

Lez. 5 – Circuiti sequenziali e Macchine a Stati Finiti

Laboratorio di Architettura degli Elaboratori

Stefano Centomo

10-13 Dicembre 2021

- 1 Circuiti Sequenziali e Macchine a Stati Finiti
- 2 Circuiti sequenziali in SIS
- 3 Esercizi

1 Circuiti Sequenziali e Macchine a Stati Finiti

2 Circuiti sequenziali in SIS

3 Esercizi

Un **circuito sequenziale** può essere modellato utilizzando una **macchina a stati finiti** (Finite State Machine - FSM) definita come una 6-upla $M = (S, I, O, \delta, \lambda, s)$ dove:

S insieme degli stati

I insieme degli ingressi

O insieme delle uscite

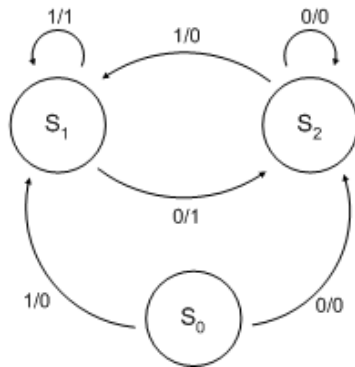
δ funzione di stato prossimo che ad una coppia (ingresso, stato presente) associa lo stato prossimo

λ funzione d'uscita che associa un valore per l'uscita ad uno stato presente (*FSM di Moore*) oppure ad una coppia (ingresso, stato presente) (*FSM di Mealy*)

s stato di reset (a volte potrebbe non essere definito)

State Transition Graph

Solitamente una FSM si rappresenta per mezzo di un **grafo delle transizioni** (State Transition Graph - STG) dove i nodi sono gli stati e gli archi rappresentano le transizioni da uno stato all'altro. Ad ogni transizione corrisponde un insieme dei valori di ingresso ed un insieme dei valori di uscita.



I circuiti sequenziali possono essere **sincroni** se i valori delle uscite assumono significato in corrispondenza di un evento su un segnale di sincronismo (solitamente detto *clock*), oppure **asincroni** se i valori delle uscite cambiano al variare degli ingressi senza tener conto del segnale di sincronismo.

L'uscita di un circuito sequenziale dipende dai valori in ingresso anche negli istanti passati, occorre quindi una sorta di memoria di tale storia passata.

Questa memoria è rappresentata dallo stato.

L'elettronica digitale mette a disposizione due tipi di componenti detti **latch** e **flip-flop** in grado di memorizzare il valore di un bit (stato binario).

Occorre quindi codificare tutti gli stati che può assumere il circuito mediante dei numeri binari e poi utilizzare dei latch o flip-flop per memorizzare tali numeri.

L'uscita di un circuito sequenziale dipende dai valori in ingresso anche negli istanti passati, occorre quindi una sorta di memoria di tale storia passata. Questa memoria è rappresentata dallo stato.

L'elettronica digitale mette a disposizione due tipi di componenti detti **latch** e **flip-flop** in grado di memorizzare il valore di un bit (stato binario). Occorre quindi codificare tutti gli stati che può assumere il circuito mediante dei numeri binari e poi utilizzare dei latch o flip-flop per memorizzare tali numeri.

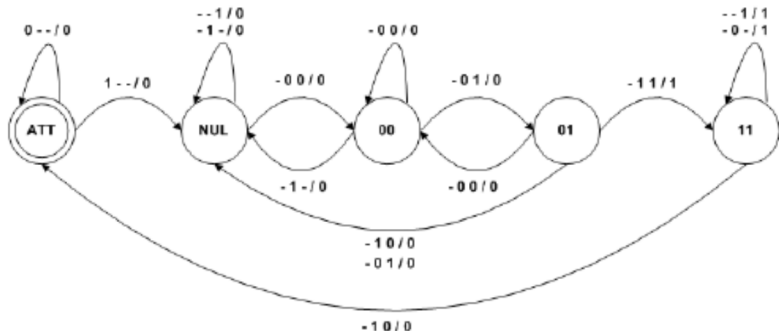
Una FSM con N stati necessita di $\log_2 N$ componenti di memoria elementari.

- 1 Circuiti Sequenziali e Macchine a Stati Finiti
- 2 Circuiti sequenziali in SIS
- 3 Esercizi

Per modellare un circuito sequenziale in SIS è necessario definire la tabella delle transizioni ed eventualmente definire manualmente la codifica degli stati (esiste un comando che permette di eseguire la codifica in modo automatico).

Esempio

Si consideri il circuito sequenziale che è in grado di riconoscere la sequenza di ingresso **00 01 11**, (ingresso IN a 2 bit). Il circuito è attivo ed inizia ad analizzare i valori dell'ingresso IN quando l'ingresso START passa da 0 a 1. Nello stesso ciclo di clock in cui viene riconosciuta la sequenza **00 01 11**, l'uscita OUT passa da 0 a 1. OUT rimane a 1 fino a quando gli ingressi assumeranno il valore 10; momento in cui il circuito viene nuovamente posto in attesa che il segnale START passi da 0 a 1.

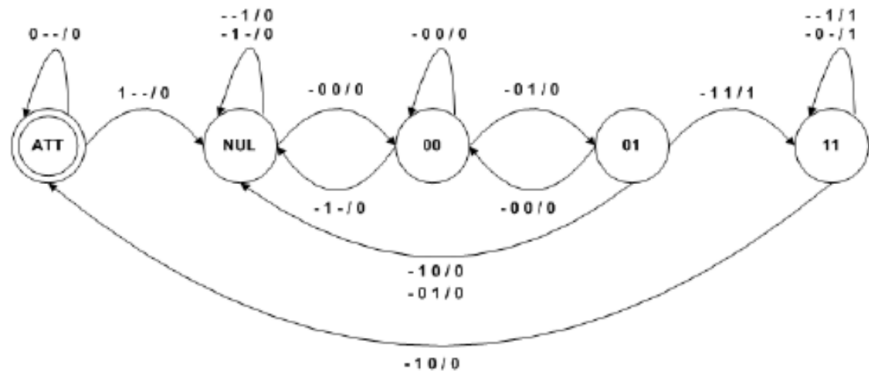


Esempio

Scegliendo di codificare gli stati come **ATT** = 000, **NUL** = 001, **00** = 010, **01** = 011 e **11** = 100, si ottiene la tabella di verità della funzione dello stato prossimo e dell'uscita OUT

a2	a1	a0	START	IN1	IN0	s2	s1	s0	OUT
0	0	0	0	-	-	0	0	0	0
0	0	0	1	-	-	0	0	1	0
0	0	1	-	0	0	0	1	0	0
0	0	1	-	-	1	0	0	1	0
0	0	1	-	1	-	0	0	1	0
0	1	0	-	0	0	0	1	0	0
0	1	0	-	0	1	0	1	1	0
0	1	0	-	1	-	0	0	1	0
0	1	1	-	1	1	1	0	0	1
0	1	1	-	0	0	0	1	0	0
0	1	1	-	1	0	0	0	1	0
0	1	1	-	0	1	0	0	1	0
1	0	0	-	1	0	0	0	0	0
1	0	0	-	-	1	1	0	0	1
1	0	0	-	0	-	1	0	0	1

Esempio



- 1 La tabella delle transizioni deve essere descritta all'interno della sezione delimitata dalle keyword `.start_kiss` e `.end_kiss`. Le transizioni devono essere specificate come un insieme di righe che riportano in ordine: valore degli ingressi, stato presente, stato prossimo, valore delle uscite.

La tabella delle transizioni deve essere preceduta da 5 righe che specificano:

- il numero di segnali di input
- il numero di segnali di output
- il numero di transizioni
- il numero di stati
- lo stato di reset

- ② Dopo la tabella delle transizioni (dopo `.end_kiss`) possono essere riportate le istruzioni necessarie per definire la codifica degli stati qualora non si decida di farla definire a SIS in modo automatico. La keyword da utilizzare per definire la codifica è `.code` seguita dal nome dello stato e dalla sua codifica binaria.
- ③ Dopo aver modellato il circuito, è possibile procedere con la minimizzazione degli stati. Una volta caricato il file `.blif`, la minimizzazione degli stati si esegue con il comando `state_minimize` `stamina`.

- 4 Generazione logica funzioni δ e λ :
 - se il file contiene già la codifica binaria degli stati (`.code`) allora occorre generare le funzioni δ e λ con il comando `stg_to_network`
 - altrimenti occorre assegnare automaticamente gli stati con il comando `state_assign jedi` (che genera anche le funzioni δ e λ). Attenzione che prima occorre minimizzare gli stati e poi farne l'assegnazione.
- 5 Eseguire la minimizzazione delle funzioni δ e λ , ad esempio lanciando lo script `script.rugged` come visto per la minimizzazione dei circuiti combinatori.

- 1 Circuiti Sequenziali e Macchine a Stati Finiti
- 2 Circuiti sequenziali in SIS
- 3 Esercizi**

Esercizio 1

Realizzare il circuito sequenziale (2 ingressi, 2 uscite) corrispondente alla seguente tabella degli stati. Sia A lo stato di reset. Minimizzare il numero degli stati e ottimizzare la logica combinatoria.

	00	01	10	11
A	D/10	D/00	A/11	D/11
B	B/1-	A/--	B/-1	C/--
C	C/0-	H/-1	I/-0	A/-0
D	A/10	A/-0	D/11	A/11
E	G/10	G/--	C/-1	I/1-
F	H/--	H/-0	A/01	B/10
G	A/10	G/-0	H/00	D/11
H	D/1-	G/-0	H/00	A/-1
I	F/--	F/11	E/00	A/01

Realizzare un circuito sequenziale con due ingressi binari in grado di riconoscere la sequenza **00 11 00 01**. Si verifichi il comportamento del circuito durante il riconoscimento della sequenza **00 11 00 11 00 01**.

Su due linee vengono trasmessi serialmente dei numeri binari a partire dal bit meno significativo (un bit ogni colpo di clock). Si progetti un sommatore seriale in grado di fornire in uscita, per ogni colpo di clock relativo a ciascun bit trasmesso, il corrispondente bit di somma. Minimizzare il numero degli stati e ottimizzare la logica combinatoria nei 2 casi:

- 1 codifica manuale degli stati
- 2 codifica automatica degli stati