

Analisi e sintesi di circuiti sequenziali

Circuiti combinatori vs. sequenziali

Circuiti combinatori

- L'output è funzione degli input correnti
- Non c'è informazione di stato
- Non memorizzano informazioni

Circuiti sequenziali (modellabili mediante finite state machine)

- L'output è funzione dello stato corrente (e eventualmente degli ingressi)
- Possiedono informazioni di stato
- Richiedono la memorizzazione di informazioni
- Utilizzano flip-flop per implementare la memorizzazione delle informazioni

Circuiti Sincroni e Asincroni

Circuiti Sincroni

- Guidati da un clock
- Tutti i flip-flop usano lo stesso clock e cambiano stato sui fronti di salita/discendenza del clock

Circuiti Asincroni

- Nessun clock
- Possono cambiare stato in qualsiasi istante
- Più veloci in teoria (non devono attendere il clock) ma soggetti a ritardi variabili

Modello generale di circuiti sequenziali

- Due parti:
 - A. i **flip-flop che implementano la parte di memorizzazione**
 - B. la **logica combinatoria per produrre gli input per i flip flop e per produrre l'output**
- La logica combinatoria può essere realizzata mediante **gate, oppure mediante ROM e PLA**

Modelli di circuiti sequenziali

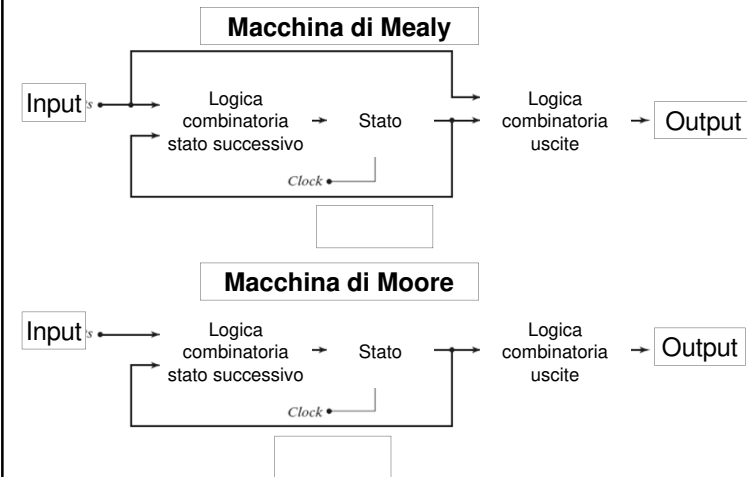
Macchine di Mealy

- Gli output sono funzione dello stato corrente e degli input
- La macchina a stati prevede valori di ingresso e di uscita per ciascuna transizione tra stati

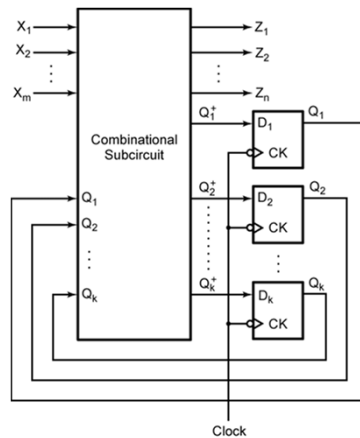
Macchine di Moore

- Gli output sono funzione dello stato corrente
- Gli output sono indipendenti dagli input
- La macchina a stati prevede un valore di output per ciascuno stato

Macchine di Mealy e Moore



Modello macchina di Mealy



Modello macchina di Moore

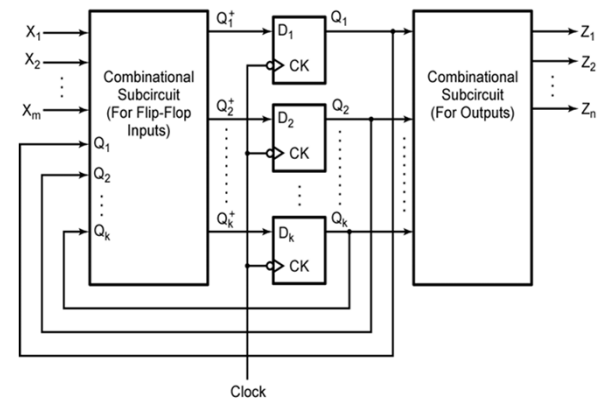


Diagramma degli stati di un circuito sequenziale

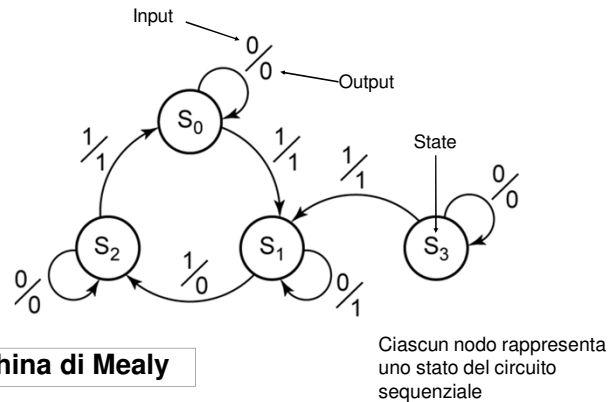
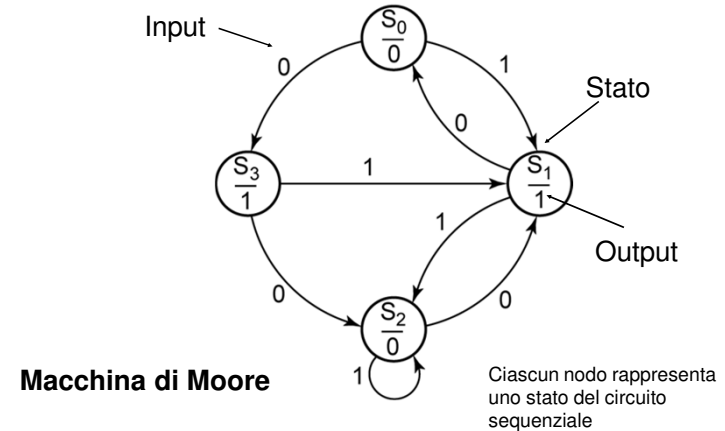


Diagramma degli stati di un circuito sequenziale



Progettazione circuiti sequenziali

Progettazione circuiti sequenziali

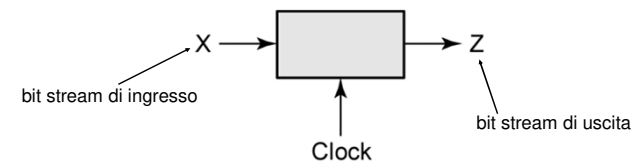
- Analisi specifiche
- Costruzione diagramma degli stati
- Costruzione tabella degli stati
- Minimizzazione degli stati (se necessario)
- Codifica degli stati
- Creazione tabella degli stati codificata
- Selezione tipo di flip flop (se non già specificato)
- Determinare espressioni di ingresso dei flip-flop e espressione/i delle uscite
- Disegno diagramma circuito

Esempio: Sequence Detector

Sequence Detector (Mealy)

Per illustrare la progettazione di una macchina di Mealy, progetteremo in **rilevatore di sequenze**

Il circuito è del tipo



Sequence Detector (Mealy)

Supponiamo di voler realizzare un rilevatore di sequenze in maniera tale che **qualsiasi sequenza che termina per 101 venga rilevata e produca un output Z=1**

Il circuito **NON si resetta quando una sequenza viene rilevata**

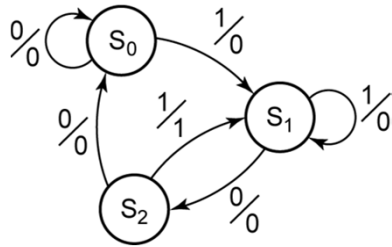
Esempio:

$X =$ 0 0 1 1 0 1 1 0 0 1 0 1 0 1 0 0
 $Z =$ 0 0 0 0 0 1 0 0 0 0 0 1 0 1 0 0
 (time: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15)

Disegniamo la state machine...

- Non sappiamo ancora di quanti flip-flop abbiamo bisogno, quindi rappresentiamo gli stati con S_0, S_1 , etc.
- Usiamo S_0 per rappresentare lo stato iniziale (circuito resettato)
- Se il circuito riceve uno 0, si resta nello stato S_0 perché la sequenza che ci interessa non inizia con uno 0

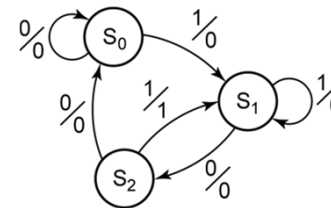
Possibile soluzione



101

(Macchina di Mealy)

Produciamo la tabella degli stati



Di quanti flip-flop abbiamo bisogno?

Siccome abbiamo 3 stati, ci occorrono 2 bit di codifica → 2 flip-flop, uno per ciascun bit

Stato corrente	Stato successivo		Output	
	X = 0	X = 1	X = 0	X = 1
S ₀	S ₀	S ₁	0	0
S ₁	S ₂	S ₁	0	0
S ₂	S ₀	S ₁	0	1

Creiamo la tabella delle transizioni

Stato corrente	Stato successivo		Output	
	X = 0	X = 1	X = 0	X = 1
S ₀	S ₀	S ₁	0	0
S ₁	S ₂	S ₁	0	0
S ₂	S ₀	S ₁	0	1

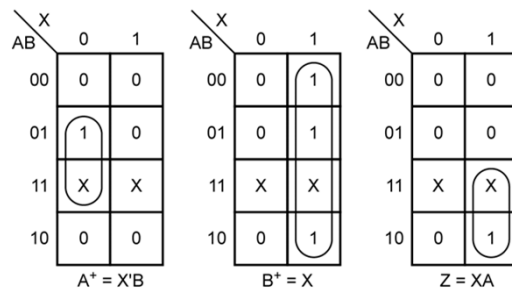
AB	A ⁺ B ⁺		Z	
	X = 0	X = 1	X = 0	X = 1
00	00	01	0	0
01	10	01	0	0
10	00	01	0	1

Implementiamo il
circuito con Flip-Flop D

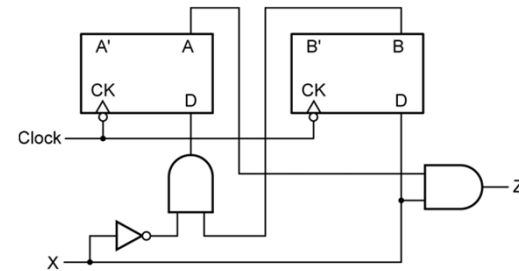
Mappe di Karnaugh

AB	A^*B^*		Z	
	X = 0	X = 1	X = 0	X = 1
00	00	01	0	0
01	10	01	0	0
10	00	01	0	1

Due per pilotare i flip-flop, una per l'uscita (Z)



Rappresentazione circuitale



Macchina di Moore

- La procedura per derivare l'automa di Moore è simile a quella usata per l'automa di Mealy.
- Differenza: **l'output è codificato nello stato**
- Quindi dobbiamo modificare la macchina precedente in maniera tale che vada in uno stato con output 1 se all'ingresso arriva una sequenza che termina con 101

Progettiamo l'automa
a stati...

Possibile soluzione

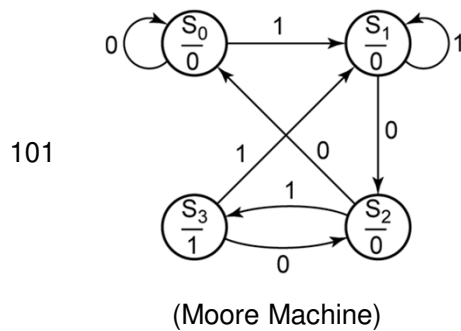
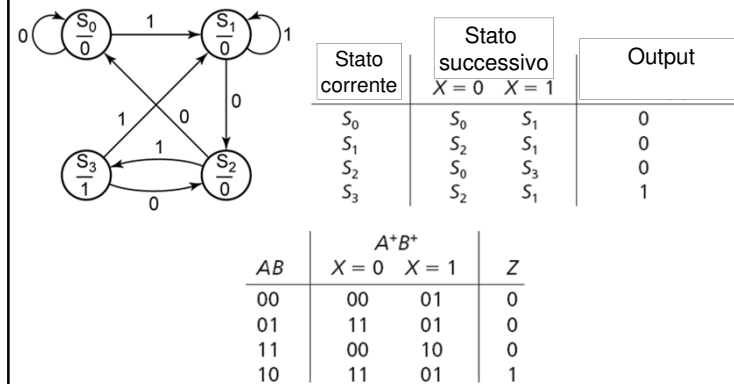


Tabella degli stati



Mappe di Karnaugh

AB	A*B*		Z
	X = 0	X = 1	
00	00	01	0
01	11	01	0
11	00	10	0
10	11	01	1

AB \ X	0	1
00	0	0
01	1	0
11	0	1
10	1	0

D_A

AB \ X	0	1
00	0	1
01	1	1
11	0	0
10	1	1

D_B

Z=AB'

Mappe di Karnaugh

$$D_A = A'BX' + ABX + AB'X' = X'(A \oplus B) + ABX$$

$$D_B = A'B + AB' + A'X = A \oplus B + A'X$$

AB \ X	0	1
00	0	0
01	1	0
11	0	1
10	1	0

D_A

AB \ X	0	1
00	0	1
01	1	1
11	0	0
10	1	1

D_B

Z=AB'

Esempio

Vediamo a questo punto un esempio più complesso...

Z vale uno se le sequenza di ingresso termina per 010 oppure per 1001, zero altrimenti, senza RESET

Esempio:

X =	0	0	1	0	1	0	0	1	0	0	0	1	0	0	1	1	0
					↑		↑		↑	↑			↑		↑		
					a		b		c	d			e		f		
Z =	0	0	0	1	0	1	0	1	1	0	0	0	1	0	1	0	0

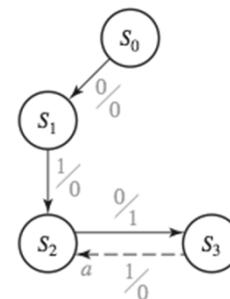
Progettiamo l'automa a stati... (Mealy)

Come procediamo?

- Cerchiamo di gestire una sequenza alla volta
- Rappresentiamo *ciascuno stato in termini degli ultimi n bit* che sono giunti all'ingresso del circuito

Prima bozza

Individua la sequenza 010

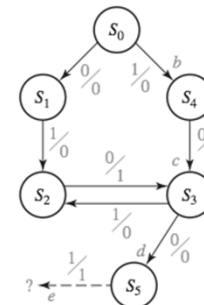


Stato	Sequenza ricevuta
S ₀	Reset
S ₁	0
S ₂	01
S ₃	010

Come proseguiamo?

- Ragioniamo sulle sequenze che gli stati codificati finora catturano...
- A questo punto dovrei aggiungere uno stato al quale arrivo da S_0 quando ho un ingresso 1
- Lo chiamo S_4
- Se poi in S_4 mi arriva 0 dove vado?
 - Notare: ho appena ottenuto la sequenza 1 0
- C'è uno stato tra quelli esistenti che raccoglieva la sequenza 1 0 (S_3) ?

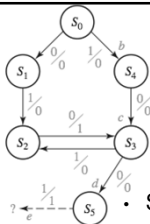
Aggiungiamo ulteriori stati



Stato	Sequenza termina per
S_0	Reset
S_1	0 (ma non 10)
S_2	01
S_3	10
S_4	1 (ma non 01)
S_5	100

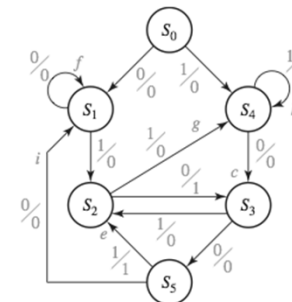
010 1001

Discussione

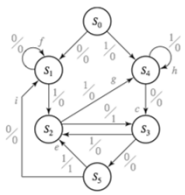


- Se dal reset mi arriva 1 vado in S_4 anziché in S_1
- Se a questo punto mi arriva uno 0 vado in S_3
 - **Perché?** Perché anche nel caso precedente in S_3 andavo quando ricevevo una sequenza che terminava per 10
- S_2 , invece, raccoglie sequenze che terminano per 01
- Se da S_3 mi arriva 0, vado in S_5 , ovvero sto raccogliendo una sequenza 100
- Il prossimo passo da S_5 sarà intercettare un 1 e andare in un altro stato producendo un 1 in uscita

Passo successivo...



Stato	Sequenza termina per
S_0	Reset
S_1	0 (ma non 10)
S_2	01
S_3	10
S_4	1 (ma non 01)
S_5	100



Discussione

- Da S₅ con ingresso 1 vado in S₂ che raccoglie una sequenza 01
 - un successivo 0 mi porterebbe in S₃ con uscita 1 (sequenza 010)
 - invece, un successivo 1 in S₂ mi porta in S₄ (1 ma non 01)
- Invece, da S₅ con ingresso 0 vado in S₁ perché ho raccolto 0 come ultimo valore, ma non sono nel caso 10 (ovvero S₃)
- Inoltre, uno 0 raccolto in S₁ e un 1 raccolto in S₄ provocano delle self-transizioni

Suggerimenti per la creazione delle state machine

Suggerimenti...

Non esiste una procedura specifica, ma è possibile seguire le seguenti linee guida:

1. Provare a costruire semplici sequenze di input/output per comprendere meglio le specifiche del problema
2. Determinare sotto quali condizioni il circuito si resetta (ovvero torna allo stato iniziale)
 - Tipicamente, al verificarsi di sequenze non contenute nelle sequenze da riconoscere, oppure situazioni di reset esplicite dalle specifiche
3. Se esistono poche (una o due) sequenze che producono output≠0, provare a costruire delle macchine parziali per tali sequenze (come fatto prima per 010 e 1001)

Suggerimenti (cont.)

4. Oppure (sempre come fatto prima) identificare sequenze che il circuito deve ricordare (e.g. 01, 10, 100, etc.) e identificare stati in corrispondenza di tali sequenze
5. **Per ogni transizione che aggiungiamo, valutiamo se dobbiamo aggiungere un nuovo stato (si verifica una situazione mai incontrata prima) o se invece devo finire in uno stato già esistente (perché ho appena raccolto una sequenza che quello stato si aspetta)**
6. **Dato un ingresso (o una combinazione di ingressi per circuiti a più ingressi), deve esserci una sola uscita da uno stato in corrispondenza di tale combinazione.**
7. Una volta che l'automa è completo, effettuare un primo test utilizzando le sequenze al punto (1), anche se ciò non è sufficiente per dimostrare la correttezza del modello.

Esempio (Mealy)

Un circuito ha un ingresso (X) e un'uscita Z. Il circuito produce un'uscita Z=1 in corrispondenza delle sequenze 0101 o 1001

Dopo aver ricevuto **4 input**, **il circuito si resetta SEMPRE**

Esempio:

$X =$	0101		0010		1001		0100
$Z =$	0001		0000		0001		0000

Progettiamo l'automa a stati... (Mealy)

Come procediamo?

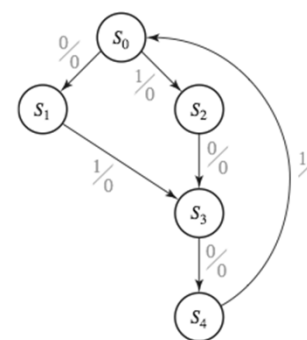
$X =$	0101		0010		1001		0100
$Z =$	0001		0000		0001		0000

Notiamo che una sequenza 10 o 01 seguita da 01 produce sempre 1

Quindi una sequenza 10 o 01 dovrebbe portarci nello stesso stato

0101 o 1001

Macchina parziale...



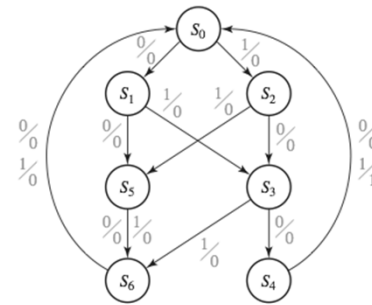
Stato	Sequenza ricevuta
S_0	Reset
S_1	0
S_2	1
S_3	01 or 10
S_4	010 or 100

0101 1001

Come completiamo la macchina?

Siccome sappiamo che comunque dopo 4 input il circuito si resetta, a partire dagli stati esistenti, costruiamo le rimanenti sequenze che ci riportano in S_0 con uscita 0

Automa completo



Stato	Sequenza ricevuta
S_0	Reset
S_1	0
S_2	1
S_3	01 o 10
S_4	010 o 100
S_5	Due input ricevuti, 1 impossibile
S_6	Tre input ricevuti, 1 impossibile

0101 1001

Automi a stati completamente specificati

Godono delle seguenti proprietà:

1. Esiste almeno un'uscita da ogni stato per ogni valore d'ingresso
2. Non esistono più archi di uscita da uno stato in corrispondenza dello stesso ingresso

Esercizi

Esercizio

- Realizzare un circuito che, una volta abilitato, provochi l'attivazione di un semaforo per regolare la partenza di una gara di F1
- In altri termini, il semaforo è dotato di 5 luci, che si accendono in sequenza.
- Assumiamo che dopo un secondo dall'accensione dell'ultima luce, tutte le luci si spengano (e restino spente)

Esercizio

- Realizzare un circuito in grado di regolare il funzionamento di un semaforo in un incrocio. L'incrocio è dotato di due semafori, A e B
- Il circuito funziona nel seguente modo:
 - dal secondo 0 al secondo 5: A verde e B rosso
 - dal secondo 6 al secondo 7: entrambi i semafori rossi
 - dal secondo 8 al secondo 13: A rosso, B verde
 - dal secondo 14 al secondo 15: entrambi i semafori rossi
- Utilizzare flip-flop J-K in configurazione T

Esercizio

- Un circuito sequenziale è dotato di un ingresso e un uscita
- L'uscita diventa 1 (e rimane 1) non appena il circuito, partendo dallo stato iniziale, ha ricevuto almeno due zeri e almeno due uno
(indipendentemente dall'ordine con cui questi siano stati ricevuti)
- Derivare un automa di Moore

Esercizio

- Un circuito sequenziale è dotato di un ingresso X e due uscite Z1 e Z2
- $Z1=1$ non appena il circuito riceve la sequenza 010, a patto che la sequenza 100 non sia mai apparsa
- $Z2=1$ quando il circuito ottiene la sequenza 100
- Nota: se Z2 assume un valore 1, non potrà più accadere che $Z1=1$, ma non è vero il contrario

Esercizio

- Un circuito sequenziale ha due ingressi (X_1, X_2), e un'uscita Z
- L'output parte dal valore 0 e resta costante a meno che non si verifichino le sequenze:
 - $(X_1, X_2) = 01, 00 \rightarrow Z$ diventa 0
 - $(X_1, X_2) = 11, 00 \rightarrow Z$ diventa 1
 - $(X_1, X_2) = 10, 00 \rightarrow Z$ cambia valore

Derivare l'automa di Moore