

# 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

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

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

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.

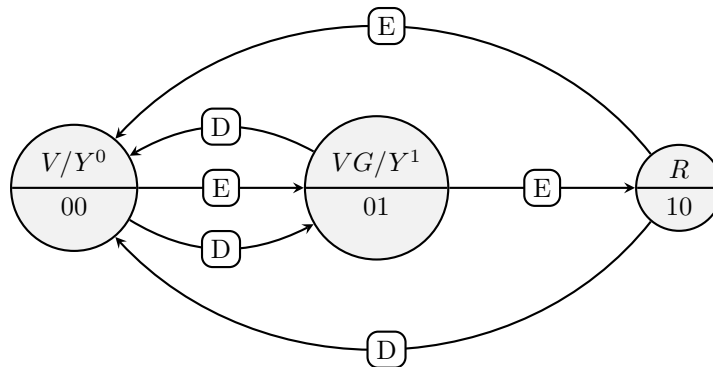


Figura 1: Diagramma degli stati del semaforo FSM di Mealy con Enable

### 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		
		$Q_1$	$Q_0$	$D_1$	$D_0$	
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 ( $V$ )
- $Q_1 \rightarrow$  led rosso acceso ( $R$ )

**AND-gate per led rosso** Impiegando una quarta porta AND è possibile rendere l'accensione del led rosso direttamente dipendente dal segnale di ENABLE (ovvero  $R = Q_1 \cdot E$ ). Dalla tabella 2 infatti si nota che l'unica differenza di funzionamento si avrebbe durante la transizione di  $E : 1 \rightarrow 0$ ; in questo caso, lo spegnimento diventerebbe asincrono rispetto al 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. 2 e si sono collegati i pin **preset** e **clear** dei D-FF a  $V_{CC}$  onde evitare reset o clear spurii.

$E \backslash Q_1 Q_0$	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  $D_1 = E \cdot Q_0$

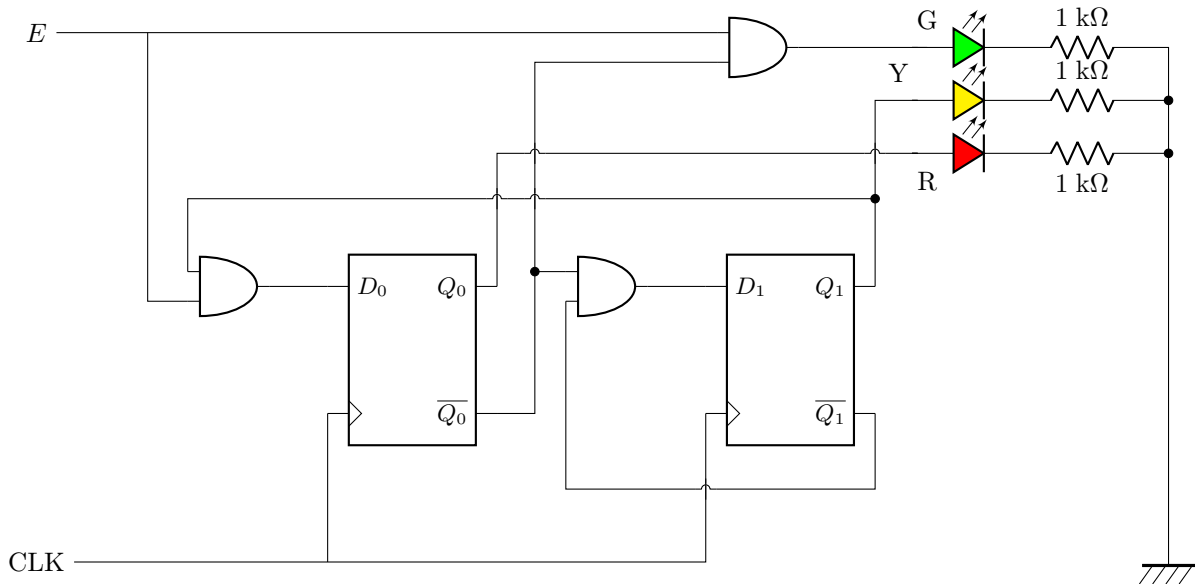


Figura 2: schema del semaforo Mealy con enable.

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  $2\text{ Hz}$  agli ingressi CLK ed  $E$  del circuito.

## 2.f Analisi e verifica del funzionamento del circuito

Osserviamo che nel caso  $E = 0$ ,  $Q_0 = 1$  e  $Q_1 = 0$  che è riportato come X nella tabella, 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.



Figura 3: 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.

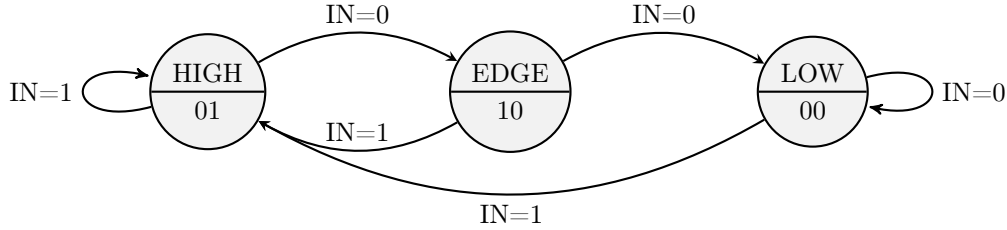


Figura 4: Edge detector FSM di Moore

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 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  $\text{OUT} = Q_1$

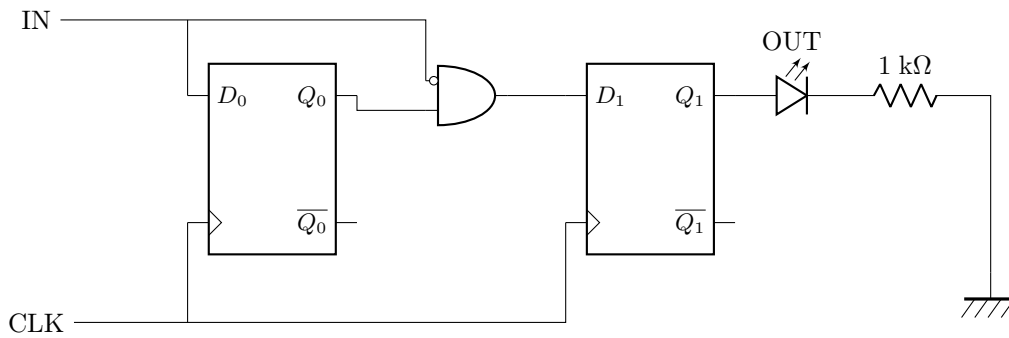


Figura 5: schema del detector Moore.

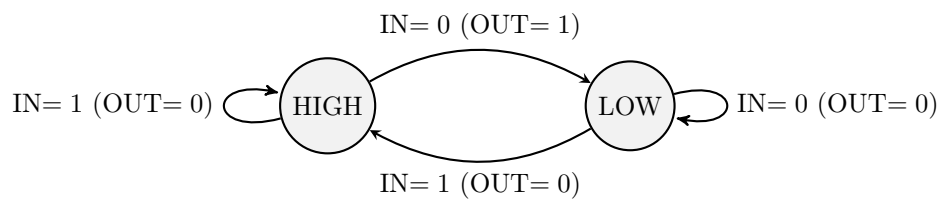


Figura 6: Edge detector FSM di Mealy

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

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

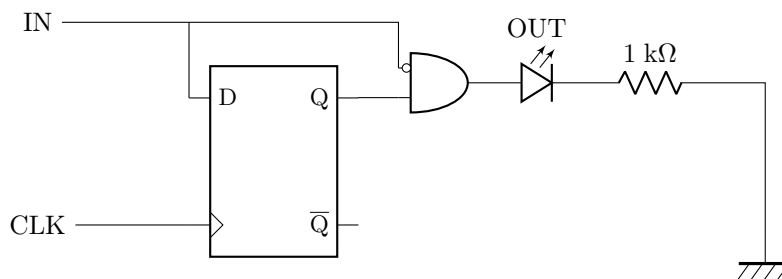


Figura 7: schema del detector Mealy.

## 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

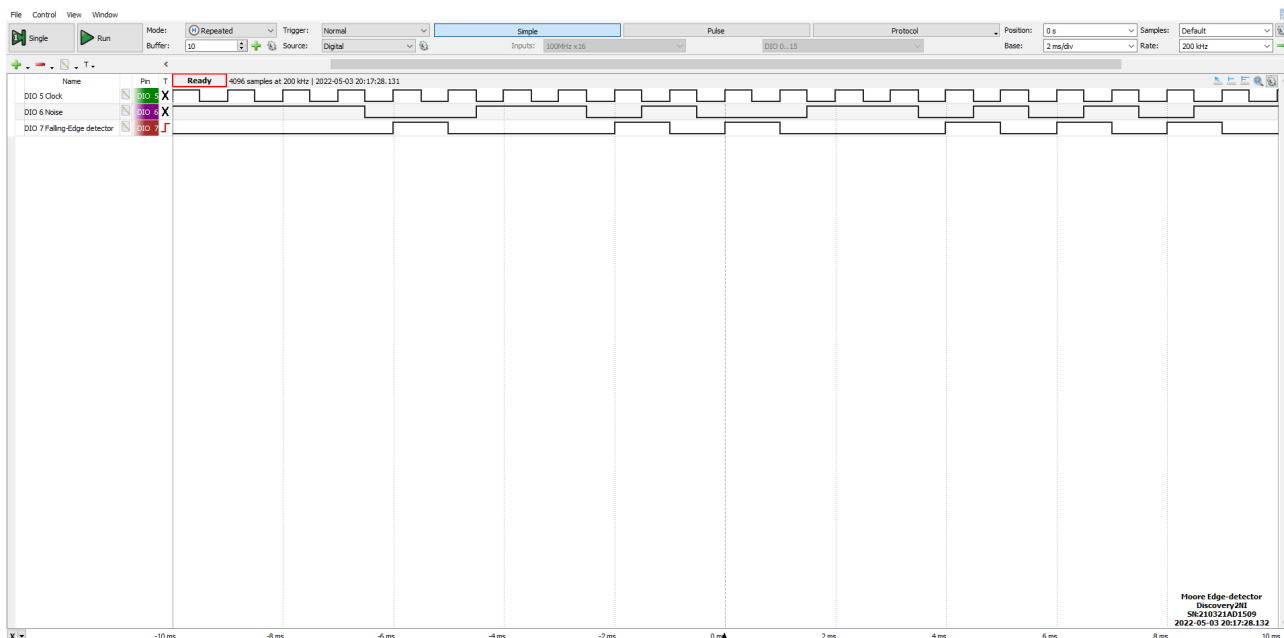


Figura 8: Acquisizione di un ciclo completo (frequenza 1 kHz) con Logic Analyzer dei segnali in ingresso e in uscita dall'edge detector di Mealy.

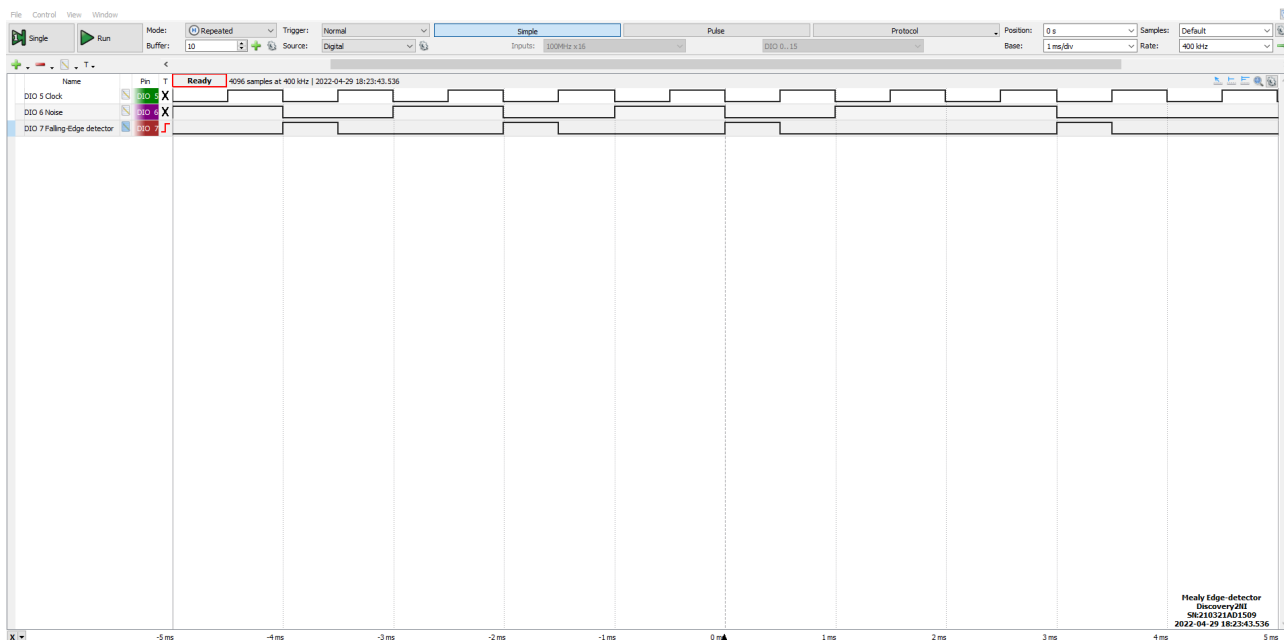


Figura 9: Acquisizione di un ciclo completo (frequenza 1 kHz) con Logic Analyzer dei segnali in ingresso e in uscita dal detector di Moore.

## 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.