Semaforo

Gruppo AA
Luca Ciambriello, Gianfranco Cordella, Leonardo Bertini

4 Maggio 2017

1 Scopo e strumentazione

L'esperienza ha come obiettivo la realizzazione di un semaforo utilizzando il concetto di macchina a stati finiti. Abbiamo utilizzato il segnale TTL in uscita dalla boccola PULSE del generatore di funzioni come clock, con un duty cycle del 50% e una tensione compresa tra 0 e 5 V. Per implementare la logica necessaria abbiamo impiegato un integrato 7400 costituito da 4 porte NAND, un integrato 7408 formato da 4 AND e un integrato 7432 contenente 4 porte OR. Due Flip-Flop di tipo D (integrato 7474) completano l'apparato strumentale, insieme a 4 resistenze di valori $R_1 = 320 \pm 3\ \Omega$, $R_2 = 329 \pm 3\ \Omega$, $R_3 = 321 \pm 3\ \Omega$, $R_4 = 325 \pm 3\ \Omega$, misurati con il tester digitale, e a tre led rispettivamente verde, giallo e rosso. Per realizzare l'ENABLE che distingua tra i due stati ABILITATO/DISABILITATO del semaforo è stato utilizzato anche uno switch a 4 bit. Si è infine fatto uso dell'oscilloscopio per osservare le forme d'onda.

2 Semaforo nello stato abilitato

Nello stato abilitato il semaforo passa dallo stato in cui è acceso il solo led verde a quello in cui sono accesi i led verde e giallo e quindi a quello in cui è acceso il solo led rosso, per poi ritornare nello stato iniziale; un diagramma di transizione è mostrato in figura (8), considerando soltanto ENABLE 0.
Abbiamo utilizzato una logica a 2 bit definendo come 00 lo stato in cui è acceso il solo led verde, 01 quello in cui sono accesi i led verde e giallo, 10 quello in cui è acceso il solo led rosso. In tabella (1) è mostrata la tabella di verità del circuito.

Per ottenere ${b_0}^{n+1}$ abbiamo utilizzato un AND avente come ingressi i negati di ${b_0}^n$ e ${b_1}^n$, ottenuti

${b_1}^n$	${b_0}^n$	${b_1}^{n+1}$	${b_0}^{n+1}$	VERDE	GIALLO	ROSSO
0	0	0	1	1	0	0
0	1	1	0	1	1	0
1	0	0	0	0	0	1
1	1	1	0	x	x	x

Tabella 1: Tabella di verità osservata per il semaforo nello stato abilitato

in corrispondenza delle uscite $\bar{Q}$ dei flip-flop (uno corrispondente al bit b_0, l'altro al bit b_1), mentre agli ingressi D vanno lo stesso ${b_0}^{n+1}$ e ${b_1}^{n+1}$, quest'ultimo pari, come evidente dalla tabella di verità, a ${b_0}^n$ (ottenuto all'uscita Q del flip-flop corrispondente al bit b_0). I due flip-flop hanno lo stesso segnale di clock, posto ad una frequenza di circa 1 Hz. L'uscita VERDE è stata ottenuta negando ${b_1}^n$, quindi corrisponde a $\bar{Q}$ del flip-flop per b_1. L'uscita GIALLO corrisponde a Q del flip-flop per b_0, mentre l'uscita ROSSO può essere considerata l'AND di GIALLO negato (quindi $\bar{b_0}^n$) e VERDE negato (quindi ${b_1}^n$). Le tre uscite VERDE, GIALLO e ROSSO vengono mandate rispettivamente alle resistenze R_2, R_3 e R_4, poste in serie ai rispettivi led. Lo schema circuitale è mostrato in figura (2).
L'ultima riga della tabella di verità può determinare valori arbitrari in uscita ai LED in quanto l'ingresso 11 non rientra nei tre stati che si ripetono ciclicamnete (infatti non corrisponde ad alcuno stato). Tuttavia noi abbiamo posto per VERDE, GIALLO e ROSSO rispettivamente 0,1,0. I valori scelti sono

tesi a semplificare la logica.
Infine, abbiamo collegato alla tensione di alimentazione, di poco inferiore a 5 V, gli ingressi di preset e clear dei flip-flop onde evitare fenomeni spuri. Le funzioni combinatorie utilizzate sono quelle mostrate, ponendo EN=0, nelle equazioni mostrate nella sezione successiva.
Dopo aver verificato il funzionamento del circuito a bassa frequenza, ci siamo posti ad una frequenza di circa 10 Hz osservando all'oscilloscopio le forme d'onda corrispondenti:

- al segnale di clock confrontato con il segnale passante dal led verde; si può osservare che il periodo del secondo è il triplo del primo e che il duty cycle è pari a 2/3 (figura (4)).
- al segnale di clock confrontato con il segnale passante dal led giallo; anche in tal caso il periodo del secondo è il triplo del primo, ma il duty cycle è pari a 1/3 (figura (5)).
- al segnale passante dal GIALLO confrontato con quello passante dal ROSSO; i due segnali hanno uguale periodo e duty cycle di 1/3, ma presentano uno sfasamento tale che quando il GIALLO scende il ROSSO, precedentemente basso, sale (figura (6)).

3 Semaforo completo

Abbiamo realizzato il circuito mostrato in figura (1), facendo uso di 4 AND, 2 OR e 3 NAND per costruire la logica (in figura (8) è mostrato il diagramma di transizione). L'implementazione è stata effettuata con una macchina di Mealy, mentre il semaforo risulta abilitato ad ENABLE basso. In tabella (2) è mostrata la tabella di verità. L'ENABLE è stato realizzato come mostrato in figura (7), dove la resistenza impiegata R_1 ha la funzione di limitare un eccessivo passaggio di corrente tra il potenziale Vcc e la massa.
Ad ENABLE alto si ha il semaforo lampeggiante, dove abbiamo i tre led spenti per lo stato 10 e solo

EN	b_1^n	b_0^n	b_1^{n+1}	b_0^{n+1}	V	G	R
0	0	0	0	1	1	0	0
0	0	1	1	0	1	1	0
0	1	0	0	0	0	0	1
0	1	1	1	0	x	x	x
1	0	0	0	1	x	x	x
1	0	1	1	0	x	x	x
1	1	0	1	1	0	0	0
1	1	1	1	0	0	1	0

Tabella 2: Tabella di verità del semaforo; ai don't care, che compaiono per gli stati proibiti, abbiamo assegnato valori che fossero in accordo con le espressioni delle funzioni logiche

il giallo acceso per lo stato 11. Dal momento che 11 va in 10 e viceversa,il circuito oscilla tra questi due stati; anche se inizialmente lo stato è 01 o in 00 (stati proibiti, con uscite ai LED arbitrarie) si giunge in questi due stati.
Abbiamo verificato il funzionamento del circuito con frequenza di clock di 1 Hz, rilevando all'oscilloscopio, con frequenza di 10 Hz, la forma d'onda dell'uscita corrispondente al giallo, ad ENABLE alto (semaforo DISABILITATO), a confronto con la forma d'onda del clock (figura (3)). Si può notare che il segnale all'ingresso del LED giallo ha periodo doppio del clock e duty cycle del 50 %
Le funzioni logiche che verificano la tabella di verità, opportunamente semplificate ed implementate nel circuito, sono:

$$b_0{}^{n+1} = (\bar{b_0}{}^n b_1{}^n)EN + (\bar{b_0}{}^n \bar{b_1}{}^n) \tag{1}$$

$$b_1{}^{n+1} = (\bar{b_0}{}^n b_1{}^n)EN + b_0{}^n \tag{2}$$

$$GIALLO = b_0{}^n \tag{3}$$

$$VERDE = \bar{b_1}{}^n \tag{4}$$

$$ROSSO = (b_1{}^n \bar{b_0}{}^n)E\bar{N} \tag{5}$$

Si può notare che si tratta di una macchina di Mealy in quanto gli output, in particolare ROSSO, sono funzione sia dello statao che dell'input di Enable.

4 Macchina a stati finiti con arduino

E' stato realizzato lo stesso semaforo tramite arduino. Sono state stabilite come porte di output le porte D9, D10, D11 che sono state poste all'ingresso dei diodi rispettivamente verde, giallo e rosso. Le uscite sono state prelevate non direttamente da arduino ma dal buffer 74LS244 ad esso connesso. Inoltre in serie ai diodi sono state lasciate le resistenze di limitazione della corrente, utilizzate anche in precedenza.
E' stato stabilita come input-pullup la porta D8, che è stata collegata tramite l'interruttore a massa. Le condizioni di interruttore in cortocircuito oppure aperto sono interpretate come stati LOW e HIGH da arduinio.
Sono stati utilizzati cinque stati corrispondenti a quali dei tre led risultano accesi. A ciascuno stato è assegnata una variabile intera: tabella (3). La logica della macchina a stati finiti implementata via

STATO	intero
V	0
VG	1
R	2
OFF	3
G	4

Tabella 3: Stati di ARDUINO

software si trova nella tabella (4). Per ciascuna transizione tra stati diversi è stabilito un tempo di transizione che può essere eventualmente modificato indipendentemente per ciascuna transizione.

EN	$STATO^n$	$STATO^{n+1}$
T	V	VG
F	V	OFF
T	VG	R
F	VG	OFF
T	R	V
F	R	OFF
T	OFF	R
F	OFF	G
T	G	R
F	G	OFF

Tabella 4: Tabella di verità del semaforo con ARDUINO. I valori logici booleani dell'enable possono essere True o False, in base al potenziale letto dalla porta D8

La macchina qui realizzata è del tipo Moore in quanto, i valori dei tre output (D9, D10, D11) all'ingresso dei LED sono determinati solamente dallo stato della macchina e non da altri input.

5 Conclusioni

E' stato realizzato un semaforo dapprima funzionante solo in modalità abilitata. In seguito è stato implementato un sistema he permettesse di decidere la modalità abilitata o disabilitata tramite l'azione

di un interruttore. In entrambi i casi si è cercato il modo più efficiente e con meno porte logiche e bit di realizzare il semaforo.

6 Grafici ed immagini

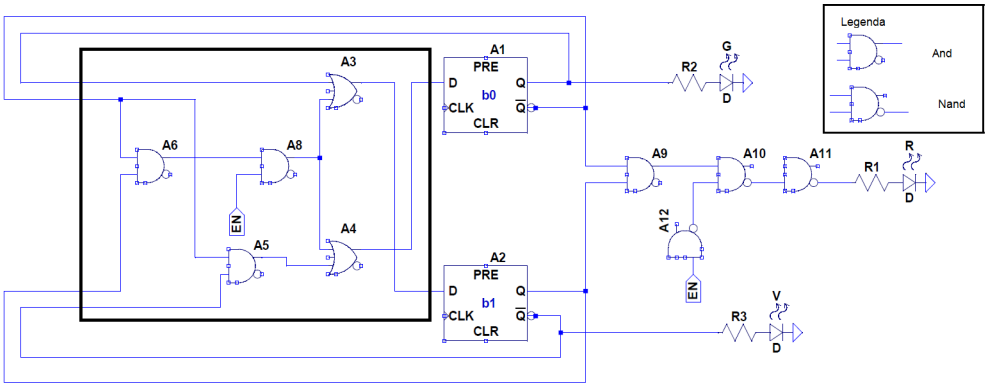

Figura 1: Schema del circuito del semaforo completo.

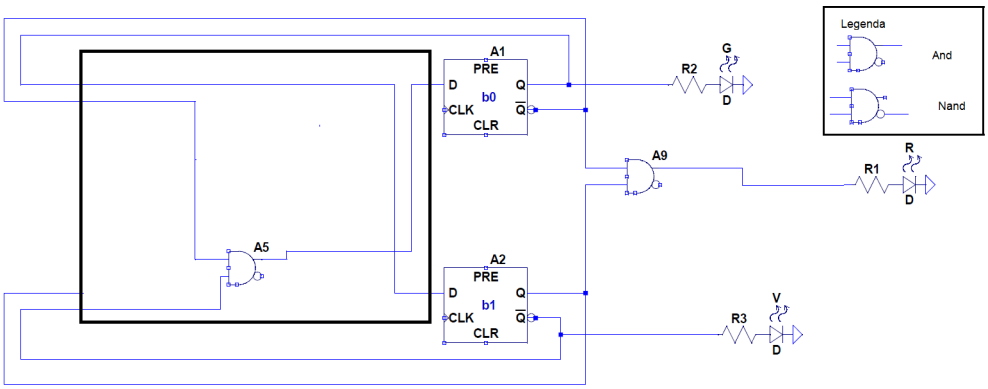

Figura 2: Schema del circuito del semaforo abilitato.

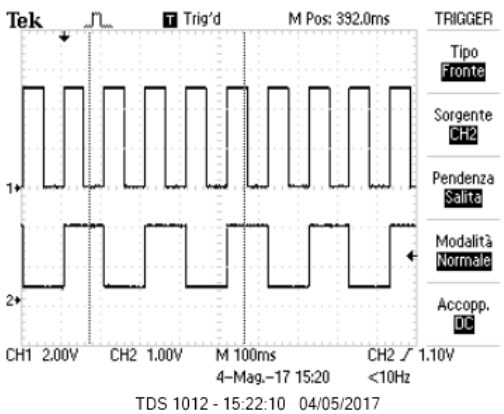

Figura 3: Il segnale in alto è il clock mentre quello in basso è il giallo lampeggiante.

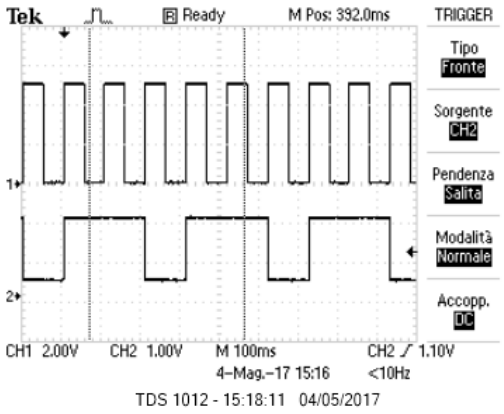

Figura 4: Il segnale in alto è il clock mentre quello in basso è il verde nel caso di semaforo abilitato.

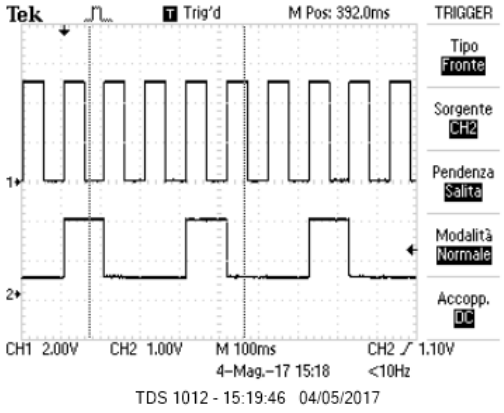

Figura 5: Il segnale in alto è il clock mentre quello in basso è il giallo nel caso di semaforo abilitato.

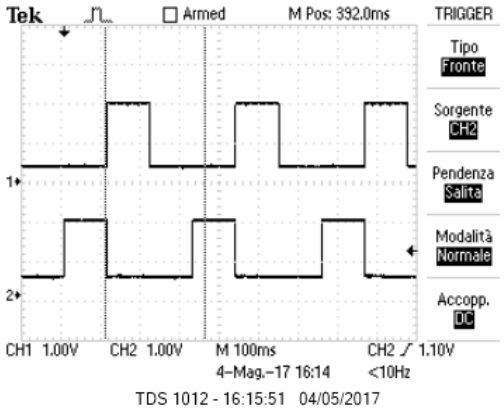

Figura 6: Il segnale in alto è il rosso mentre quello in basso è il giallo,entrambi nel caso di semaforo abilitato.

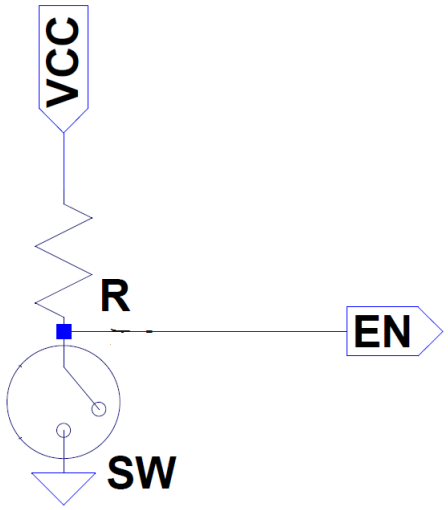

Figura 7: Schema circuitale usato per l'ENABLE.

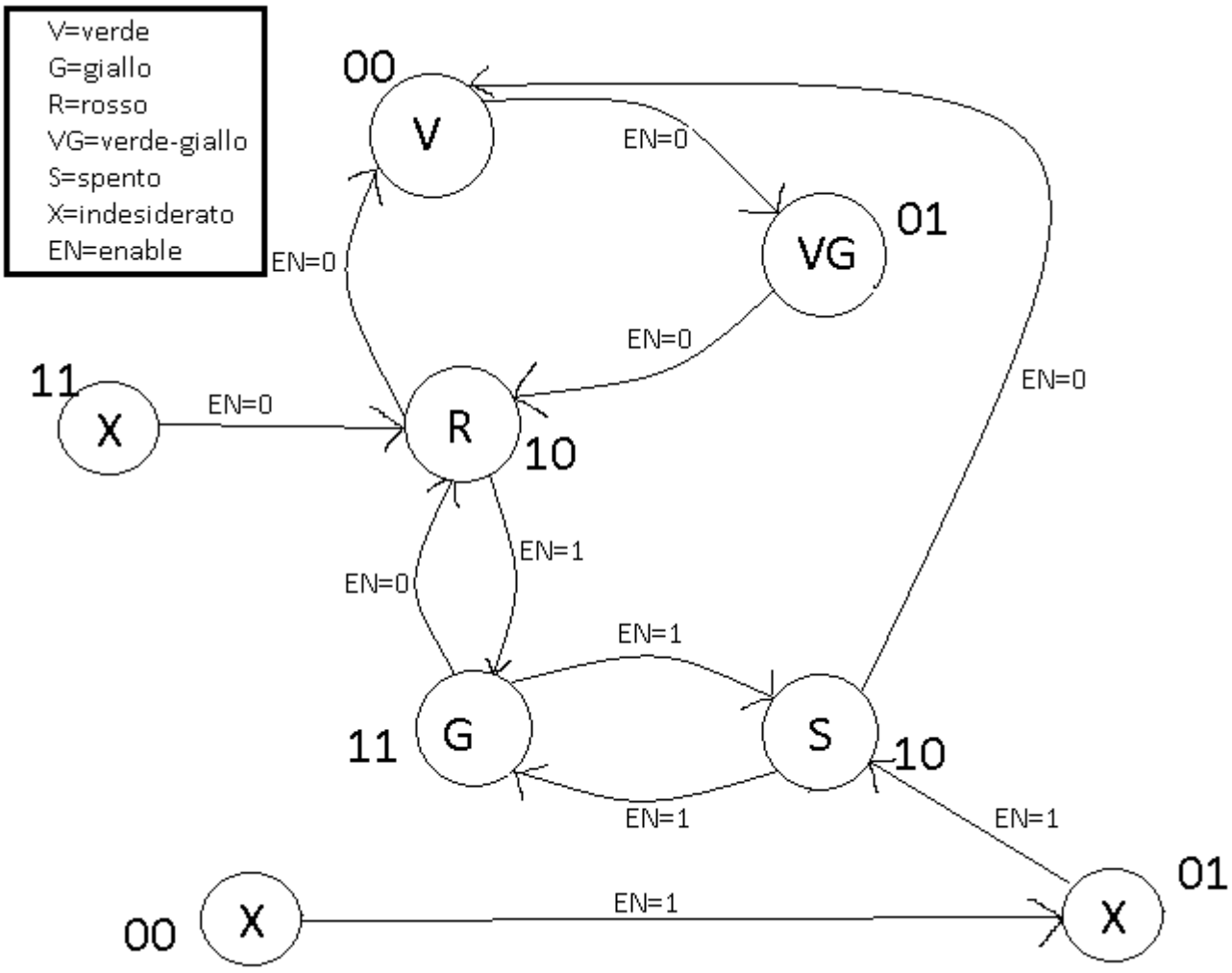

Figura 8: Diagramma di stato per il semaforo completo.