamente dopo un tempo prefissato.

Progettare il circuito sequenziale sincrono che realizza il dispositivo ipotizzando che una volta aperta la porta, l'auto sia effettivamente parcheggiata o prelevata. I segnali di uscita siano rappresentati da due variabili Y e Z codificate come segue:

00 nessuna apertura

01 apertura dell'autorimessa B

10 apertura dell'autorimessa A

11 accensione di una luce rossa per avvertire che entrambe le autorimesse sono piene, quando si vuole parcheggiare o entrambe vuote, quando si vuole prelevare un'auto.

Usare flip-flop di tipo D e codificare gli stati nella maniera seguente: 00 autorimesse vuote, 01 presente solo un'auto nell'autorimessa B, 10 presente solo un'auto nell'autorimessa A, 11 autorimesse piene.

23) In un impianto di autolavaggio il percorso delle auto è delimitato all'entrata e all'uscita da due sbarre A e B il cui movimento è comandato da un dispositivo che riceve come ingressi i segnali di due fotocellule F_A e F_B situate in corrispondenza delle sbarre stesse. Il comportamento del dispositivo è il seguente.

- La sbarra A viene chiusa quando due auto sono passate oltre la fotocellula F_A e riaperta quando entrambe le auto sono uscite.
- La sbarra B viene aperta quando due auto sono all'interno dell'autolavaggio e richiusa quando sono uscite.

Progettare il circuito di comando come rete sequenziale asincrona.

24) Data la tabella di flusso di figura che descrive una rete sequenziale sincronizzata, si supponga di partire dallo stato totale (X_2, S_3) e si determini la più lunga sequenza di stati di ingresso (con $X_i \neq X_{i-1}$) per la quale le uscite hanno sempre valore corretto, anche nell'intervallo di tempo tra un impulso di clock ed il cambiamento dell'ingresso.

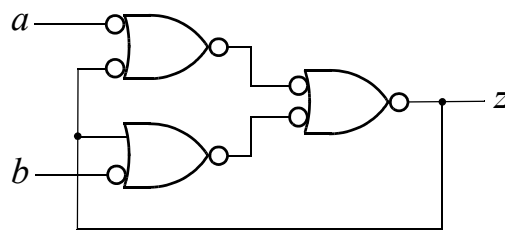
	X_1	X_2	X_3	X_4
S_1	$S_2, 1$	$S_4, 2$	$S_1, 3$	$S_3, 2$
S_2	$S_3, 2$	$S_1, 1$	$S_4, 3$	$S_3, 2$
S_3	$S_2, 3$	$S_4, 1$	$S_1, 2$	$S_2, 1$
S_4	$S_1, 3$	$S_2, 3$	$S_3, 1$	$S_2, 2$

25) Progettare una rete sequenziale con due ingressi a livelli x ed y ed una uscita z pure a livelli; la rete esegue la somma logica di x ed y , se lo stato di ingresso attuale è uguale a quello precedente, altrimenti esegue l'or esclusivo di x ed y .

26) Data la seguente tabella di flusso parzialmente specificata, trovare una tabella ridotta compatibile con essa e si disegni lo schema della rete corrispondente.

	000	010	101	110	111
a	$-, -$	$a, 0$	$a, 1$	$c, -$	$a, 0$
b	$a, 1$	$b, 0$	$b, -$	$a, 1$	$c, -$
c	$-, -$	$c, -$	$-, -$	$-, -$	$d, -$
d	$c, 1$	$d, -$	$e, 1$	$-, -$	$b, 0$
e	$b, -$	$e, 0$	$d, -$	$d, -$	$e, -$

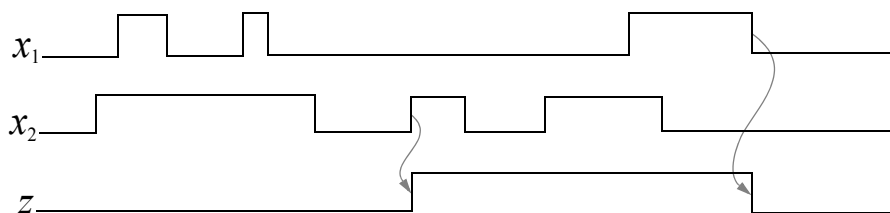
27) Studiare il comportamento della rete di figura, specificando se essa è affetta o meno da oscillazioni, e si disegni l'andamento temporale dell'uscita in funzione degli ingressi a e b .



28) L'alettone di coda di un aeromodello può stare in tre posizioni, diritto, alto, basso, ed è comandato a distanza da impulsi radio nel seguente modo. Se l'alettone si trova piegato in alto o in basso, quando arriva un impulso si riporta diritto; se invece è diritto, quando arriva un impulso si piega verso il basso se in precedenza era in alto, o viceversa si piega verso l'alto se era in basso. Progettare una rete sequenziale asincrona che comanda il movimento dell'alettone. Realizzare inoltre la rete in forma sincronizzata senza ingressi (l'impulso viene applicato all'ingresso di sincronismo degli elementi di memoria) e fare vedere che essa è facilmente ottenibile anche dalla tabella di flusso della rete asincrona. (Suggerimento: non è necessario ricercare le classi di compatibilità).

29) Progettare in forma NAND un circuito sequenziale con tre stati di ingresso a, b, c e due stati di uscita z_1, z_2 , il quale riconosce con lo stato di uscita z_2 la sottostringa di ingresso nella quale un numero pari di stati b è seguita, anche non immediatamente, da un numero dispari di stati c .

30) Progettare una rete sequenziale sincronizzata con ingressi x_1, x_2 ed uscita z , che funziona nel seguente modo (vedi figura).

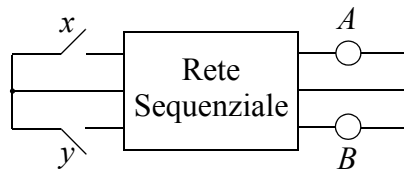


Se sull'ingresso x_1 si succedono due 1, mentre x_2 è 1 e successivamente si presenta 1 sull'ingresso x_2 , essendo 0 l'ingresso x_1 , l'uscita diventa 1 e tale rimane finché, passando x_1 da 1 a 0 con x_2 a 0, ritorna anch'essa a 0. L'uscita rimane zero per tutte le eventuali condizioni di ingresso non rispecchiate dal diagramma. Nel progetto si faccia uso di flip-flop T e almeno di un flip-flop JK.

31) Scrivere la tabella di flusso di una rete sequenziale sincronizzata, con ingressi x_1, x_2 ed una uscita impulsiva z , la quale riconosce se all'ingresso x_2 si presenta un impulso $0 \rightarrow 1 \rightarrow 0$ esattamente nell'intervallo di tempo definito dal presentarsi all'ingresso x_1 di due

impulsi $0 \rightarrow 1 \rightarrow 0 \rightarrow 1 \rightarrow 0$. Tutti gli impulsi possono avere durata arbitraria.

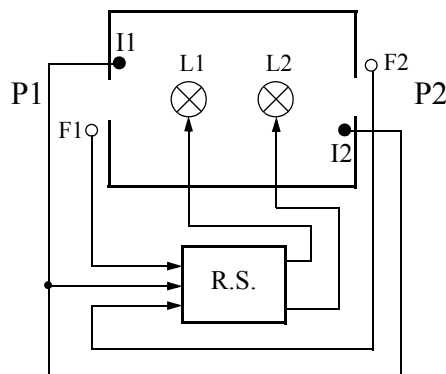
- 32) Si vuole comandare l'accensione e lo spegnimento di due lampade A e B con due interruttori x ed y , azionabili solo singolarmente, mediante una rete sequenziale che opera con le seguenti modalità:



- quando x ed y sono entrambi aperti le lampade A e B sono spente;
- la chiusura di x o di y indifferente provoca l'accensione della lampada A; la sua apertura ne provoca lo spegnimento;
- la lampada B si accende quando anche l'altro interruttore viene chiuso;
- dal momento in cui entrambi gli interruttori sono chiusi e finché non vengono di nuovo aperti entrambi, la lampada A è comandata dal secondo interruttore azionato, la lampada B dal primo.

Scrivere la tabella di flusso della rete, precisando se questa può essere realizzata sia in modo sincrono che in modo asincrono e, in caso affermativo, se ne realizzi la versione asincrona.

- 33) In una stanza ci sono due lampade L1 ed L2, comandate da due fotocellule F1 ed F2 e da due interruttori I1 e I2 situati in corrispondenza delle porte di accesso P1 e P2, come in figura:



Le fotocellule e gli interruttori fanno accendere e spegnere le lampade secondo il seguente schema. Nella stanza possono trovarsi una o al più due persone insieme, ma possono entrare solo una per volta. Una fotocellula genera un impulso solo quando rileva una persona che sta entrando, invece un interruttore genera un impulso solo quando una persona esce; i due interruttori sono collegati tra di loro in modo che sia indifferente quale dei due viene attivato.

Quando la stanza è vuota le lampade devono essere spente. Se entra una persona da P1, fa accendere L1; se entra da P2, fa accendere L2. Quando la persona esce, non importa

da quale porta, fa spegnere la lampada che era accesa.

Se nella stanza c'è già una persona e ne entra una seconda dalla stessa porta, non provoca alcun effetto; se invece la seconda persona entra dalla porta opposta, fa accendere la lampada che si trovava spenta. Quando una delle due persone esce, non accade nulla se è accesa una sola lampada; altrimenti si spegne la lampada opposta a quella che la persona aveva fatto accendere entrando. Le persone possono lasciare la stanza solo in ordine inverso a quello di entrata.

Scrivere la tabella di flusso ridotta di una rete sequenziale asincrona, eventualmente non normale, che riceve in ingresso gli impulsi generati dalle fotocellule e dai due interruttori collegati e produce due uscite per fare accendere e spegnere L1 ed L2 rispettivamente.

- 34) Facendo uso di flip-flop T, progettare una rete sequenziale sincronizzata di Moore la quale riconosce con un impulso in uscita se in ingresso, a partire dall'istante t_i , si presenta la sequenza 0 1 0, essendo stato l'ingresso uguale a 1 1 0 rispettivamente negli istanti t_{i-3} , t_{i-2} , t_{i-1} . Le variazioni dell'ingresso sono sincrone con il clock.
- 35) Una rete sequenziale asincrona ha il seguente funzionamento. Quando l'ingresso cambia da 0 ad 1, l'uscita della rete cambia pure da 0 ad 1 e rimane 1 finché l'ingresso rimane 1.
Quando l'ingresso cambia da 1 a 0, l'uscita cambia da 1 a 0, quindi di nuovo ad 1 e infine a 0.
Se l'ingresso cambia nuovamente da 0 ad 1 quando l'uscita è definitivamente diventata 0, la rete riprende il funzionamento proprio, altrimenti attende che l'ingresso ritorni a 0, conservando immutata l'uscita, dopodiché riprende a funzionare propriamente dall'inizio.
Dire se si tratta di una rete normale e svilupparne il progetto.
- 36) Scrivere la tabella di flusso di una rete sequenziale che accetta in ingresso le cifre tra 0 e 9 e denuncia in uscita il riconoscimento di un qualunque numero compreso tra la vostra data di nascita D, scritta nel formato ggmmaa, ed il numero D+10.
- 37) Progettare una rete sequenziale con due ingressi, x_1 ed x_2 , ed una uscita z , la quale funziona nel modo seguente; quando x_1 diventa 1 e quindi 0, mentre x_2 è 0, e successivamente x_2 diventa 1 per due volte mentre x_1 è 1, l'uscita diventa 1 quando x_1 cade a 0 e rimane tale fino al primo 1 su x_2 che si presenta mentre x_1 è 0.
- 38) Progettare una rete sequenziale asincrona normale con tre stati di ingresso a , b , c , che riconosce se in una sequenza continua di stati di ingresso sono presenti le sottosequenze acb oppure bac , anche interallacciate.
- 39) Progettare una rete sequenziale sincronizzata con un ingresso x ed una uscita z , che funziona nel seguente modo. Quando in ingresso si presenta una variazione $0 \rightarrow 1$ oppure $1 \rightarrow 0$ e all'istante precedente l'ingresso era rispettivamente 0 oppure 1, l'uscita diventa 1; in tutti gli altri casi l'uscita rimane 0. Nel progetto si usino flip-flop JK.
Esempio: $x \dots 1110100110001\dots$

$y \dots 0001000101001\dots$

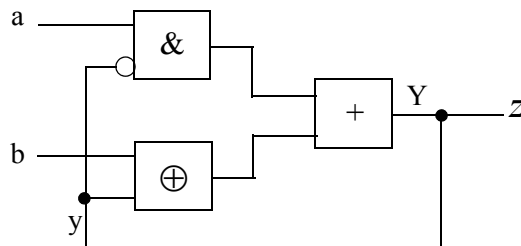
- 40) Progettare una rete sequenziale sincronizzata con un ingresso x , sincrono con il clock, ed una uscita z , nella quale l'uscita permane ad 1 per un tempo pari al numero di periodi di clock in cui l'ingresso rimane costantemente a zero o ad uno, diminuito di due.

Esempio:

$x: \dots 0000011110100111100100011111\dots$

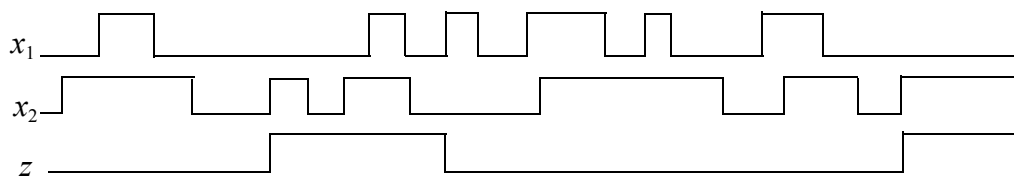
$z: \dots 0011100100000011000001001111\dots$

- 41) Ricavare la tabella di flusso della rete di figura e analizzarne il funzionamento:



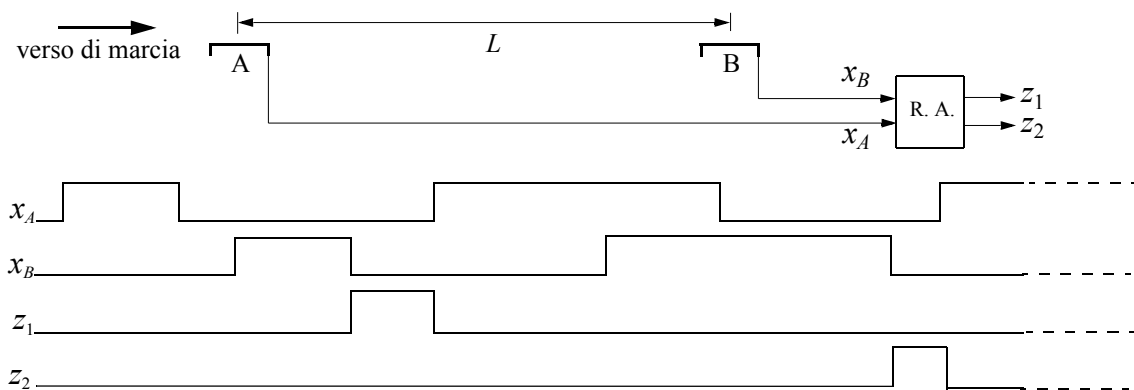
- 42) Progettare una rete sequenziale con due ingressi x_1, x_2 ed una uscita z , la quale diventa 1 quando entrambi gli ingressi diventano 1, qualunque sia l'ordine secondo il quale essi assumono tale valore (insieme, oppure prima x_1 e poi x_2 o viceversa). La rete può essere realizzata come asincrona? Motivare la risposta.

- 43) Progettare una rete sequenziale sincronizzata con due ingressi x_1, x_2 ed una uscita z che funziona nel modo seguente. Quando l'ingresso x_1 cambia da 0 ad 1, mentre x_2 è fisso ad 1, l'uscita assume valore 1 al prossimo fronte $0 \rightarrow 1$ che si presenta su x_2 , mentre x_1 è 0 e tale rimane fino al momento in cui x_1 cambia di nuovo da 0 ad 1, mentre x_2 è 0 (v. figura).



- 44) Una compagnia di traghetti vuole distinguere automaticamente al momento dell'imbarco le auto di passo inferiore ad un certo valore L da quelle di passo uguale o superiore. Per questo lungo la corsia di imbarco vengono posizionate due piattaforme mobili A e B, a distanza L l'una dall'altra; quando un'auto esercita con le ruote anteriori una pressione su una piattaforma, si genera un segnale x_i ($i = A$ o B) di valore 1, che viene riportato a 0 solo quando la piattaforma viene di nuovo premuta dall'auto con le ruote posteriori. I due segnali x_A e x_B sono inviati in ingresso ad una rete asincrona con due uscite z_1 e z_2 che assumono valore 10 o 01 rispettivamente nel caso di passo inferiore a L oppure uguale o maggiore. Lo stato di uscita si mantiene fino all'arrivo di

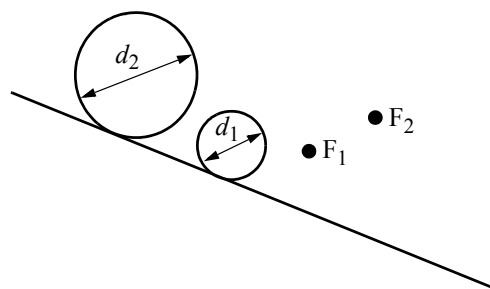
un'altra auto che lo riporta a 00. (v. figura). Progettare la rete asincrona



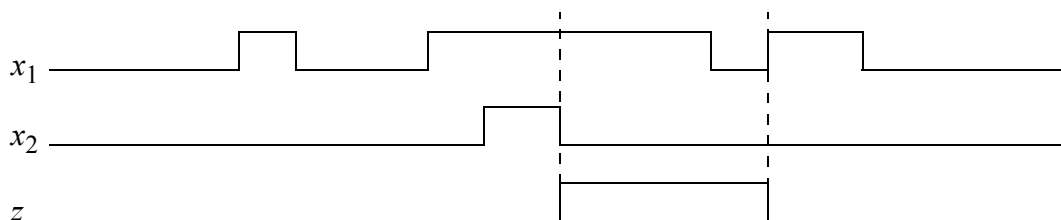
- 45) Progettare una rete sequenziale sincronizzata con due ingressi x_1 ed x_2 ed una uscita z , la quale funziona come segue. Quando su x_1 si presenta la sequenza 011 e corrispondentemente su x_2 si hanno un numero dispari di 1, l'uscita diventa 1, altrimenti resta 0. Si utilizzino flip-flop di tipo JK.
- 46) Progettare una rete sequenziale sincronizzata con due ingressi x_1 e x_2 ed una uscita z , la quale funziona nel modo seguente. Ad ogni impulso di clock la rete legge una nuova coppia di ingresso (x_1, x_2); quando in ingresso viene riconosciuta la sequenza 10, 00, 01, l'uscita viene posta ad 1 per un periodo di clock quindi ritorna a zero e rimane zero per qualunque altra situazione. La rete riconosce anche sequenze parzialmente sovrapposte. Nel progetto utilizzare flip-flop JK.
- 47) Una rete sequenziale sincronizzata possiede due ingressi x_1 e x_2 ed una uscita z e funziona da ritardo programmabile secondo il seguente schema. Se $x_2 = 0$, l'uscita al generico istante t vale $z(t) = x_1(t-1)$. Se $x_2 = 1$, l'uscita al generico istante t vale $z(t) = x_1(t-2)$. Realizzare la rete utilizzando flip-flop JK.
- 48) Una rete sequenziale sincronizzata possiede due ingressi x_1 e x_2 e due uscite z_1 e z_2 e funziona secondo il seguente schema. Se $x_2 = 0$, l'uscita vale 10 se sull'ingresso x_1 si presenta la sequenza 110. Se $x_2 = 1$, l'uscita vale 01 se sull'ingresso x_1 si presenta la sequenza 011. In tutti gli altri casi l'uscita vale 11. Realizzare la rete utilizzando flip-flop JK.
- 49) Una rete sequenziale sincronizzata possiede due ingressi x_1 e x_2 ed una uscita z e funziona secondo il seguente schema. Quando $x_2 = 0$, e sull'ingresso x_1 si presenta la sequenza 101, l'uscita va ad 1 quando x_2 va ad 1 e vi rimane fino a quando il primo tra x_1 e x_2 torna a zero. In tutti gli altri casi l'uscita rimane 0. Realizzare la rete utilizzando flip-flop JK.
- 50) Progettare una rete sequenziale con due ingressi, x_1 ed x_2 , ed una uscita z , la quale

funziona nel modo seguente; quando x_1 diventa 1 e quindi 0, mentre x_2 è 0, e successivamente x_2 diventa 1 per due volte mentre x_1 è 1, l'uscita diventa 1 quando x_1 cade a 0 e rimane tale fino al primo 1 su x_2 che si presenta mentre x_1 è 0.

- 51) In una fonderia, per separare cilindri di acciaio in due gruppi in funzione dei diametri d_1 ed d_2 , essi vengono fatti rotolare su un piano inclinato lungo il quale intercettano una o entrambe le fotocellule F_1 ed F_2 come in figura. Gli impulsi generati dalle fotocellule quando sono oscurate sono elaborati da una rete sequenziale che effettua il riconoscimento dei cilindri e in conseguenza produce un'uscita appropriata per stabilire percorsi differenti per i cilindri di diametro d_1 e per quelli di diametro d_2 . In assenza di cilindri sul piano inclinato, la rete mantiene il percorso stabilito per ultimo. Progettare la rete.



- 52) Realizzare una rete sequenziale sincronizzata con un ingresso x ed una uscita z che funziona nel seguente modo. All'istante generico t l'uscita assume valore 1 in corrispondenza del secondo di due 1 consecutivi che si presentano in ingresso dopo che si sono succeduti su di esso almeno due zeri. In tutti gli altri casi l'uscita vale 0. Usare latch SR.
- 53) Realizzare una rete sequenziale asincrona normale con due ingressi x_1 e x_2 ed una uscita z , avente il seguente comportamento. Quando su x_1 arriva un impulso positivo, di durata qualsiasi, mentre x_2 è zero e successivamente arriva su x_2 un impulso positivo, pure di durata qualsiasi, mentre x_1 è uno, l'uscita diventa uno e tale rimane fino al successivo impulso positivo su x_1 , che la riporta a zero. In tutte le altre situazioni l'uscita rimane zero.



- 54) Progettare una rete sequenziale sincronizzata che in una sequenza binaria di cinque bit riconosce se sono presenti almeno due zeri e due uno. Nel progetto fare uso di flip-flop JK.
- 55) Progettare una rete sequenziale con due ingressi x ed y ed una uscita z che si comporta nel seguente modo. Quando x (o y) diventa 1 e l'altro ingresso y (o x) rimane 0 e successivamente anch'esso diventa 1, oppure quando entrambi contemporaneamente diventano 1, l'uscita assume valore 1 all'istante in cui y va a 0 e x resta ad 1; l'uscita rimane 1 finché x va a 0, indipendentemente dal fatto che y cambi o rimanga immutato. In tutti gli altri casi l'uscita resta 0.
- 56) Un circuito sequenziale asincrono con due ingressi x_1 e x_2 ed una uscita z funziona nel seguente modo. Quando $x_1 = 0$ e x_2 cambia da 0 ad 1, l'uscita cambia da 1 a 0, oppure rimane 0. Quando $x_2 = 1$ e x_1 cambia da 0 ad 1, l'uscita cambia da 0 ad 1 oppure resta 1. Quando x_1 e x_2 cambiano insieme, l'uscita viene complementata. Per tutti gli altri stati di ingresso l'uscita diventa o rimane 1. Ricavare la tabella di flusso del circuito.
- 57) Un circuito sequenziale sincronizzato con due ingressi x_1 e x_2 ed una uscita z deve riconoscere se sull'ingresso x_1 si presenta la sequenza 010 mentre x_2 è 1, oppure la sequenza 101 mentre x_2 è 0. Le due sequenze possono essere interallacciate. Progettare il circuito utilizzando flip-flop JK.
- 58) Progettare una rete sequenziale sincronizzata che in una sequenza binaria di cinque bit riconosce se sono presenti almeno due zeri e due uno. Nel progetto si faccia uso di flip-flop JK.
- 59) Progettare una rete sequenziale sincronizzata che in una sequenza binaria di cinque bit riconosce se sono presenti almeno due zeri non consecutivi e due uno pure non consecutivi. Nel progetto si faccia uso di flip-flop JK.
- 60) Progettare un circuito asincrono normale con due ingressi x_1 ed x_2 ed una uscita z , la quale cambia da 0 ad 1 o da 1 a 0 ogni volta che si verifica un cambiamento da 0 ad 1 su uno qualunque dei due ingressi. Non sono possibili cambiamenti simultanei dei due ingressi.
- 61) E' ben noto il quesito del lupo, della capra e del cavolo: un tale deve trasferire da una sponda all'altra di un fiume un lupo, una capra ed un cavolo servendosi di una barca che, oltre a lui, può portare solo il lupo, o la capra oppure il cavolo. Quel tale deve fare in modo da non lasciare soli su alcuna delle due sponde il lupo e la capra, perchè il lupo mangerebbe la capra, nè la capra ed il cavolo perchè la capra mangerebbe il cavolo, ma solo al più il lupo ed il cavolo. Quale sequenza di passaggi deve effettuare quel tale per effettuare con successo il trasbordo degli animali e del cavolo?
 Si vuole realizzare una rete sequenziale sincrona che simula un gioco basato sul quesito sopra ricordato, secondo le seguenti specifiche.
 La rete possiede due ingressi x_1 e x_2 , mediante i quali si codificano gli eventi *barca senza carico*, *barca con lupo*, *barca con capra*, *barca con cavolo*. Un giocatore ad ogni passo sceglie l'evento che ritiene corretto nella sequenza risolutiva del quiz, definendo tramite deviatori il valore binario di x_1 e di x_2 e la rete evolve in accordo ad esso e alla

situazione interna corrente, producendo in uscita una delle risposte *nuovo passo*, *errore*, *successo* oppure nessuna risposta. Le uscite sono visualizzate su un display a segmenti luminosi il quale, nel caso di nessuna risposta, cioè prima dell'inizio di una sequenza di gioco, rimane oscuro.

Poichè l'intervento del giocatore è asincrono rispetto al clock, è necessario che ogni evento di ingresso selezionato sia memorizzato e acquisito dalla rete al primo impulso di clock utile, in corrispondenza del quale la rete provvede anche a cancellarlo, portando la configurazione di ingresso a quella di default associata all'evento *barca senza carico*.

Progettare la rete che simula il gioco, la logica di sincronizzazione degli ingressi e quella necessaria per la visualizzazione dell'uscita.

- 62) Progettare due reti sequenziali sincronizzate con le seguenti caratteristiche.

Rete 1

La rete possiede due ingressi x_1 e x_2 ed una uscita z ; quando x_1 cambia da 0 ad 1, mentre x_2 rimane stabile, l'uscita assume il valore di x_2 ; quando x_2 cambia da 1 a 0, mentre x_1 rimane stabile, l'uscita assume il valore complementare dell'ingresso x_1 ; in tutti gli altri casi l'uscita conserva il proprio valore. Realizzare la rete con flip-flop JK.

Rete 2

Anche questa rete possiede due ingressi v_1 e v_2 ed una uscita u . Quando un impulso si presenta sull'ingresso v_1 e successivamente un impulso su v_2 ed uno ancora su v_1 , l'uscita diventa 1 e rimane tale fino ad un nuovo impulso su v_2 , in corrispondenza del quale ritorna 0. In ogni altra situazione l'uscita conserva valore 0. Progettare la rete con flip-flop D.

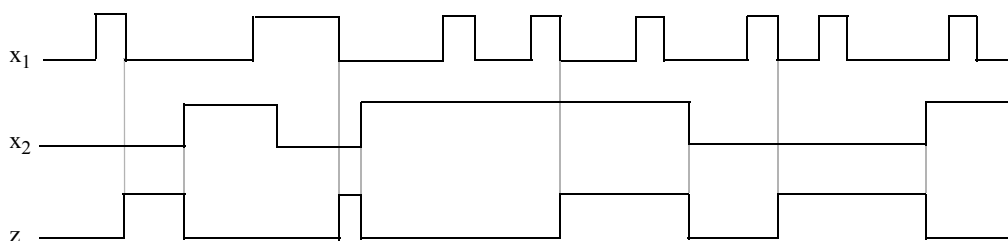
- 63) Un circuito sequenziale sincronizzato possiede due ingressi x_1 ed x_2 ed una uscita z ed opera nel seguente modo.

Quando $x_2 = 0$, z rimane 0 finchè si presenta su x_1 un impulso alla fine del quale z diventa 1, mantenendo poi tale valore finchè x_2 diventa 1; allora z ritorna a 0 e vi rimane.

Quando $x_2 = 1$ e all'ingresso x_1 si presenta la sequenza 101, z diventa 1 e mantiene tale valore finchè x_2 diventa 0; allora anche z ritorna a 0 e vi rimane.

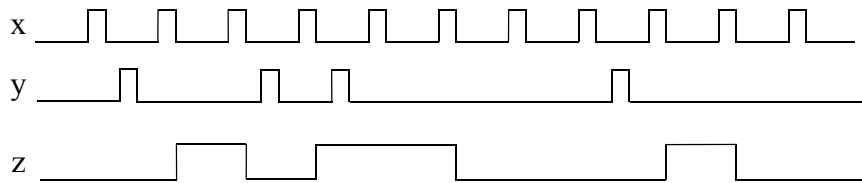
In ogni altro caso z rimane inalterata.

Progettare il circuito come macchina di Moore, utilizzando per la parte combinatoria una memoria di sola lettura.



- 64) Un circuito sequenziale possiede due ingressi ad impulsi x ed y ed una uscita a livelli z . Gli impulsi sull'ingresso x sono periodici, mentre quelli sull'ingresso y sono casuali,

ma quando si verificano sono accettati solo se si presentano tra un impulso di x e l'altro. L'uscita z è 1 durante l'intervallo tra due impulsi consecutivi su x , purchè nell'intervallo precedente si sia avuto un impulso su y , altrimenti rimane 0 (v. figura). Progettare il circuito facendo uso di flip-flop JK.



- 65) Un circuito sequenziale sincronizzato possiede due ingressi x_1 ed x_2 ed un'uscita z . Quando l'ingresso x_1 assume valore 0 e lo mantiene per tre periodi di clock consecutivi e durante tale intervallo l'ingresso assume valore 1 e lo mantiene per due periodi di clock consecutivi, l'uscita passa da 0 ad 1 e rimane tale finchè una transizione di x_1 da 1 a 0 riporta a 0 anche l'uscita. In tutti gli altri casi l'uscita rimane a zero.

Progettare il circuito facendo uso di flip-flop JK.