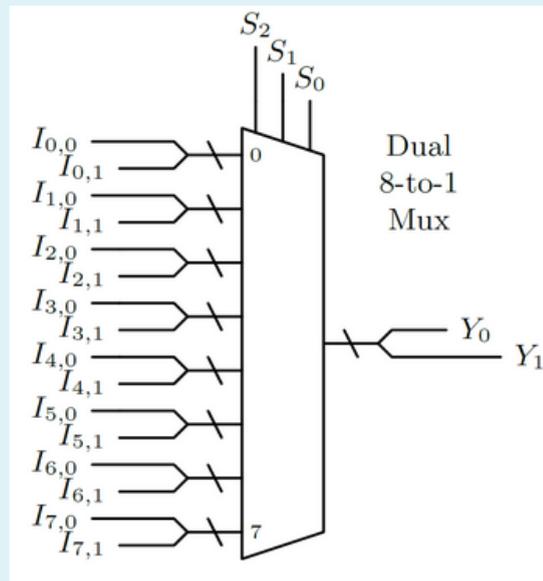


SVOLGERE I SEGUENTI ESERCIZI SU FOGLIO DI CARTA CHE SCANSIONERETE ALLA FINE DEL TEST

ESERCIZIO 1



Si desidera realizzare la funzione logica a 4 ingressi (a, b, c, d) e 2 uscite (Y_0, Y_1) tramite il dual-mux 8-to-1 rappresentato in figura, avente 3 ingressi di selezione (S_2, S_1, S_0) e 8 ingressi di dati a 2 bit ciascuno ($I_{0,0}, I_{0,1}, \dots, I_{7,0}, I_{7,1}$). Le funzioni Y_0 ed Y_1 sono definite come segue:

$$Y_0(a,b,c,d) = \sum m (0, 2, 3, 6, 9, 11, 14, 15)$$

$$Y_1(a,b,c,d) = \sum m (0, 1, 3, 4, 7, 8, 11, 13, 15)$$

- a) Si illustri come collegare i segnali (a,b,c,d) al multiplexer al fine di ottenere le funzioni Y_0 ed Y_1 richieste
- b) Si proponga una descrizione in VHDL della funzione $Y_0(a,b,c,d)$

ESERCIZIO 2

Un sistema digitale avente un ingresso X ed un'uscita Z deve riconoscere la sequenza di ingresso "1011". Sono consentite sovrapposizioni tra i bit riconosciuti: l'ultimo '1' di una sequenza corretta riconosciuta deve essere riconosciuto come il primo '1' della sequenza successiva. L'uscita Z deve commutare a '1' appena il sistema riceve all'ingresso l'ultimo bit della sequenza corretta, mentre deve valere '0' in tutti i casi in cui non è stata riconosciuta la sequenza corretta. Usare Flip-Flop di tipo D positive edge triggered.

Svolgere i seguenti passi:

- a) Disegnare il diagramma di stato del sistema
- b) Dire se si tratta di una macchina di Mealy o di Moore, motivando la risposta
- c) Riportare la tabella degli stati e dell'uscita
- d) Codificare gli stati con una codifica Gray
- e) Riportare la tabella degli stati codificata
- f) Minimizzare le equazioni di aggiornamento di stato e di uscita
- g) Disegnare il circuito sequenziale che realizza il sistema descritto

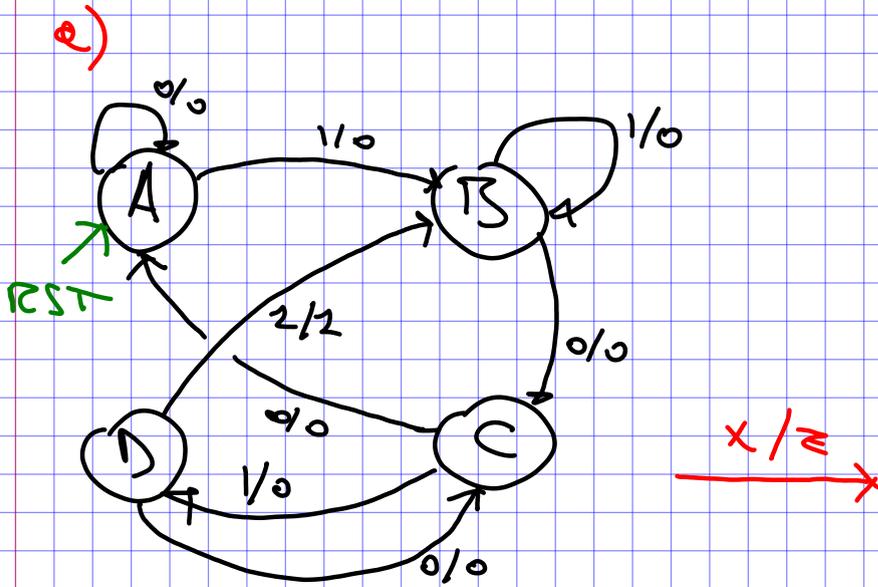
b) Descrizione VHDL di Y_0

Nota: sono possibili più descrizioni
equivalenti; quella riportata è
un esempio

```
1 library IEEE;
2 use IEEE.std_logic_1164.all;
3
4 entity FunctY0 is
5
6     port ( abcd: in std_logic_vector (3 downto 0);
7           y0: out std_logic );
8
9 end FunctY0;
10
11 architecture beh of FunctY0 is
12
13 begin
14
15     y0 <= '1' when abcd = "0000" else
16         '1' when abcd = "0010" else
17         '1' when abcd = "0011" else
18         '1' when abcd = "0110" else
19         '1' when abcd = "1001" else
20         '1' when abcd = "1011" else
21         '1' when abcd = "1110" else
22         '1' when abcd = "1111" else
23         '0';
24
25 end beh;
26
```

ESERCIZIO 2

2 → 0 → 1 → 2



A: No Ric,
 B: ric 1
 C: ric 2 0
 D: ric 2 0 2

b) Ready: uscita \neq dipende da x oltre che dalla stato

c) Tabella di stato:

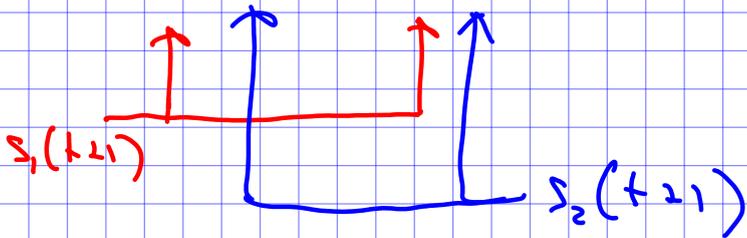
S	S(t+1)		Z	
	x=0	x=1	x=0	x=1
A	A	B	0	0
B	C	B	0	0
C	A	D	0	0
D	C	B	0	2

d) Codice Gray degli stati

	s_1	s_2
A →	0	0
B →	0	1
C →	1	1
D →	1	0

e) Tabella codificata

s_1	s_2	$S(t+1)$		Z	
		$x=0$	$x=1$	$x=0$	$x=1$
0	0	0	0	0	0
0	1	1	0	0	0
1	1	0	0	0	0
1	0	1	0	0	1



1) $S_1(t+1)$:

s_1	0	0	1	1
s_2	0	1	1	0
x				
0		2		2
1			2	

$$S_1(t+1) =$$

$$\frac{1}{s_1} s_2 x + s_1 s_2 x + s_1 \frac{1}{s_2} x$$

$S_2(t+1)$:

s_1	0	0	1	1
s_2	0	1	1	0
x				
0		2		1
1	1	2	2	

$$S_2(t+1) =$$

$$= \frac{1}{s_1} s_2 + s_1 \frac{1}{s_2} + \frac{1}{s_2} x$$

optimal $\frac{1}{s_1} x$

N : $N = s_1 \frac{1}{s_2} x$

g) Circuiti sequenziali:

(linee di retroazione non disegnate esplicitamente)

