

Si noti quindi che un contatore Johnson realizzato con n flip-flop possiede un numero di stati pari a $2n$.

Se all'accensione o in seguito ad un errore causato da un rumore elettrico, il circuito dovesse trovarsi in uno stato compreso nel ciclo rappresentato in nero, sarebbe necessario compiere un RESET per rientrare nel ciclo principale.

ESEMPIO 7.25

Supponendo lo stato iniziale del contatore ad anello in fig. 7.74a corrispondente a $Q_1 \div Q_5 = 10100$, determinare la sequenza degli stati nei successivi 7 periodi di clock.

Soluzione

La sequenza degli stati è rappresentata nella tabella di fig. 7.74b.

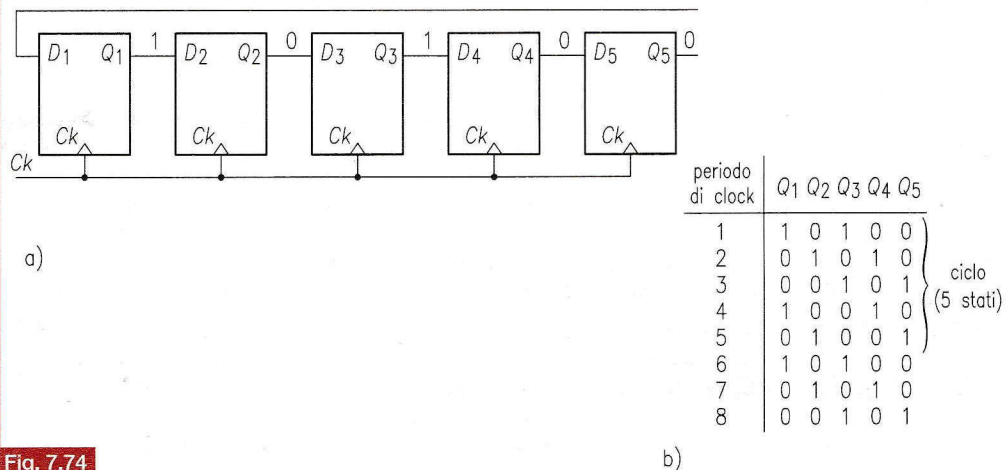


Fig. 7.74

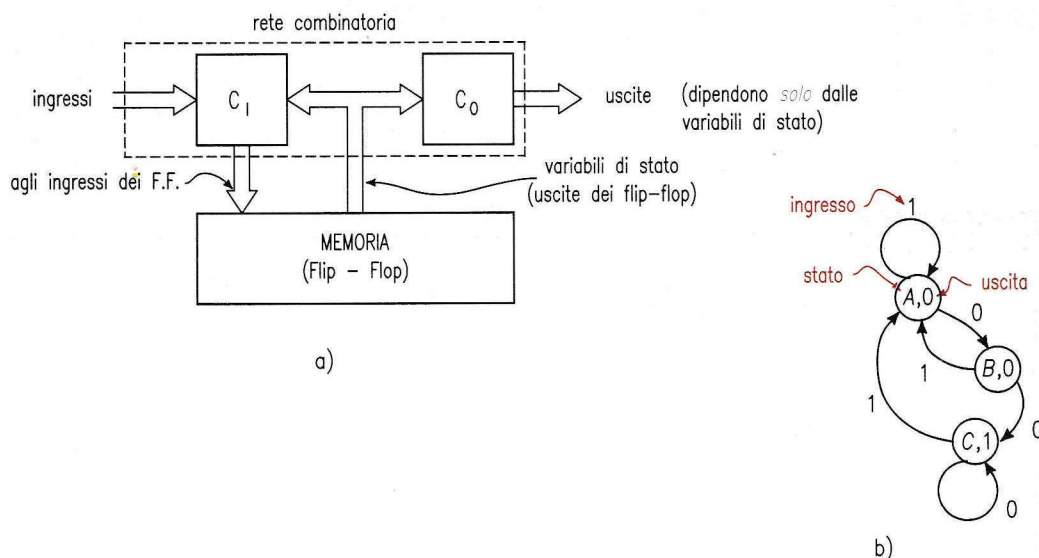
7.8 Reti sequenziali sincrone con ingressi

I contatori analizzati e progettati nei precedenti paragrafi sono reti sequenziali particolarmente semplici, in quanto non presentano variabili d'ingresso, ma si limitano a percorrere un ciclo prefissato secondo la temporizzazione scandita dal segnale di clock. Di conseguenza, lo stato raggiunto dal contatore in un dato istante, dipende solo dallo stato in cui il circuito si trovava nell'istante precedente.

Spesso è invece necessario poter influenzare l'evoluzione degli stati inviando segnali in ingresso alla rete, come ad esempio in un contatore dotato di un ingresso *UP/DOWN*, che determina il verso del conteggio, oppure in una rete di controllo per un semaforo che, durante il ciclo normale, tenga anche conto della pressione di un pulsante per forzare il verde in una direzione.

Le **reti sequenziali sincrone con ingressi** possono essere di due tipi:

- Def.** 1) **rete di Moore:** rete sequenziale sincrona in cui le variabili d'uscita, in un dato istante, dipendono solo da quelle di stato e non sono direttamente influenzate dai valori presenti in ingresso nello stesso istante.



Rete sequenziale sincrona di Moore: a) struttura; b) esempio di diagramma degli stati.

Fig. 7.75

Nello schema a blocchi di una rete di Moore (fig. 7.75a) si nota la rete combinatoria C_1 , che determina lo stato futuro in base ai valori degli ingressi e dello stato presente, e la rete C_0 , che determina i valori delle uscite come combinazione delle variabili di stato.

Le uscite di una rete di Moore commutano sempre in sincronismo con il clock, perché sono ottenute tramite una combinazione delle uscite dei flip-flop, tutti sincronizzati dal medesimo segnale di clock.

Nella fig. 7.75b è rappresentato il *diagramma degli stati* di una rete di Moore con un ingresso ed un'uscita.

In ogni circoletto è contenuta una lettera alfabetica, che identifica lo stato, ed il valore dell'uscita associato allo stato. In realtà ogni stato è individuato da una combinazione di valori delle uscite dei flip-flop; questi valori rappresentano variabili interne al circuito e, poiché generalmente non è interessante evidenziarli nel diagramma degli stati come accadeva per i contatori, sono associati a lettere alfabetiche.

Da ogni stato escono due frecce, corrispondenti ai due valori possibili della variabile d'ingresso, che vengono indicati a fianco delle frecce.

La freccia indica quindi la transizione dello stato, e di conseguenza dell'uscita associata, quando, in corrispondenza del fronte attivo di clock, l'ingresso assume il valore specificato a fianco.

Il funzionamento della rete relativa al diagramma degli stati di fig. 7.75b è il seguente:

l'uscita si porta al valore 1 quando in ingresso si presentano almeno due 0 consecutivi; si noti infatti che dallo stato A, associato al valore d'uscita 0, è necessario applicare il valore 0 in ingresso per due periodi di clock successivi per giungere allo stato C, associato al valore d'uscita 1. Successivi valori 0 in ingresso mantengono la rete nello stato C, mentre un valore 1 in ingresso riporta la rete nello stato A e l'uscita al valore 0.

Un flip-flop costituisce il più semplice esempio di rete di Moore; nella fig. 7.76 si riportano i diagrammi degli stati dei flip-flop D, T e JK.

Def. 2) **rete di Mealy:** rete sequenziale sincrona in cui le variabili d'uscita, in un dato istante, dipendono sia da quelle di stato che da quelle d'ingresso.

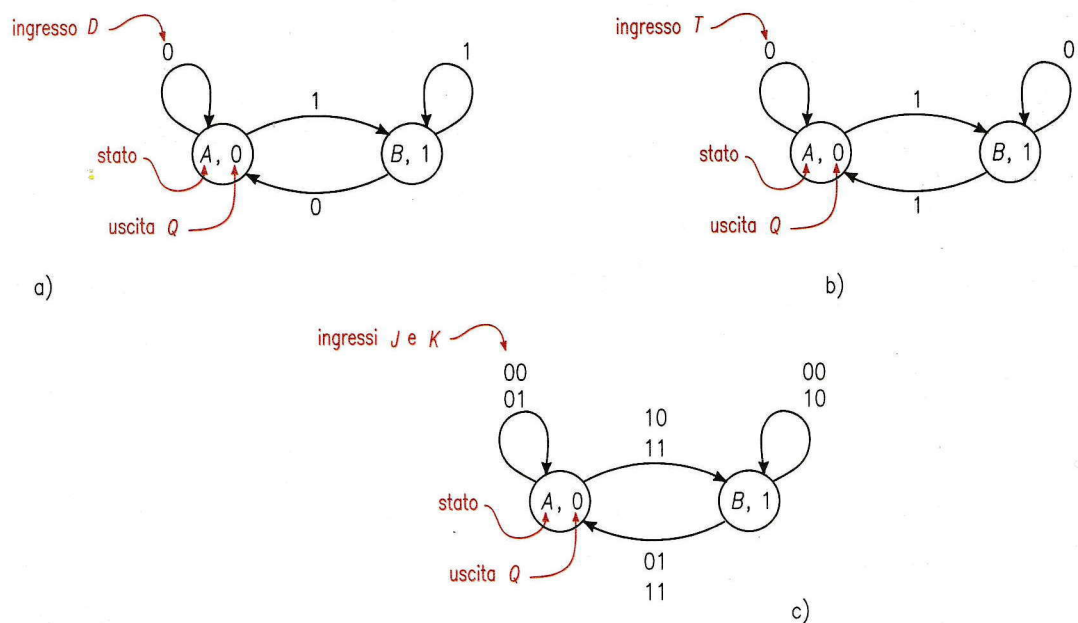


Fig. 7.76 Diagrammi degli stati dei flip-flop a) D; b) T; c) JK.

Nello schema a blocchi (fig. 7.77a) si nota infatti che la rete combinatoria C_0 determina i valori delle uscite in base alle variabili di stato ed a quelle d'ingresso. Di conseguenza le commutazioni degli ingressi influenzano immediatamente i valori delle uscite, che quindi potrebbero commutare anche in istanti diversi da quelli di clock.

Per tenere conto di questo funzionamento, il diagramma degli stati di una rete di Mealy riporta i valori delle uscite in prossimità delle frecce a fianco di quelli degli ingressi, come indicato nella fig. 7.77b.

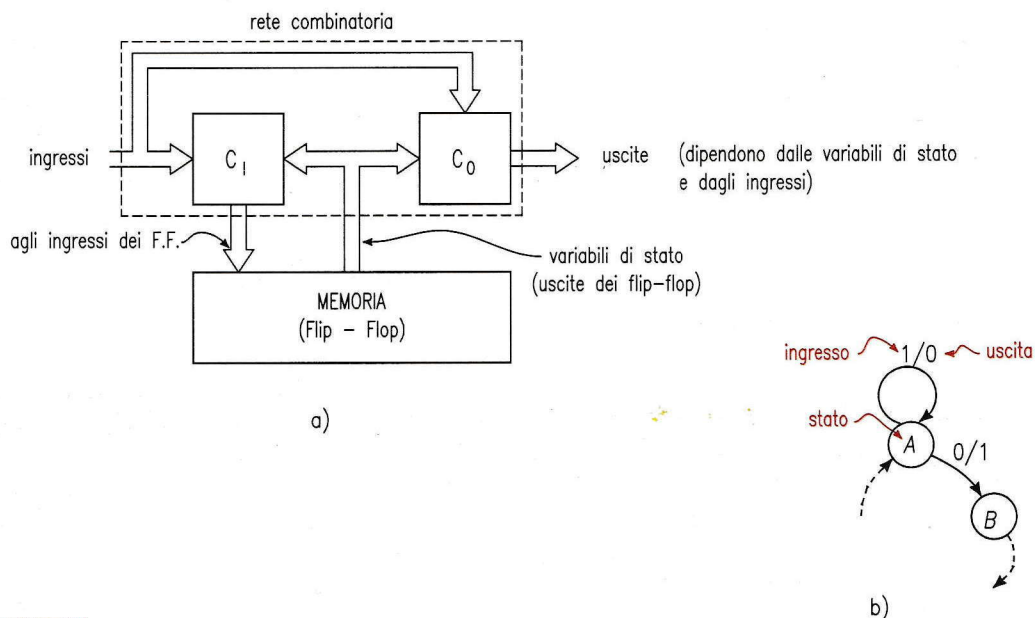


Fig. 7.77 Rete sequenziale sincrona di Mealy: a) struttura; b) esempio di diagramma degli stati.

Non si approfondisce oltre la descrizione delle reti di Mealy; gli esempi e gli esercizi di progetto proposti nel testo riguardano esclusivamente le reti di Moore.

7.8.1 Progetto di una rete sequenziale sincrona di Moore



Si descrive ora il procedimento per **progettare una rete di Moore** che soddisfi una serie di specifiche espresse a parole (si vedano gli esempi 7.26, 7.27, 7.28):

- 1) dalla descrizione a parole si individuano le *variabili d'ingresso e d'uscita* della rete, associando ai loro stati i valori logici 0 e 1.
- 2) Seguendo le specifiche si traccia il *diagramma degli stati*; si verifichi che da ogni stato, inizialmente identificato con una lettera alfabetica, esca un numero di frecce pari al numero di tutte le possibili combinazioni delle variabili d'ingresso.
- 3) Si determina il *numero di flip-flop necessari* per rappresentare tutti gli stati del diagramma, tenendo conto che con N bistabili si possono memorizzare 2^N stati diversi; il *tipo di flip-flop*, se non indicato nelle specifiche, può essere scelto indifferentemente tra D, JK o T.
- 4) Si *compila una tabella* con 3 colonne, in cui ad ogni *lettera che identifica lo stato* sono associate una combinazione di *valori delle uscite Q dei flip-flop* (variabili di stato) ed il relativo *valore delle uscite della rete*. L'associazione tra stato (lettera) e variabili di stato (combinazione binaria) è arbitraria, ma bisogna considerare che più la codifica degli stati è simile ai corrispondenti valori delle uscite della rete, più risulterà semplice la rete combinatoria C_O che genera le variabili d'uscita; in corrispondenza delle combinazioni non associate ad alcuno stato, il valore delle uscite è indifferente (*don't care*).
- 5) Si *sintetizza la rete combinatoria C_O* in forma minima, trasformando la tabella del punto 4 in tante mappe di Karnaugh quante sono le variabili d'uscita e ricavando le espressioni che legano le variabili d'uscita con quelle di stato.
- 6) Si *compila la tabella di flusso* così organizzata:
 - a) 1^a colonna: elenca tutte le possibili combinazioni delle *variabili d'ingresso e di stato presente*, che rappresentano gli ingressi della rete combinatoria C_i ;
 - b) 2^a colonna: indica lo *stato futuro* relativo ad ognuna delle combinazioni della 1^a colonna, dedotto dal diagramma degli stati;
 - c) 3^a colonna: riporta i valori che devono assumere gli *ingressi di ogni flip-flop* per provocare la transizione, specificata dalle colonne 1 e 2, dell'uscita corrispondente.
- 7) Si *sintetizza la rete combinatoria d'ingresso C_i* in forma minima, trasformando in mappe di Karnaugh le colonne 1 e 3 della tabella di flusso compilata nel punto 6 e ricavando le espressioni che legano gli ingressi di ogni flip-flop con le variabili di stato presente e le variabili d'ingresso della rete.
- 8) Si disegna la rete complessiva.

ESEMPIO 7.26

Si vuole realizzare una rete in grado di rivelare la presenza di almeno due livelli ALTI consecutivi su una linea, in corrispondenza dei fronti di salita di un segnale di clock, fornito su un'altra linea (CK).

La rete deve comunicare la rivelazione della sequenza significativa ponendo un livello ALTO sull'uscita, in sincronismo con il segnale di clock.

Soluzione

Poiché l'uscita deve commutare in sincronismo con il clock e non deve essere influenzata da variazioni dell'ingresso tra un fronte di clock e il successivo, è necessario realizzare una rete di Moore, seguendo il procedimento illustrato precedentemente.

- 1) La rete possiede un ingresso I ed un'uscita U , oltre all'ingresso di clock (CK) che dev'essere collegato a tutti i flip-flop. Si sceglie l'associazione in logica positiva tra livelli elettrici e valori logici: BASSO = 0, ALTO = 1.
- 2) Si traccia il *diagramma degli stati* (fig. 7.78a); i tre stati individuati hanno il seguente significato:
 - A: l'ultimo dato ricevuto è 0, si è in attesa di un 1; l'uscita vale $U = 0$;
 - B: si è ricevuto il primo 1, si è in attesa del secondo 1; l'uscita vale $U = 0$;
 - C: si sono ricevuti almeno due 1 consecutivi e lo si segnala ponendo $U = 1$.
- 3) Poiché il diagramma è composto da tre stati sono necessari due flip-flop ($2^2 = 4 > 3$), che si scelgono di tipo JK.
- 4) Si compila la tabella di fig. 7.78b, dove ad ogni stato viene associata una coppia di valori delle variabili di stato (uscite dei flip-flop: Q); si noti che alla coppia 11 non è associato nessuno stato del diagramma e quindi il corrispondente valore dell'uscita U è indifferente (*don't care*).
- 5) Dalla tabella di fig. 7.78b si ricava la mappa di Karnaugh di fig. 7.78c e si sintetizza la rete combinatoria C_o , che risulta data da:

$$U = Q_1$$

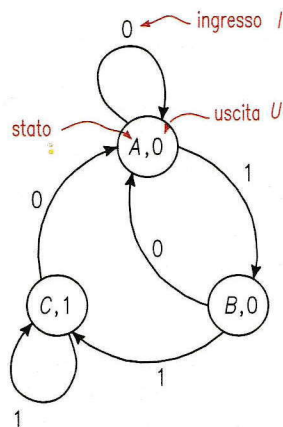
di conseguenza non è necessaria nessuna porta logica, ma l'uscita U è connessa direttamente alla variabile di stato Q_1 .

- 6) Si compila la tabella di flusso in fig. 7.78d; si noti che alla combinazione 11 delle variabili di stato presente, che non corrisponde a nessuno stato nel diagramma ma che potrebbe però comparire all'accensione del circuito o in caso d'errore, si fa corrispondere lo stato futuro A ($Q_1 = Q_0 = 0$).
- 7) Si sintetizza la rete combinatoria C_i , compilando una mappa per ogni ingresso dei flip-flop con i valori delle colonne 1 e 3 della tabella di flusso (fig. 7.78e); dalle mappe si ricavano le seguenti funzioni logiche:

$$J_1 = I \cdot Q_1; \quad K_1 = \bar{I} + Q_0; \quad J_0 = I \cdot \bar{Q}_1; \quad K_0 = 1$$

- 8) Si disegna quindi la rete complessiva (fig. 7.78f).

(Esercitazioni di laboratorio n° 28 ob. 1 e n° 33).



stato	variabili di stato		uscita
	Q_1	Q_0	
A	0	0	0
B	0	1	0
C	1	0	1
	1	1	X

b)

Q_1	Q_0	
	0	1
0	0	0
1	1	X

$$U = Q_1$$

c)

ingresso	stato presente		stato futuro	ingressi dei Flip-Flop			
I	Q_1	Q_0	$Q_1 Q_0$	J_1	K_1	J_0	K_0
0	0	0	0 0	0	X	0	X
1	0	0	0 1	0	X	1	X
0	0	1	0 0	0	X	X	1
1	0	1	1 0	1	X	X	1
0	1	0	0 0	X	1	0	X
1	1	0	1 0	X	0	0	X
0	1	1	0 0	X	1	X	1
1	1	1	0 0	X	1	X	1

d)

I	$Q_1 Q_0$			
	00	01	11	10
0	0	0	X	X
1	0	1	X	X

$$J_1 = I Q_0$$

I	$Q_1 Q_0$			
	00	01	11	10
0	X	X	1	1
1	X	X	1	0

$$K_1 = \bar{I} + Q_0$$

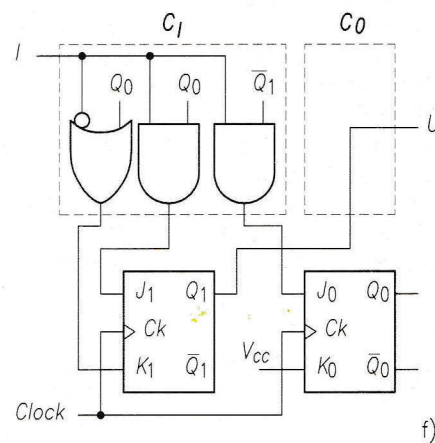
I	$Q_1 Q_0$			
	00	01	11	10
0	0	X	X	0
1	1	X	X	0

$$J_0 = I \bar{Q}_1$$

I	$Q_1 Q_0$			
	00	01	11	10
0	X	1	1	X
1	X	1	1	X

$$K_0 = 1$$

e)



f)

Fig. 7.78

ESEMPIO 7.27

Si realizzi un circuito per controllare il funzionamento di un semaforo posto all'incrocio di due strade, come indicato nella fig. 7.79a.

Il ciclo normale del semaforo, temporizzato da un segnale di clock con periodo 10 s, è il seguente:

- verde nella direzione 1 (V_1) e rosso nella direzione 2 (R_2) per 30 s;
- verde più arancione nella direzione 1 (V_1, A_1) e rosso nella direzione 2 (R_2) per 10 s;
- rosso nella direzione 1 (R_1) e verde nella direzione 2 (V_2) per 30 s;
- rosso nella direzione 1 (R_1) e verde più arancione nella direzione 2 (V_2, A_2) per 10 s;

Si supponga che l'accensione delle lampade del semaforo sia provocata dal livello ALTO dei segnali di uscita.

Inoltre il semaforo è dotato di un sistema che segnala, tramite un livello ALTO su un'apposita linea (I), l'avvicinarsi di un mezzo di soccorso nella direzione 1; in questa eventualità il circuito deve consentire il passaggio nella direzione 1 nel tempo più rapido possibile.

Soluzione

- La rete presenta in ingresso il segnale I che rivela l'avvicinarsi di un mezzo di soccorso nella direzione 1. Le uscite dovrebbero essere sei, una per ogni colore delle lampade nelle due direzioni, ma si osserva che il verde in una direzione appare sempre in concomitanza del rosso nell'altra direzione; inoltre il rosso e il verde nella medesima direzione sono sempre complementari. Risultando quindi:

$$V_1 = R_2 = \bar{V}_2 = \bar{R}_2$$

si eseguirà il progetto considerando solo le tre variabili d'uscita V_1, A_1 e A_2 .

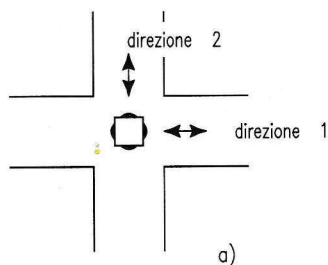
Si sceglie l'associazione in logica positiva tra livelli elettrici e valori logici: BASSO=0, ALTO=1.

- Si traccia il *diagramma degli stati* (fig. 7.79b) dove si nota che il ciclo normale ($I=0$) mantiene il verde nella direzione 1 ($V_1=1$) per tre periodi di clock, negli stati A, B, C . Nello stato D , al verde si aggiunge l'arancione ($A_1=1$) segnalando, per un periodo di clock, che sta per scattare il rosso. Negli stati E, F, G si ha il verde nella direzione 2 (si ricordi che $V_2 = \bar{V}_1$), a cui si aggiunge l'arancione ($A_2=1$) nello stato H . Nel caso di arrivo di un mezzo di soccorso ($I=1$) si passa allo stato A , se nella direzione 1 c'è già il verde, oppure allo stato H , se nella direzione 1 c'è il rosso.
- Per rappresentare gli otto stati sono necessari tre *flip-flop*; si scelgono flip-flop di tipo D.
- Si compila la *tabella* di fig. 7.79c che codifica ogni stato con una combinazione delle variabili di stato.
- Dalla tabella di fig. 7.79c si ricavano le *mappe di Karnaugh* di fig. 7.79d e si sintetizza la *rete combinatoria* C_0 , che fornisce le variabili d'uscita (V_1, A_1, A_2) in funzione di quelle di stato:

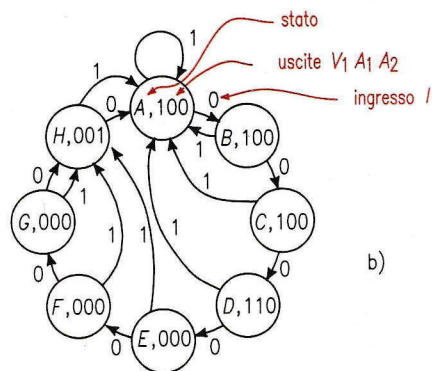
$$V_1 = \bar{Q}_2; \quad A_1 = Q_2 Q_1 \bar{Q}_0; \quad A_2 = Q_2 Q_1 Q_0$$

- Si compila la *tabella di flusso* di fig. 7.79e; l'uguaglianza tra le colonne 2 e 3 consegue dalla tabella d'eccitazione del flip-flop D.
- Si sintetizza la *rete combinatoria* C_1 , compilando una mappa per ogni ingresso dei flip-flop (fig. 7.79f), con i valori rilevati nelle colonne 1 e 3 della tabella di flusso.
- Si lascia allo studente il compito di disegnare la rete complessiva.

(Esercitazione di laboratorio n° 28 ob. 2 e n° 33).



stato	variabili di stato			uscite		
	Q_2	Q_1	Q_0	V_1	A_1	A_2
A	0	0	0	1	0	0
B	0	0	1	1	0	0
C	0	1	0	1	0	0
D	0	1	1	1	1	0
E	1	0	0	0	0	0
F	1	0	1	0	0	0
G	1	1	0	0	0	0
H	1	1	1	0	0	1



c)

Q_2	$Q_1 Q_0$	00	01	11	10
0		1	1	1	1
1		0	0	0	0

$$V_1 = \bar{Q}_2$$

Q_2	$Q_1 Q_0$	00	01	11	10
0		0	0	0	0
1		0	0	1	0

$$A_2 = Q_0 Q_1 Q_2$$

Q_2	$Q_1 Q_0$	00	01	11	10
0		0	0	1	0
1		0	0	0	0

$$A_1 = Q_0 Q_1 \bar{Q}_2$$

Q_2	$Q_1 Q_0$	00	01	11	10
0		0	0	1	0
1		1	1	0	1
1		1	1	0	1
1		0	0	0	0

$$D_2 = \bar{Q}_2 Q_1 Q_0 + Q_2 \bar{Q}_1 + Q_2 \bar{Q}_0$$

Q_2	$Q_1 Q_0$	00	01	11	10
0		0	1	0	1
1		0	1	0	1
1		1	1	0	1
1		0	0	0	0

$$D_1 = \bar{Q}_1 Q_0 + \bar{Q}_1 \bar{Q}_0 + Q_2 \bar{Q}_1 + Q_2 Q_1 \bar{Q}_0$$

Q_2	$Q_1 Q_0$	00	01	11	10
0		1	0	0	1
1		1	0	0	1
1		1	1	0	1
1		0	0	0	0

$$D_0 = \bar{Q}_0 + Q_2 \bar{Q}_0 + Q_2 \bar{Q}_1$$

ingresso	stato presente			stato futuro			ingressi Flip-Flop		
I	Q_2	Q_1	Q_0	Q_2	Q_1	Q_0	D_2	D_1	D_0
0	0	0	0	0	0	1	0	0	1
1	0	0	0	0	0	0	0	0	0
0	0	0	1	0	1	0	0	1	0
1	0	0	1	0	0	0	0	0	0
0	0	1	0	0	1	1	0	1	1
1	0	1	0	0	0	0	0	0	0
0	0	1	1	1	0	0	1	0	0
1	0	1	1	0	0	0	0	0	0
0	1	0	0	1	0	1	1	0	1
1	1	0	0	1	1	1	1	1	1
0	1	0	1	1	1	0	1	1	0
1	1	0	1	1	1	1	1	1	1
0	1	1	0	1	1	1	1	1	1
1	1	1	0	0	0	0	0	0	0
0	1	1	1	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0

e)

Fig. 7.79

ESEMPIO 7.28

Si vuole realizzare una rete che abiliti l'apertura dello sportello di un armadio blindato, quando quattro pulsanti (Y , Z , C , R) vengono attivati nel modo seguente:

dopo la pressione di R (*Reset*) è necessario premere una volta Y , una volta Z e poi entrambi contemporaneamente; ogni pressione di Y e di Z deve essere convalidata premendo e rilasciando C (*Clock*).

Soluzione

- 1) Si considerano solo Y e Z come variabili d'ingresso perché C viene utilizzata come clock della rete, in quanto i suoi fronti fungono da sincronizzazione per i flip-flop, ed R può essere collegata agli ingressi *CLEAR* dei flip-flop per portare la rete nello stato iniziale. Per tutti i pulsanti si sceglie la seguente associazione:

pulsante premuto = livello elettrico ALTO = stato logico 1.

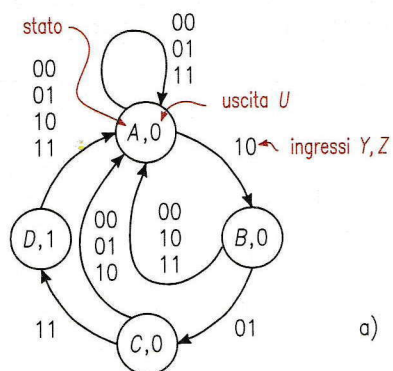
Si decide che l'uscita U della rete segnali l'avvenuta sequenza corretta, portandosi a livello ALTO (stato logico 1).

- 2) Si traccia il *diagramma degli stati* (fig. 7.80a); si tenga conto che in qualunque istante, la pressione del pulsante R , collegato agli ingressi *CLEAR* dei flip-flop, porta immediatamente allo stato iniziale A . Dopo la pressione dei pulsanti secondo la sequenza corretta, si giunge allo stato D , a cui corrisponde $U=1$, che abilita l'apertura della serratura.
- 3) Per rappresentare i quattro stati del diagramma sono necessari due flip-flop, che si scelgono di tipo T.
- 4) 5) Si compila la *tabella* di fig. 7.80b e si ricava immediatamente la *rete combinatoria* C_o , rappresentata dalla relazione:

$$U = Q_1 Q_0.$$

- 6) Si compila la *tabella di flusso* di fig. 7.80c, basandosi sul diagramma degli stati (fig. 7.80a) e sulla tabella di eccitazione del flip-flop T.
- 7) Dalla *tabella di flusso* si compilano le mappe di fig. 7.80d e si ricavano le funzioni di commutazione che descrivono la rete combinatoria C_p .
- 8) Si disegna la rete complessiva fig. 7.80e.

Nella pratica i rimbalzi dei contatti del pulsante C provocano, ad ogni pressione, la nascita di più fronti di salita che vengono interpretati come segnali di clock dai flip-flop; ciò produce la lettura ripetuta dei dati posti in ingresso e quindi un funzionamento diverso da quello previsto. Per ovviare a questo inconveniente è necessario aggiungere al pulsante C un circuito antibounce, scelto tra quelli illustrati nel par. 7.2. (Esercitazione di laboratorio n° 28, ob. 3 e n° 33).



stato	variabili di stato	uscita
	$Q_1 Q_0$	U
A	0 0	0
B	0 1	0
C	1 0	0
D	1 1	1

$$U = Q_1 Q_0$$

b)

ingressi+ stato presente	stato futuro	ingressi Flip-Flop
$Y Z Q_1 Q_0$	$Q_1 Q_0$	$T_1 T_0$
0 0 0 0	0 0	0 0
0 1 0 0	0 0	0 0
1 0 0 0	0 1	0 1
1 1 0 0	0 0	0 0
0 0 0 1	0 0	0 1
0 1 0 1	1 0	1 1
1 0 0 1	0 0	0 1
1 1 0 1	0 0	0 1
0 0 1 0	0 0	1 0
0 1 1 0	0 0	1 0
1 0 1 0	0 0	1 0
1 1 1 0	1 1	0 1
0 0 1 1	0 0	1 1
0 1 1 1	0 0	1 1
1 0 1 1	0 0	1 1
1 1 1 1	0 0	1 1

c)

$Q_1 Q_0$	00	01	11	10	
$Y Z$	00	0	0	1	1
01	0	1	1	1	
11	0	0	1	0	
10	0	0	1	1	

$T_1 = Q_1 Q_0 + \bar{Y} Q_1 + \bar{Z} Q_1 + \bar{Y} Z Q_0$

$$T_1 = Q_1 Q_0 + \bar{Y} Q_1 + \bar{Z} Q_1 + \bar{Y} Z Q_0$$

		$Q_1 Q_0$			
		00	01	11	10
$Y Z$	00	0	1	1	0
	01	0	1	1	0
	11	0	1	1	1
	10	1	1	1	0

$$T_0 = Q_0 + Y Z Q_1 + Y \bar{Z} Q_1$$

d)

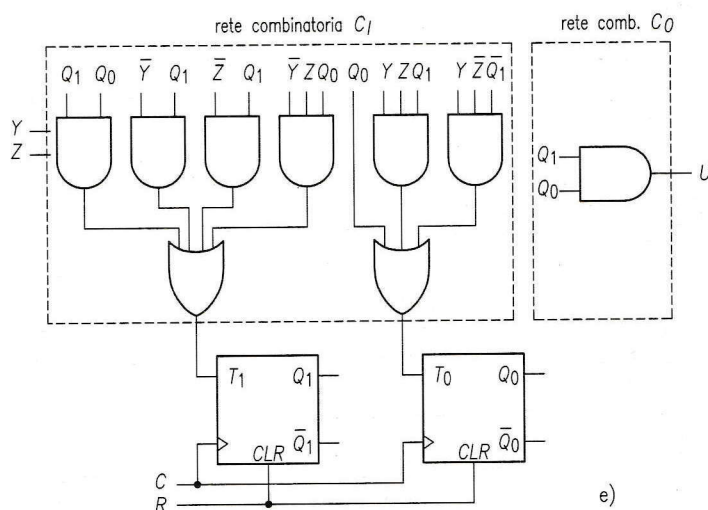


Fig. 7.80

7.9 Logica cablata e logica programmabile

Come si è visto negli esempi precedenti, la sintesi sincrona di Moore permette di risolvere tutti quei problemi in cui l'andamento delle uscite, sincrone con il clock, è dipendente dalla sequenza dei valori che si presentano sugli ingressi.

Per problemi semplici come quelli appena risolti, si ottengono circuiti relativamente semplici ed economici, ma se il funzionamento richiesto è più complesso la realizzazione della rete con porte logiche e flip-flop potrebbe non risultare quella ottima.

Def. Questo tipo di progettazione viene detta in **logica cablata** (*wired logic*), a sottolineare il fatto che la funzione logica nasce dall'opportuna connessione di elementi come porte logiche, flip-flop, registri, contatori, ecc.
Una diversa filosofia consiste nel progettare il circuito in **logica programmabile** (*programmable logic*), basata sull'utilizzo di un sistema a *microprocessore* (*hardware*) che produce valori in uscita in funzione di quelli di ingresso, eseguendo una serie di istruzioni detta *programma* (*software*), codificate all'interno di memorie elettroniche collegate al microprocessore.

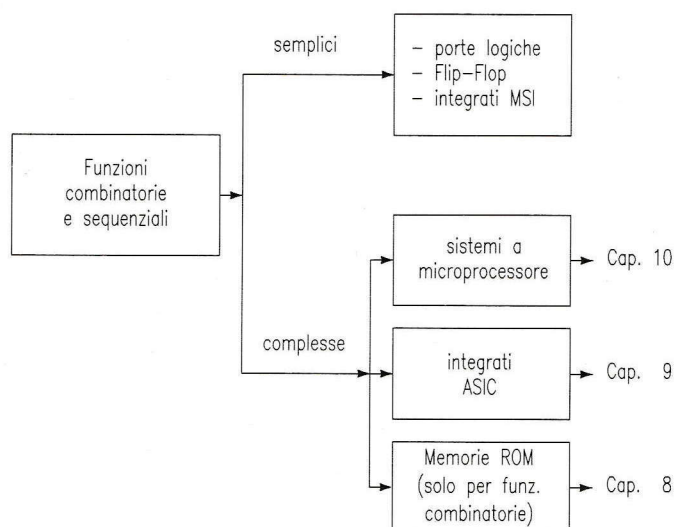


Fig. 7.81 Possibili scelte per l'implementazione di funzioni combinatorie o sequenziali.

L'utilizzo di un sistema a microprocessore, che presenta in genere un'architettura complessa e costosa, è giustificato solo sopra un certo livello di complessità delle funzioni richieste al circuito.

La logica programmabile è più flessibile di quella cablata, poiché è sufficiente modificare le istruzioni del programma in memoria per ottenere un diverso funzionamento del circuito; al contrario la logica cablata assicura, in genere, una maggiore velocità di esecuzione. La logica programmabile è descritta nel cap. 10.

Una terza possibilità, che può ottimizzare progetti complessi, è data dall'utilizzo di **circuiti integrati per applicazioni specifiche (ASIC)**, che sono trattati nel cap. 9.

Come già accennato a proposito dei circuiti combinatori, è possibile realizzare funzioni logiche utilizzando le **memorie ROM**, descritte nel cap. 8.

Lo schema in fig. 7.81 evidenzia le tre possibilità per la realizzazione di reti combinatorie e sequenziali.