

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

Gruppo 1.AC
Matteo Rossi, Bernardo Tomelleri

4 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

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

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

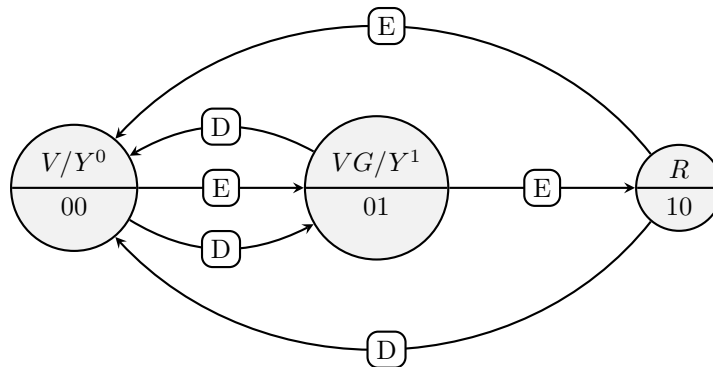
$$C_d = 97 \pm 4 \text{ nF}$$

2 Implementazione di un semaforo con circuiti integrati

2.a Diagramma a stati del semaforo

Nel caso di semaforo “abilitato” i possibili output sono 3 e abbiamo quindi deciso di implementarli con 3 stati interni della macchina; nel caso di semaforo “disabilitato” i possibili output sono 2, ma possono essere ottenuti riutilizzando due degli stati precedenti e l’input “enable” tramite un’opportuno circuito combinatorio (FSM di Mealy).

Si è deciso di codificare in due bit di memoria i tre stati (00), (01) e (10) che corrispondono, quando $E = 1$ (abilitato), ai tre output “verde”, “verde-giallo”, “rosso” (che si susseguono in modo ciclico). Quando invece $E = 0$ (disabilitato) vengono usati solamente (00) e (01), che corrispondono agli output del semaforo “spento” e “giallo”.



2.b Codifica degli stati della macchina

La scelta della codifica in termini di bit è riassunta nella tabella che segue: Q_0 e Q_1 rappresentano rispettivamente lo stato delle uscite del primo e del secondo flip-flop e S è lo stato associato.

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

Tabella 1: codifica binaria degli stati del semaforo.

ENABLE		current state		next state		
		Q_1	Q_0	D_1	D_0	
1	G	0	0	0	1	GY
	GY	0	1	1	0	R
	R	1	0	0	0	G
	...	1	1	X	X	...
0	OFF	0	0	0	1	Y
	Y	0	1	0	0	OFF
	OFF	1	0	0	0	OFF
	...	1	1	X	X	...

Tabella 2: Tabella di verità per il semaforo a 2 bit.

2.c Tabelle di verità

Si nota in particolare dal diagramma degli stati che il segnale di input ENABLE modifica l'output in maniera asincrona, mentre dalla tabella di verità (tabella 2) si può distinguere come lo stato futuro dipenda da quello precedente sfruttando la sincronia del segnale di clock per incrementare il contatore.

Di conseguenza, è possibile associare agli output led G, Y, R una corrispondenza in termini di bit secondo la logica:

- $Q_0 = 1 \rightarrow$ led giallo acceso (Y)
- $Q_1 = 0$ (o equivalentemente: $\overline{Q_1} = 1$) \rightarrow led verde acceso (G)
- $Q_1 = 1 \rightarrow$ led rosso acceso (R)

tale logica è sempre sottomessa al segnale di ENABLE, che consente che i led verde e rosso si possano accendere effettivamente.

Questa codifica è quella sfruttata per la costruzione vera e propria del circuito collegando i led alle uscite dei Flip-Flop (ad esempio: led giallo in serie a Q_0).

Per la precisione, dal momento che l'accensione dei led verde e rosso è consentita solo quando ENABLE = 1, si possono utilizzare due porte AND per definire

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

AND-gate per led rosso Si potrebbe usare una porta AND per rendere l'attivazione del led rosso dipendente dal segnale di ENABLE. Da tabella 2 infatti si nota che l'unica differenza di funzionamento si avrebbe durante la transizione di ENABLE: $1 \rightarrow 0$. In questo caso, lo spegnimento diventerebbe asincrono col segnale di clock anziché sincrono.

2.d Mappe di Karnaugh e logica combinatoria

Le seguenti tabelle di Karnaugh aiutano a derivare le funzioni logiche degli stati futuri in dipendenza da quelli correnti:

2.e Costruzione del circuito

Si è assemblato il circuito riportato in fig. 1 e si sono collegati i pin **preset** e **clear** dei D-FF a V_{CC} per evitare reset o clear spuri.

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

Tabella 3: 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 4: Tabella di Karnaugh per $D_1 = E \cdot Q_0$

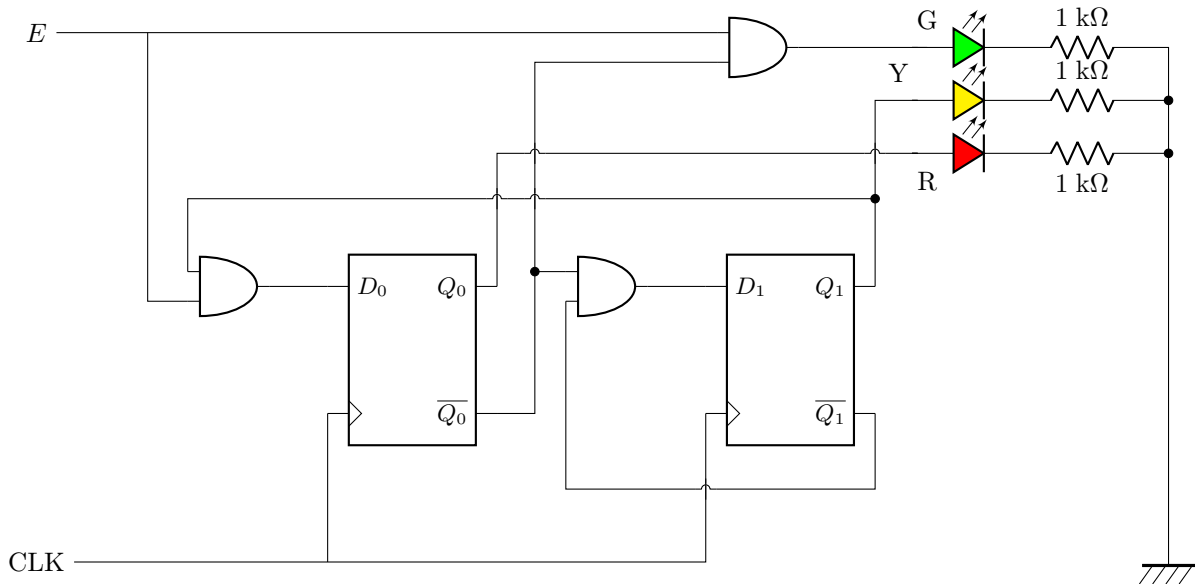


Figura 1: schema del semaforo Mealy con enable.

Per studiarne il comportamento generiamo nei due pin DIO 0 (CLOCK) e DIO 1 (ENABLE) dell'AD2 due segnali di clock di frequenza $f = 20\text{ Hz}$ e 2 Hz agli ingressi CLK ed E del circuito.

2.f Analisi e verifica del funzionamento del circuito

Osserviamo che nel caso $E = 0$, $Q0 = 1$ e $Q1 = 0$ che è riportato come X nella tabella, produce l'output "rosso". Questo vale a dire che se il semaforo è "rosso" con $E = 1$ 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.



Figura 2: Acquisizione di un ciclo completo (frequenza 1 Hz) con Logic Analyzer dei segnali in ingresso e in uscita dal semaforo.

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); a questo scopo utilizzeremo come componenti uno switch, 3 led (rosso, verde e giallo) e una resistenza da $330\ \Omega$ 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 si presenta un fronte di discesa secondo il modello di Moore e secondo quello di Mealy.

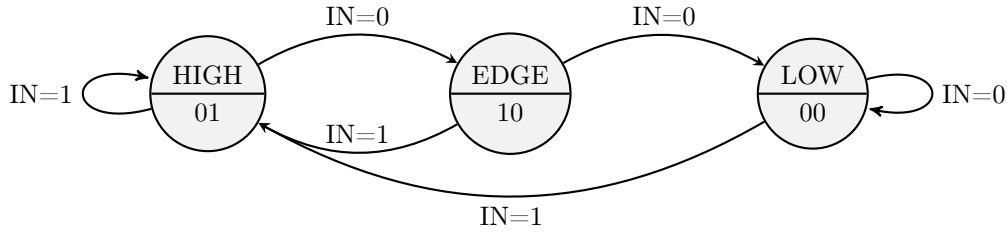


Figura 3: Edge detector FSM di Moore

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 5: codifica binaria degli stati del detector di Moore.

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

Tabella 6: 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 7: 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 8: Tabella di Karnaugh per $\text{OUT} = Q_1$

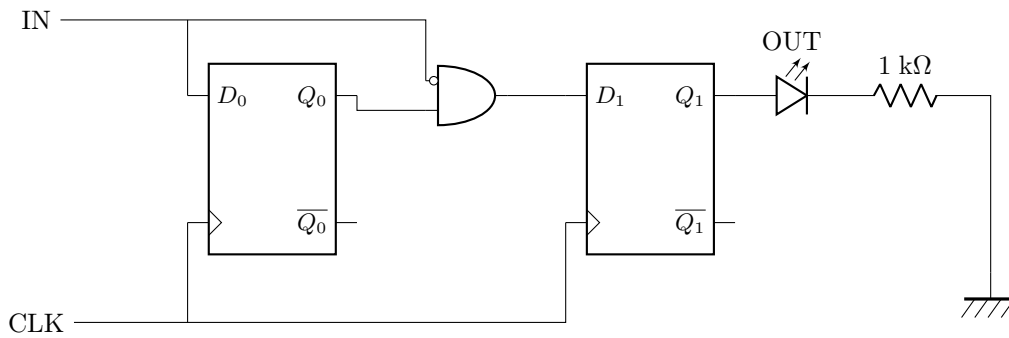


Figura 4: schema del detector Moore.

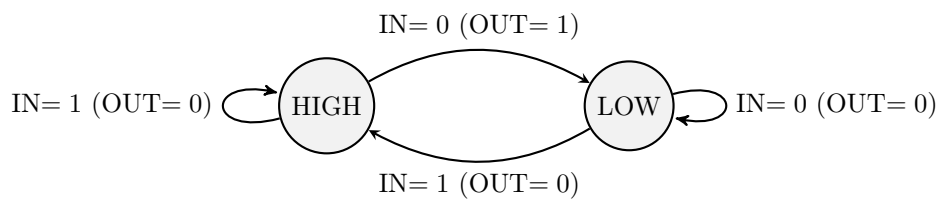


Figura 5: Edge detector FSM di Mealy

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

Tabella 9: codifica binaria degli stati del detector di Mealy. $D = \text{IN}$; $\text{OUT} = \overline{\text{IN}} \cdot Q$

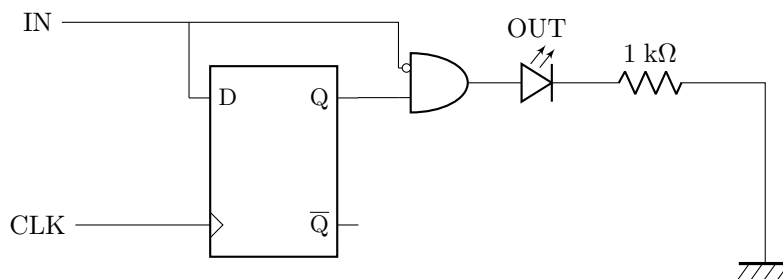


Figura 6: schema del detector Mealy.

5.b Definizione dello stream di bit casuali in ingresso

Con la funzione Patterns di Waveform si invia un segnale di clock di frequenza $f_{clk} = 1 \text{ kHz}$ al pin (CLK) dei Flip-Flop e si acquisiscono i segnali in uscita con la funzione Logic dello stesso.

5.c Verifica del funzionamento e analisi della temporizzazione

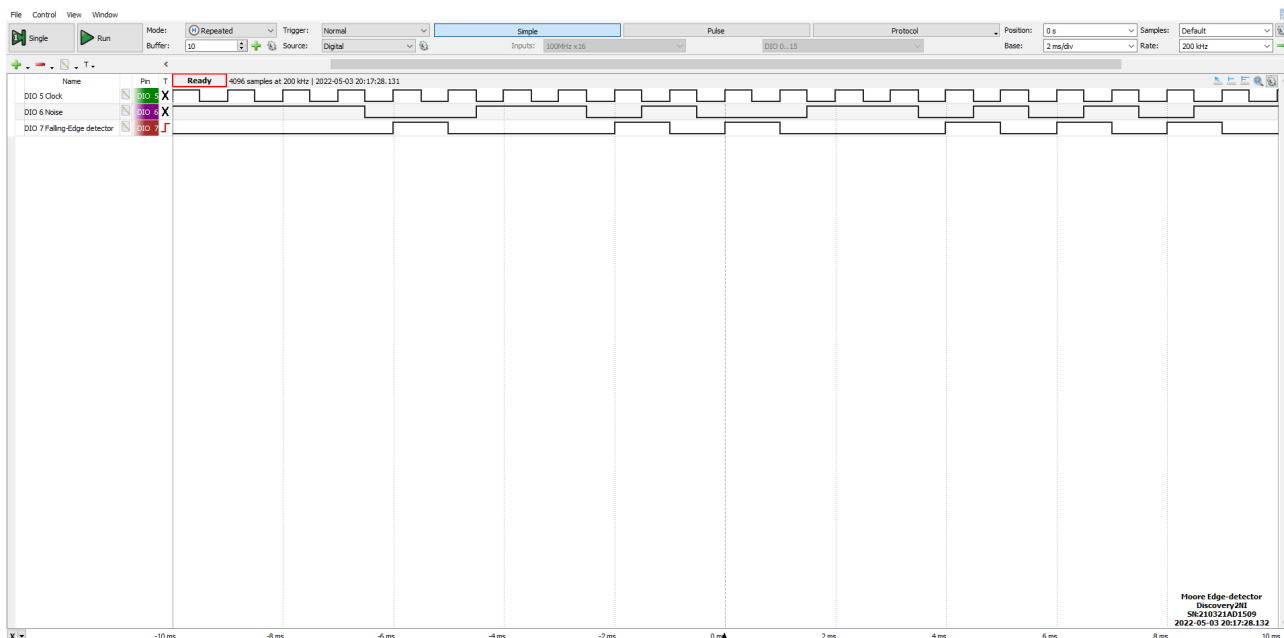


Figura 7: Acquisizione di un ciclo completo (frequenza 1 kHz) con Logic Analyzer dei segnali in ingresso e in uscita dall'edge detector.

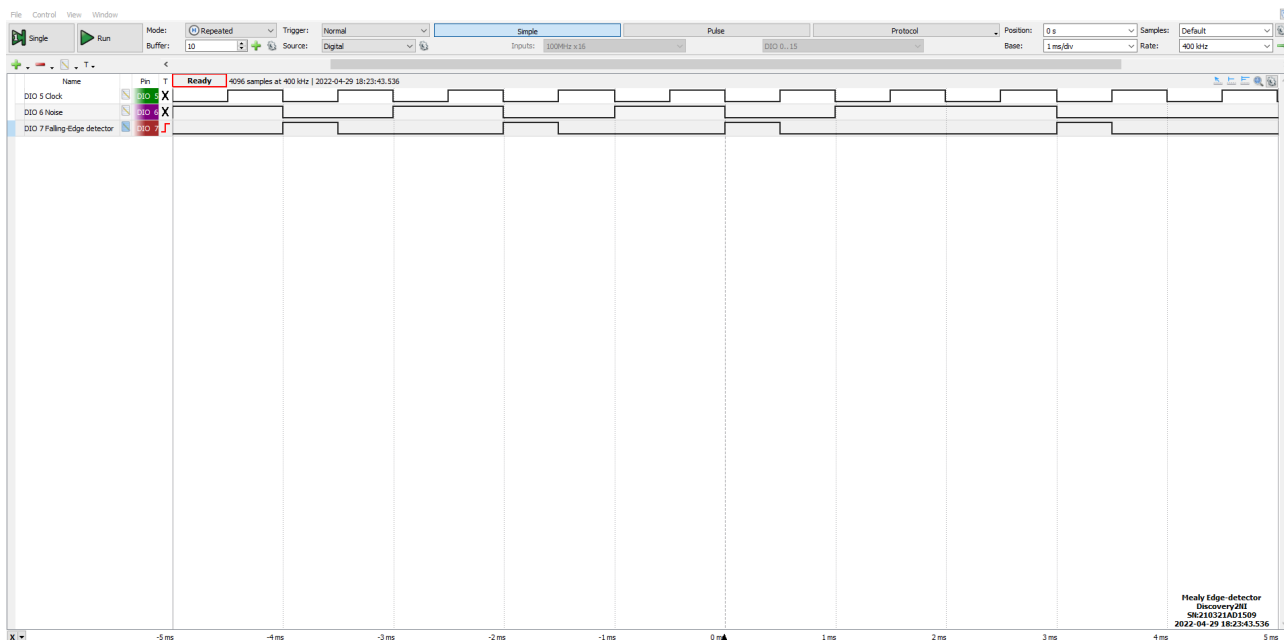


Figura 8: Acquisizione di un ciclo completo (frequenza 1 kHz) con Logic Analyzer dei segnali in ingresso e in uscita dal semaforo.

Conclusioni e commenti finali

Si è riusciti a verificare il corretto funzionamento di circuiti logici sequenziali di crescente 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 l'effetto dei tempi di propagazione delle porte sul loro comportamento, seppur in maniera limitata dalla 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.