

EsD3: Macchine a stati finiti: semafori e riconoscitore di fronti

Gruppo 1.AC
Matteo Rossi, Bernardo Tomelleri

5 maggio 2022

1 Misura componenti dei circuiti

Riportiamo per completezza il valore della tensione continua di alimentazione per i circuiti integrati misurata con il multimetro

$$V_{CC} = 4.99 \pm 0.03 \text{ V}$$

e il valore di capacità del condensatore di disaccoppiamento che collega le linee di alimentazione a massa (sempre misurato con il multimetro)

$$C_d = 97 \pm 4 \text{ nF}$$

2 Implementazione di un semaforo con circuiti integrati

2.a Diagramma a stati del semaforo

Quando il semaforo è in modalità “abilitato” ($E = 1$) si susseguono ciclicamente 3 possibili output, per cui sono stati implementarli con 3 stati interni della macchina. Mentre nel caso di semaforo “disabilitato” ($E = 0$) i possibili output si riducono a 2, e si possono esprimere utilizzando due degli stati precedentemente codificati a seconda del valore logico dell’input ENABLE tramite un circuito FSM di Mealy.

Scegliamo di codificare con due bit di memoria (Q_1Q_0) i tre stati: 00, 01 e 10 che corrispondono, quando $E = 1$, ai tre output “verde” (V), “verde-giallo” (VG), “rosso” (R). Quando invece $E = 0$ vengono usati solamente i primi due 00 e 01, che corrispondono agli output del semaforo “spento” (Y^0) e “giallo” (Y^1) come si può vedere già nel diagramma a stati riportato in fig. 1.

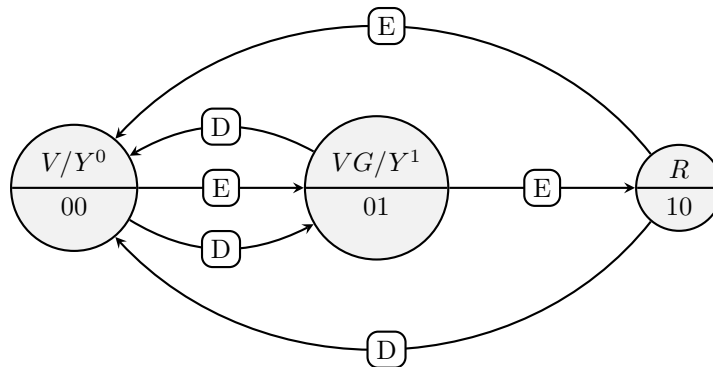


Figura 1: Diagramma degli stati del semaforo FSM di Mealy con Enable

2.b Codifica degli stati della macchina

La codifica in termini di bit degli stati della macchina è riassunta nella tabella 1, dove Q_0 e Q_1 corrispondono rispettivamente al valore logico delle uscite del primo e del secondo flip-flop, mentre S_E e S_D rappresentano lo stato associato al semaforo a seconda che questo sia ENABLED o DISABLED.

S_D	S_E	Q_1	Q_0
Y^0	V	0	0
Y^1	VG	0	1
	R	1	0

Tabella 1: Codifica binaria scelta per gli stati del semaforo.

ENABLE	current state			next state		
	LED	Q_1	Q_0	D_1	D_0	LED
1	V	0	0	0	1	VG
	VG	0	1	1	0	R
	R	1	0	0	0	V
	\ominus	1	1	X	X	\ominus
0	Y^0	0	0	0	1	Y^1
	Y^1	0	1	0	0	Y^0
	Y^0	1	0	0	0	Y^0
	\ominus	1	1	X	X	\ominus

Tabella 2: Tabella di verità per le transizioni di stato del semaforo Mealy

E	Q_0	Q_1	G	Y	R
1	0	0	1	0	0
1	0	1	1	1	0
1	1	0	0	0	1
1	1	1	X	X	X
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	X	X	X
0	1	1	X	X	X

Tabella 3: Corrispondenza tra gli stati della FSM e i valori di tensione digitali in ingresso ai LED del semaforo

2.c Tabelle di verità

Come visto nel diagramma degli stati il segnale di input ENABLE modifica l'uscita del circuito in maniera asincrona, mentre dalla tabella di verità (tabella 2) si intuisce come lo stato futuro dipenda da quello precedente incrementando come un contatore in maniera sincrona rispetto al segnale di clock.

Come esplicitato nella tabella 3 possiamo collegare gli output della macchina ai led di colori Verde, Giallo e Rosso secondo lo schema logico seguente:

LED verde acceso $G = Q_1 = 0 \ (\iff \overline{Q_1} = 1)$

LED giallo acceso $Y = Q_0 = 1$

LED rosso acceso $R = Q_1 = 1$

in modo tale che nella realizzazione pratica del circuito, in corrispondenza degli opportuni stati della FSM in uscita dai Flip-Flop si accendano le corrette combinazioni di LED (ad esempio: $Y = Q_0 \implies$ LED giallo in serie a Q_0).

Occorre specificare che, nel caso in cui $E = 0$, $Q_0 = 1$ e $Q_1 = 0$ (riportato come X nella tabella 3, produce l'output "rosso". Questo vale a dire che se il semaforo è "rosso" con ENABLE alto e questo viene messo a zero, rimarrà rosso fino al fronte di salita successivo del clock, dopo di cui si spegne come previsto dalla modalità disabilitato. Mentre per quanto riguarda l'accensione dei led verde, questa è consentita solo quando ENABLE = 1

- $\overline{Q_1} \cdot E \implies$ LED verde acceso (G)
- $Q_1 \implies$ LED rosso acceso (R)

AND-gate per LED rosso Facendo uso di una quarta porta AND è possibile rendere l'accensione del LED rosso dipendente dal segnale di ENABLE (ovvero $R = Q_1 \cdot E$) in maniera analoga a quanto fatto per il LED verde. Dalla tabella 2 infatti si nota che l'unica differenza di funzionamento si avrebbe durante la transizione di $E : 1 \rightarrow 0$. Per cui in questa variante del circuito, lo spegnimento diventerebbe asincrono rispetto al segnale di clock anziché sincrono.

2.d Mappe di Karnaugh e logica combinatoria

Si riportano di seguito le tabelle di Karnaugh impiegate per derivare le espressioni logiche degli stati futuri $D_1 D_0$ in funzione di quelli correnti $Q_1 Q_0$:

$E \backslash Q_1 Q_0$	00	01	11	10
0	1	0	X	0
1	1	0	X	0

Tabella 4: Tabella di Karnaugh per $D_0 = \overline{Q_0} \cdot \overline{Q_1}$

$E \backslash Q_1 Q_0$	00	01	11	10
0	0	0	X	0
1	0	1	X	0

Tabella 5: Tabella di Karnaugh per $D_1 = E \cdot Q_0$

2.e Costruzione del circuito

Si è montato il circuito riportato in fig. 2 con due positive-edge-triggered D Flip-Flop (da chip integrato 74LS74) e 3 porte AND (da chip integrato 74LS08) e si sono collegati i pin PRESET e CLEAR dei D-FF a V_{CC} onde evitare reset o clear spuri.

3 Implementazione software della logica combinatoria con AD2/ROM

3.a Costruzione del circuito

3.b Implementazione delle tabelle di verità in ROM

3.c Verifica del funzionamento del circuito

3.d Variante svizzera del semaforo ROM

4 Implementazione in software dei semafori con MCU (Arduino)

Si vuole ricostruire il circuito precedente per il controllo di un semaforo tramite un microcontrollore (nel nostro caso utilizzeremo Arduino UNO); a questo scopo utilizzeremo come componenti uno switch, 3 led (rosso, verde e giallo) e una resistenza da 330 Ω per limitare la corrente che scorre nei led.

4.a Collegamento LED semaforo alle uscite

4.b Definizione interruttore di Enable

4.c Implementazione del codice per la FSM

4.d Versione svizzera del semaforo con Enable via Arduino

5 Falling-edge detector

5.a Progettazione FSM e costruzione dei circuiti

Si vuole realizzare una FSM che riceve uno stream di bit su una linea di ingresso e che accende un LED tutte le volte che su questo si presenta un fronte di discesa secondo il modello di Moore e un'altra secondo quello di Mealy.

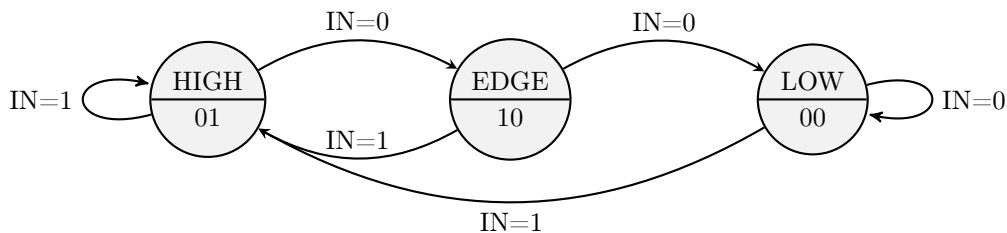


Figura 4: Edge detector FSM di Moore

State	IN	Q_1	Q_0	D_1	D_0	OUT
LOW	0	0	0	0	0	0
	1	0	0	0	1	0
EDGE	0	1	0	0	0	1
	1	1	0	0	1	1
HIGH	0	0	1	1	0	0
	1	0	1	0	1	0

Tabella 6: codifica binaria degli stati del detector di Moore.

5.b Definizione dello stream di bit casuali in ingresso

Con la funzione Patterns di Waveform si invia un segnale (in DIO 5) di clock di frequenza $f_{\text{clk}} = 1 \text{ kHz}$ al pin (CLK) dei Flip-Flop e si genera con il canale DIO 6 uno stream di dati random alla stessa frequenza $f = f_{\text{clk}}$, già dalla schermata di Patterns è possibile notare come le commutazioni di stato del segnale pseudocasuale avvengano in corrispondenza dei fronti di discesa del segnale di clock.

IN \ Q_1Q_0	00	01	11	10
0	0	0	X	0
1	1	1	X	1

Tabella 7: Tabella di Karnaugh per $D_0 = \text{IN}$

IN \ Q_1Q_0	00	01	11	10
0	0	1	X	0
1	0	0	X	0

Tabella 8: Tabella di Karnaugh per $D_1 = \overline{\text{IN}} \cdot Q_0$

IN \ Q_1Q_0	00	01	11	10
0	0	0	X	1
1	0	0	X	1

Tabella 9: Tabella di Karnaugh per $\text{OUT} = Q_1$

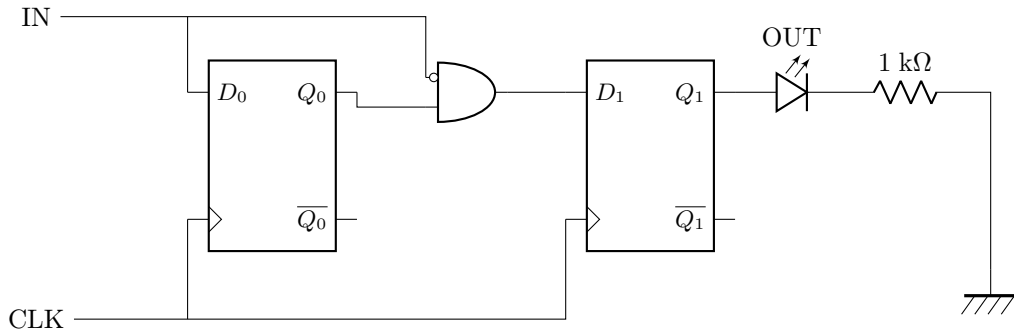


Figura 5: schema del detector Moore.

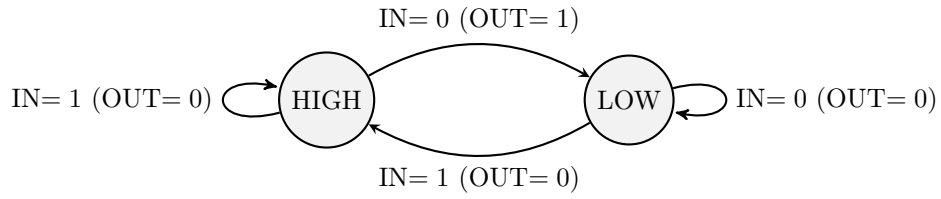


Figura 6: Edge detector FSM di Mealy

$D = \text{IN}$	Q	OUT
0	0	0
0	1	1
1	0	0
1	1	0

Tabella 10: codifica binaria degli stati del detector di Mealy. $D = \text{IN}$; $\text{OUT} = \overline{\text{IN}} \cdot Q$

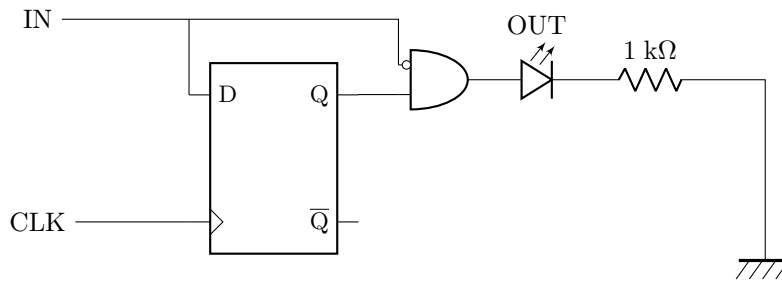


Figura 7: schema del detector Mealy.

Come prima si sono mantenuti i pin PRESET e CLEAR dei D-FF collegati a V_{CC} per evitare reset o clear spuri.

5.c Verifica del funzionamento e analisi della temporizzazione

Si sono acquisiti i segnali in ingresso (CLK = DIO 5, IN = DIO 6) e in uscita (OUT = DIO 7) dai circuiti edge-detector con la funzione Logic Analyzer dell'AD2, di cui riportiamo i risultati per il circuito di Moore in fig. 8 e per la FSM di Mealy in fig. 9. Notiamo nel primo caso come l'uscita assume valore alto in maniera

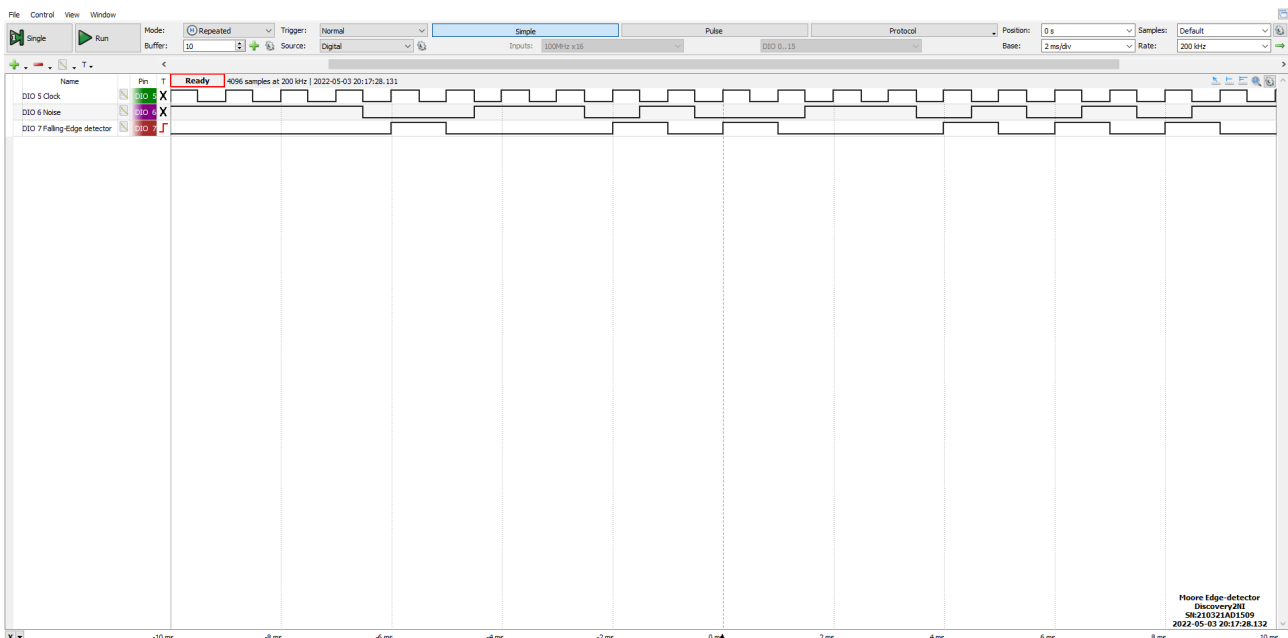


Figura 8: Acquisizione di un ciclo completo (frequenza 1 kHz) con Logic Analyzer dei segnali in ingresso e in uscita dall'edge detector di Moore.

sincrona rispetto al segnale di clock, mentre nell'implementazione di Mealy l'uscita non aspetta il successivo fronte d'onda del clock per salire al livello logico alto e accendere il LED.

Conclusioni e commenti finali

Si è riusciti a progettare, costruire e verificare il corretto funzionamento di circuiti logici combinatori di diversa complessità e svariate applicazioni (e.g., sistemi di controllo e misura) costruiti con porte NOT, NAND, OR e D-FF. Inoltre si è riusciti ad apprezzare le diverse modalità di funzionamento delle macchine a stati finiti implementate secondo i modelli Moore e Mealy, ponendo particolare attenzione alle loro diverse temporizzazioni nei cambiamenti di stato, nonostante la bassa risoluzione temporale dell'AD2.

Dichiarazione

I firmatari di questa relazione dichiarano che il contenuto della relazione è originale, con misure effettuate dai membri del gruppo, e che tutti i firmatari hanno contribuito alla elaborazione della relazione stessa.

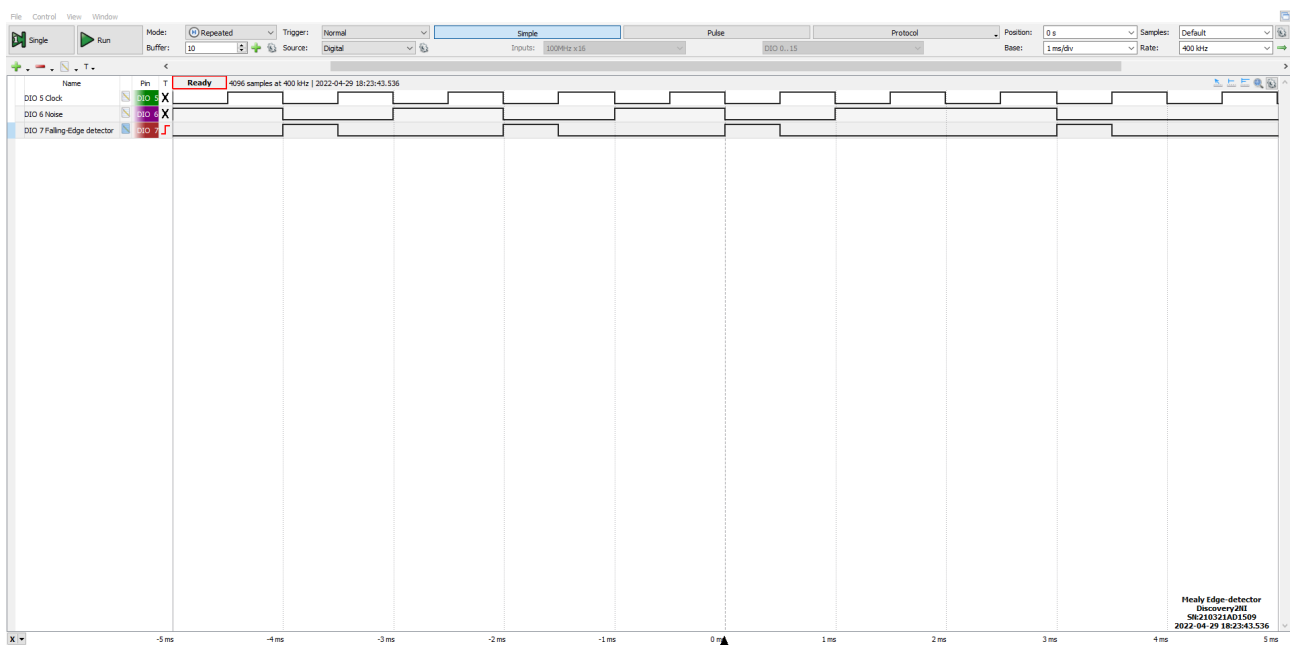


Figura 9: Acquisizione di un ciclo completo (frequenza 1 kHz) con Logic Analyzer dei segnali in ingresso e in uscita dal detector di Mealy.