



Es12B. Macchine a stati finiti: semaforo.

Gruppo 1G.BS

Riccardo Riolo

Francesco Zazzu

5 marzo 2020

Scopo e strumentazione

Lo scopo dell'esercitazione è la progettazione di un semaforo come applicazione del concetto di macchina a stati finiti (FSM).

Il materiale a disposizione è: un IC SN7400 (quattro porte NAND a due ingressi); due IC SN74LS74 (due FF tipo *D*); un IC 7408 (quattro porte AND a due ingressi); *switch* a 4 interruttori; tre LED, verde, giallo, rosso.

A causa dell'impossibilità di recarsi ai laboratori didattici si discute l'esperienza con l'ausilio di simulazioni al calcolatore dei circuiti in esame.

Si è scelto di implementare il semaforo anche nella modalità disabilitato, dato l'esiguo numero di modifiche al circuito base.

Progettazione della macchina a stati finiti

La sequenza degli stati nella modalità abilitato è ripetuta ciclicamente: LED verde acceso \rightarrow LED verde e giallo accesi \rightarrow LED rosso acceso. In modalità disabilitato oscilla tra lo stato LED giallo acceso e spento. Ogni stato dura un periodo di *clock*.

Preliminarmente si è disegnato il diagramma di stato della FSM con le transizioni relative alle due modalità di utilizzo (i *don't care* sono aggiunti *a posteriori*). Lo si riporta in figura 1. La codifica in bit è $E = 1$ per il semaforo abilitato mentre $E = 0$ disabilitato. Nel primo caso gli stati Q_0Q_1 sono: 00 per il verde; 01 per il verde-giallo; 10 per il rosso; 11 è indeterminato. Nel secondo caso 01 è il giallo acceso e 01,10 sono giallo spento. Si implementa una macchina di Mealy poiché le uscite dipendono oltre che dallo stato attuale anche dell'ingresso E .

Si è proceduto a compilare le tabelle di verità 1 dove $Q_{0n}Q_{1n}$ è lo stato attuale, $Q_{0,n+1}Q_{1,n+1}$ è il successivo e R, G, V sono rispettivamente rosso, giallo, verde accesi. L'esiguo numero di variabili rende evidenti le relazioni:

$$Q_{0,n+1} = E \cdot Q_{1n}, \quad Q_{1,n+1} = \overline{Q_{0n}} \cdot \overline{Q_{1n}},$$

$$R = E \cdot Q_{0n}, \quad G = Q_{1n}, \quad V = E \cdot \overline{Q_{0n}}.$$

Le scelte di codifica hanno come *don't care* lo stato 11 che dalle relazioni sopra ha come stato successivo 00 nel caso $E = 0$ e 10 nel caso $E = 1$; non presenta perciò problemi di sorta perché se l'implementazione si trova per caso in 11 prosegue il corso normale al successivo periodo di *clock*.

Circuito

Si è implementato il circuito con l'uso del programma di simulazione *System Vision*, lo schema circuitale è riportato in figura 3. Gli stati sono implementati in un registro composto da due FF di tipo D: 'ff1' per Q_0 e 'ff2' per Q_1 . La simulazione comincia in entrambi i casi $E = 0$ ed $E = 1$ nello stato 00. Le uscite sono mostrate nelle figure 5 e 4.

E	Q_{0n}	Q_{1n}	$Q_{0,n+1}$	$Q_{1,n+1}$	V	G	R
1	0	0	0	1	1	0	0
1	0	1	1	0	1	1	0
1	1	0	0	0	0	0	1
1	1	1	(1)	(0)	(0)	(1)	(1)
0	0	0	0	1	0	0	0
0	0	1	0	0	0	1	0
0	1	0	0	0	0	0	0
0	1	1	(0)	(0)	(0)	(1)	(0)

Tabella 1: Tabella di verità per la modalità abilitato ($E = 1$) e per la modalità disabilitato ($E = 0$). I *don't care* sono riportati tra parentesi.

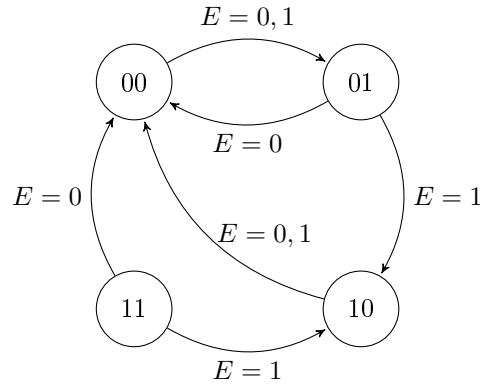


Figura 1: Diagramma degli stati.

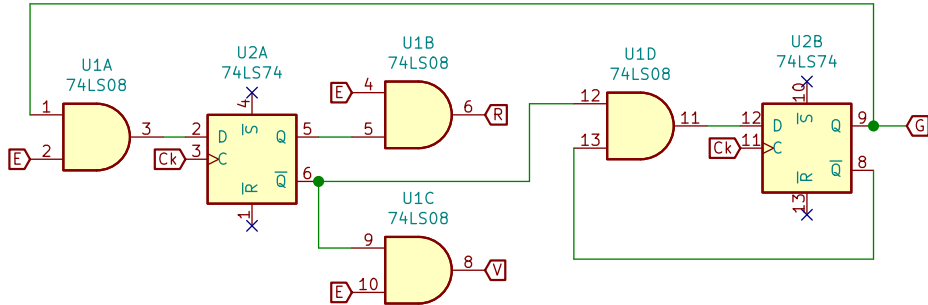


Figura 2: Diagramma del circuito.

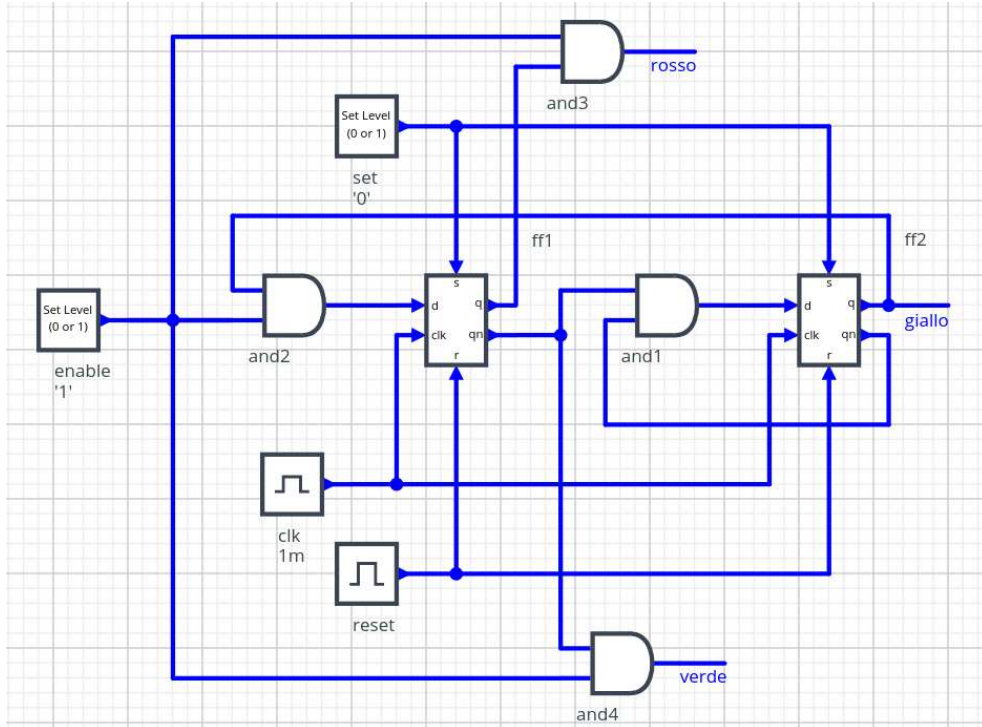


Figura 3: Circuito nel programma di simulazione.

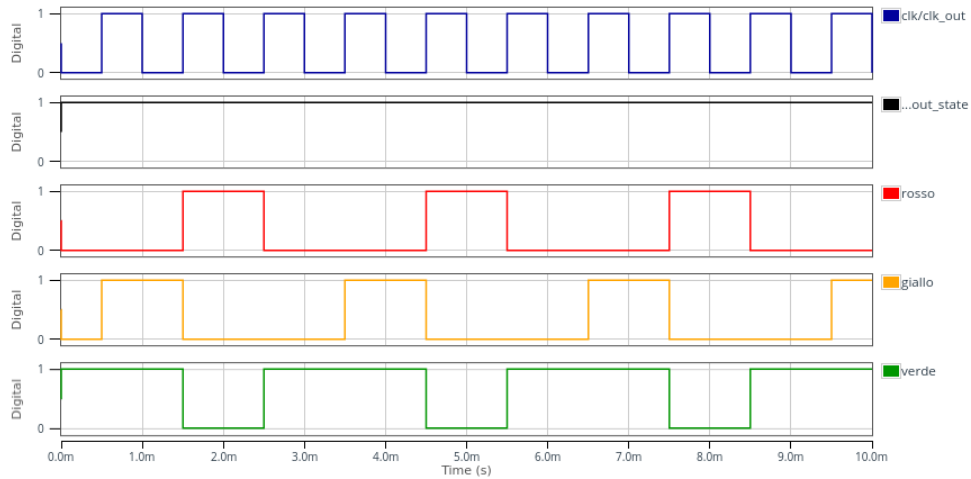


Figura 4: Simulazione in modalità abilitato, gli stati perdurano per un impulso di *clock*. I segnali sono, dall'alto: *clock*, *E*, *R*, *G*, *V*.

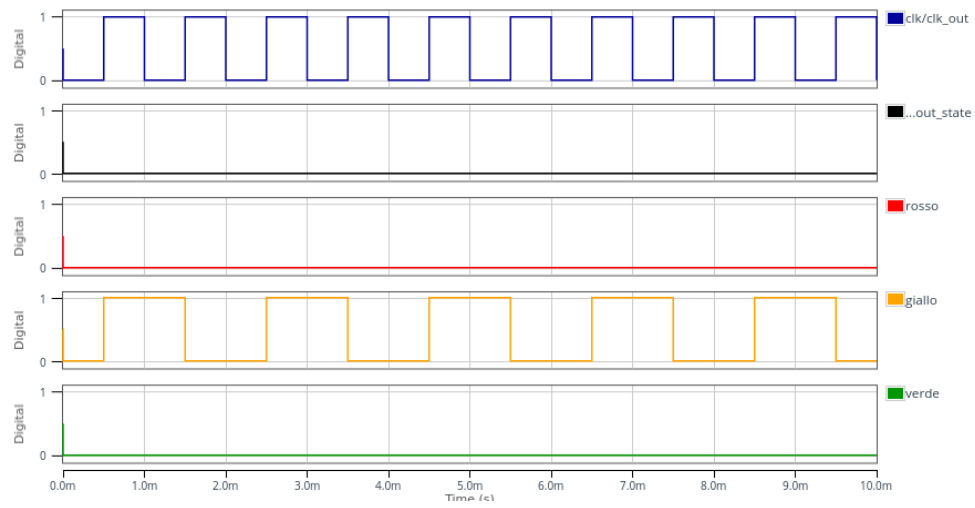


Figura 5: Simulazione in modalità disabilitato, gli stati perdurano per un impulso di *clock*. I segnali sono, dall'alto: *clock*, *E*, *R*, *G*, *V*.