

Le macchine a stati finiti

Come si è visto in apertura di appendice, le reti logiche digitali possono essere classificate come combinatorie o sequenziali: i sistemi sequenziali contengono delle informazioni di stato memorizzate negli elementi di memoria interni al sistema. Il comportamento di un sistema sequenziale dipende sia dall'insieme di ingressi forniti al sistema che dal contenuto della memoria interna, che rappresenta lo stato del sistema. I sistemi sequenziali quindi non sono descrivibili mediante le tabelle di verità: occorre invece ricorrere a descrizioni basate su *macchine a stati finiti* (spesso dette solo *macchine a stati*). Una macchina a stati finiti è caratterizzata da un insieme di stati e da due funzioni dette *funzione di stato futuro* e *funzione di uscita*; l'insieme di stati corrisponde a tutti i possibili valori degli elementi di memoria interna, per cui se vi sono n bit di memoria vi saranno 2^n stati. La funzione di stato futuro è una funzione combinatoria che, dato il valore dell'ingresso e dello stato presente (o stato corrente), calcola il valore del prossimo stato assunto dal sistema; la funzione di uscita produce invece le uscite a partire dallo stato presente e dagli ingressi. La figura B.27 riporta in modo schematico i diversi elementi.

Le macchine a stati finiti trattate in questa appendice e nel Capitolo 5 sono di tipo *sincrono*, cioè lo stato cambia in corrispondenza dei cicli di clock, e si calcola un nuovo stato ad ogni nuovo ciclo di clock; in pratica gli elementi di stato sono aggiornati solo in corrispondenza dei fronti del clock. Tale metodologia è usata in tutto questo paragrafo e nei Capitoli 5 e 6, per cui non si indicherà esplicitamente il segnale di clock. Nel Capitoli 5 e 6 le macchine a stati finiti sono utilizzate per il controllo delle azioni del processore e dell'unità di elaborazione.

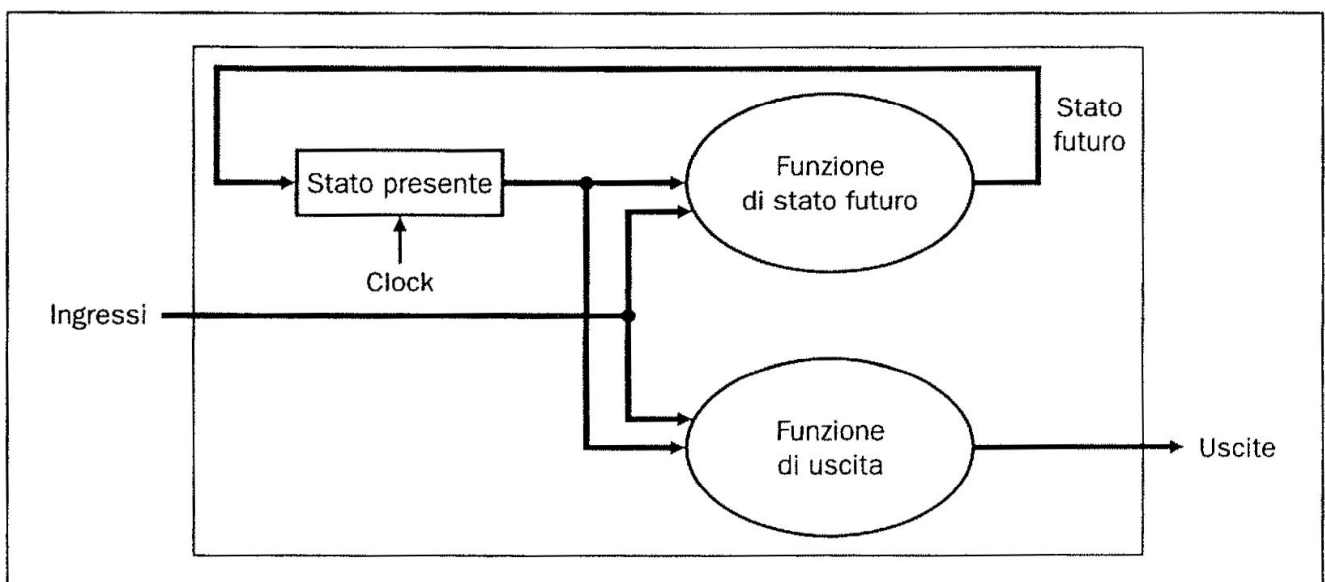


Figura B.27 Una macchina a stati consiste di elementi di memoria interni che memorizzano lo stato e di due funzioni combinatorie: la funzione di stato futuro e la funzione di uscita. Spesso si usa una versione più semplice della funzione di uscita, vincolandola a ricevere in ingresso solo il valore dello stato corrente; ciò non modifica le capacità del circuito sequenziale, ma ha influenza sulla sua struttura interna.

Per illustrare il funzionamento di una macchina a stati finiti e analizzarne le tecniche di progetto si utilizzerà un semplice esempio: il controllo di un semaforo (i Capitoli 5 e 6 contengono esempi più dettagliati dell'uso delle macchine a stati finiti nel controllo del processore). Quando si usa una macchina a stati finiti come controllore, la funzione di uscita viene spesso limitata in modo da dipendere solo dallo stato presente: macchine a stati finiti di tale tipo sono dette *macchine di Moore*, e saranno le sole che si utilizzeranno in questo testo. Se la funzione di uscita dipendesse sia dallo stato presente che dall'ingresso corrente, si direbbe che la macchina è una *macchina di Mealy*; i due tipi di macchine hanno capacità equivalenti, e si possono convertire l'una nell'altra seguendo un procedimento del tutto meccanico. Il vantaggio fondamentale delle macchine di Moore è di essere più veloci, mentre le macchine di Mealy possono rivelarsi più piccole, poiché possono richiedere un numero di stati minore rispetto alle corrispondenti macchine di Moore. La differenza è approfondita nel Capitolo 5.

L'esempio che verrà sviluppato è relativo al controllo di un semaforo posto all'incrocio di una strada in direzione nord-sud con una strada in direzione est-ovest; per semplicità si considereranno solo le luci verdi e rosse, mentre l'aggiunta del giallo è lasciata al lettore come semplice esercizio. Si vuole che le luci non cambino più velocemente di una volta ogni 30 secondi in entrambi i versi, per cui si userà un clock di 0,033 Hz in modo che la macchina cambi stato al più una volta ogni 30 secondi. Vi sono due segnali di uscita:

- *luceNS*: Quando si afferma questo segnale, viene accesa la luce verde sulla strada nord-sud; quando il segnale non è affermato, viene accesa la luce rossa sulla strada nord-sud.
- *luceEO*: Quando si afferma questo segnale, viene accesa la luce verde sulla strada est-ovest; quando il segnale non è affermato, viene accesa la luce rossa sulla strada est-ovest.

Vi sono inoltre due ingressi:

- *autoNS*: Indica che vi è un'auto in corrispondenza del rilevatore posto sotto l'asfalto di fronte al semaforo sulla strada nord-sud (sia in direzione nord che in direzione sud).
- *autoEO*: Indica che vi è un'auto in corrispondenza del rilevatore posto sotto l'asfalto di fronte al semaforo sulla strada est-ovest (sia in direzione est che in direzione ovest).

Il semaforo deve modificare la direzione nella quale viene presentato il verde solo se vi è un'auto in attesa su tale strada; altrimenti il semaforo deve continuare a mostrare il verde sulla stessa strada da cui proveniva l'ultima auto che è passata nell'incrocio.

Per implementare questo semplice semaforo sono necessari due stati:

- *verdeNS*: Il semaforo è verde nella direzione nord-sud.
- *verdeEO*: Il semaforo è verde nella direzione est-ovest.

È altresì necessario definire la funzione di stato futuro, che può essere specificata per mezzo di una tabella come la seguente.

Stato presente	Ingressi		Stato futuro
	autoNS	autoEO	
verdeNS	0	0	verdeNS
verdeNS	0	1	verdeEO
verdeNS	1	0	verdeNS
verdeNS	1	1	verdeEO
verdeEO	0	0	verdeEO
verdeEO	0	1	verdeEO
verdeEO	1	0	verdeNS
verdeEO	1	1	verdeNS

Si noti che la specifica non indicava cosa fare nel caso in cui vi fossero delle auto in entrambe le direzioni: in questo caso la funzione di stato futuro sopra riportata modifica lo stato in modo da garantire che un flusso continuo di auto in una direzione non possa bloccare le auto nell'altra direzione.

La macchina a stati finiti risulta completamente specificata indicando anche la funzione di uscita:

Stato presente	Uscite	
	luceNS	luceEO
verdeNS	1	0
verdeEO	0	1

Prima di passare all'implementazione, conviene esaminare una rappresentazione grafica, molto usata per rappresentare le macchine a stati finiti. In questa rappresentazione si usano i nodi per indicare gli stati; all'interno dei nodi si riporta l'elenco delle uscite affermate in tale stato. Si usano degli archi orientati per indicare la funzione di stato futuro, in cui le etichette sugli archi specificano le condizioni di ingresso mediante funzioni logiche. La rappresentazione grafica della macchina a stati finiti appena vista è riportata in figura B.28.

Una macchina a stati finiti può essere implementata mediante un registro per memorizzare lo stato presente ed un circuito logico combinatorio per calcolare le funzioni di stato futuro e di uscita; la figura B.29 mostra l'aspetto di un'ipotetica macchina a stati finiti con 4 bit di stato, corrispondenti a 16 stati. Per implementare

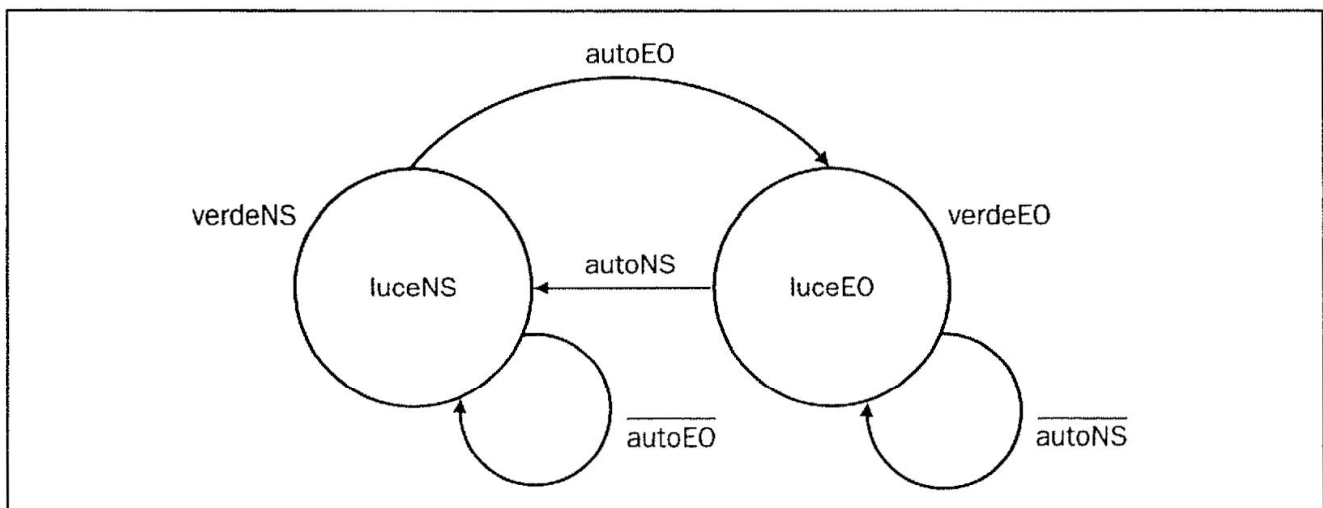


Figura B.28 Rappresentazione grafica del controllore di semaforo a due stati. Le funzioni logiche relative alle transizioni di stato sono state semplificate; ad esempio, la transizione da verdeNS a verdeEO si può ricavare dalla tabella degli stati futuri come $(\text{autoNS} \cdot \text{autoEO}) + \text{autoNS} \cdot \text{autoEO}$ equivalente ad autoEO .

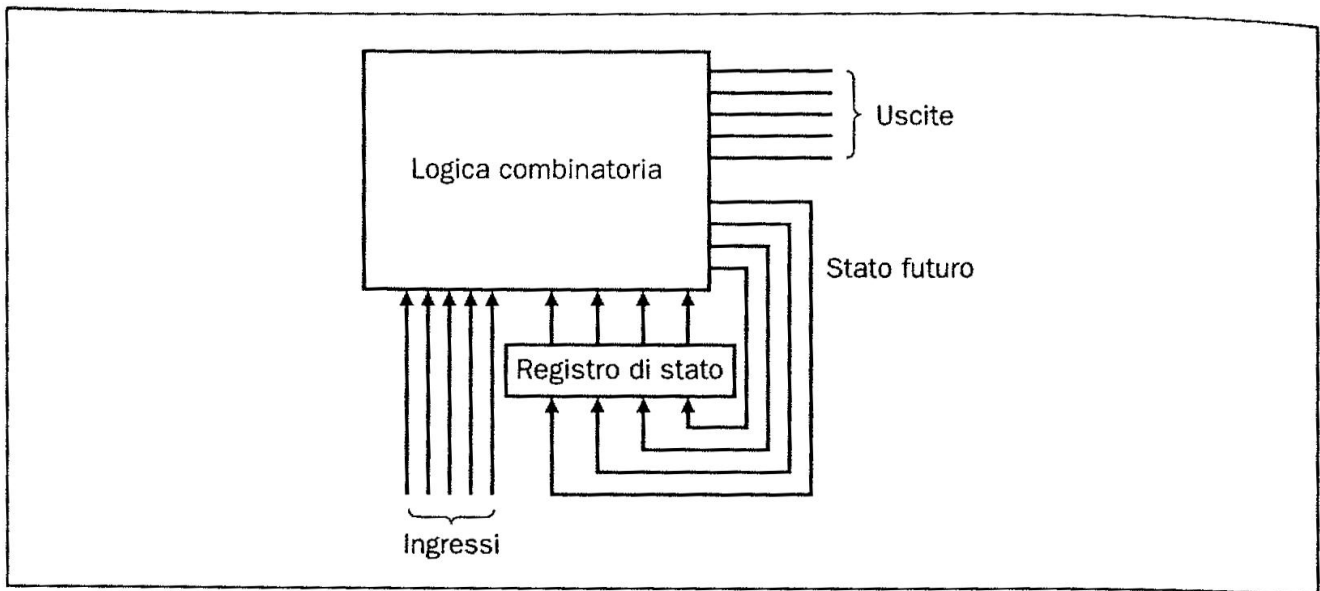


Figura B.29 L'implementazione di una macchina a stati finiti consiste di un registro di stato che contiene lo stato presente e di un circuito logico combinatorio che calcola le funzioni di stato futuro e di uscita. Le due funzioni sono spesso divise ed implementate da due circuiti logici separati, il che spesso fa risparmiare porte logiche.

in tal modo la macchina a stati finiti occorre innanzitutto assegnare dei valori numerici agli stati, mediante un processo che prende il nome di *codifica degli stati* o *assegnazione degli stati*. Per esempio si potrebbe assegnare lo stato 0 a verdeNS e lo stato 1 a verdeEO, ottenendo un registro di stato di un solo bit. La funzione di stato futuro diverrà quindi

$$\text{stato futuro} = \overline{(\text{stato presente} \cdot \text{autoEO})} + (\text{stato presente} \cdot \overline{\text{autoNS}})$$

dove lo stato presente è il contenuto del registro di stato (0 o 1) e lo stato futuro è il valore dell'uscita della funzione di stato futuro, che verrà scritto nel registro di stato al termine del ciclo di clock. La funzione di uscita è altrettanto semplice:

$$\text{luceNS} = \overline{\text{stato presente}}$$

$$\text{luceEO} = \text{stato presente}$$

Il circuito logico combinatorio è spesso implementato ricorrendo a logica strutturata, ad esempio una PLA, che può essere costruita automaticamente a partire dalle tabelle delle funzioni di stato futuro e di uscita. Vi sono infatti programmi di progettazione assistita da calcolatore (CAD, computer-aided design) che partono da una rappresentazione grafica o testuale di una macchina a stati finiti e producono automaticamente un'implementazione ottimizzata. Nei Capitoli 5 e 6 le macchine a stati finiti sono utilizzate nelle unità di controllo dell'esecuzione del processore; l'Appendice C analizza in dettaglio l'implementazione di tali unità di controllo sia mediante PLA sia mediante ROM.