

Prova d'esame di
RETI LOGICHE (Prof. E. Torti)
Lauree in Ing. Elettronica e Informatica DM 270
5 settembre 2022

CANDIDATO:

n. matricola:

- 1) Si progetti il sistema di controllo di un montacarichi secondo le seguenti specifiche. Il montacarichi si muove tra due piani (terra e primo), in base alla pressione di due pulsanti A e B. Durante la pressione del pulsante A ($A=1$), il montacarichi deve spostarsi verso il piano terra (qualora non si trovi già al piano terra). Durante la pressione del pulsante B ($B=1$) il montacarichi deve spostarsi verso il primo piano (qualora non si trovi già al primo piano). Se nessuno dei pulsanti viene premuto il montacarichi resta fermo nella posizione in cui si trova. La presenza del montacarichi viene rilevata tramite due sensori C e D ($C=1$ se il montacarichi è al piano terra, $D=1$ se il montacarichi è al primo piano, C e D entrambi a 0 se il montacarichi si trova tra i due piani). Il sistema di controllo presenta tre uscite: Y1, Y2 ed Y3. Y1 dovrà valere uno solo quando il montacarichi deve essere spostato verso il basso, Y2 dovrà valere 1 solo quando il montacarichi deve essere spostato verso l'alto. L'uscita Y3 indica la presenza di un guasto nel sistema, cioè quando i sensori C e D sono entrambi pari ad 1 (in questo caso le uscite Y1 e Y2 sono pari a 0). Se i pulsanti A e B vengono premuti contemporaneamente, il sistema assegna la priorità al pulsante B. Si indichino tabella di verità, equazioni booleane in forma minima e si realizzi il controllore utilizzando un opportuno decodificatore.
- 2) Si descrivano struttura e caratteristiche di una memoria ROM. Data una ROM 32 X 6 con ingresso di abilitazione, si progetti una memoria ROM 96 X 12 utilizzando un numero opportuno di chip ed un decodificatore.
- 3) Si descriva la struttura di un circuito sommatore/sottrattore che, in base ad un bit di selezione S effettua la somma ($S=0$) oppure la sottrazione ($S=1$) di due numeri binari A e B posti in ingresso. Descrivere poi come sia possibile individuare l'overflow.
- 4) Si progetti un circuito in grado di riconoscere la sequenza 101010 all'interno di una sequenza arbitraria di bit. Il circuito dovrà essere dotato di un ingresso X e di un'uscita Z. L'uscita Z dovrà assumere il valore 1 tutte le volte che all'ingresso X sono già stati applicati i valori 10101 e l'ingresso corrente vale 0. In tutti gli altri casi l'uscita Z deve valere 0. Si preveda anche la presenza di un segnale di Reset asincrono che riporti il circuito nello stato iniziale. Si utilizzi la codifica one-hot per la codifica degli stati e si ricavino le equazioni del sistema. Quanto vale il costo degli ingressi del circuito, considerando il costo di ogni singolo flip-flop pari a 14 (ignorare il costo delle porte NOT)?
Esempio di sequenza da riconoscere: X: 0 0 0 1 0 1 0 1 0 1 0 0
uscita Z: 0 0 0 0 0 0 0 0 1 0 1 0
- 5) Si progetti un circuito sequenziale in grado di operare secondo la seguente tabella di stato. Si utilizzino solo flip-flop di tipo D e si calcoli il costo degli ingressi del circuito, considerando il costo di ogni singolo flip-flop pari a 14 (ignorare il costo delle porte NOT)? Come cambierebbero le equazioni ed i costi qualora si adottasse la codifica one-hot? Si indichi inoltre se il circuito è di tipo Mealy o Moore, giustificando la risposta.

Stato Presente	Stato Futuro		Uscita
	X=0	X=1	Z
00	00	01	0
01	10	01	1
10	10	11	1
11	11	00	0

- 6) Si descrivano l'utilità, il funzionamento e le diverse tipologie di ingressi diretti considerando un generico circuito sequenziale.
- 7) Si consideri il seguente codice VHDL che descrive un full adder a 4 bit. Commentare sinteticamente gli statement del codice (riportare sul foglio il numero della riga seguito dal relativo commento). Spiegare poi come andrebbe aggiunta una porta OVF che indichi il verificarsi della condizione di overflow. Modificare il codice in modo da introdurre una porta S che permetta di scegliere tra operazione di somma (S=0) o sottrazione (S=1). Si renda infine l'architettura sincrona aggiungendo un segnale di clock.

```

1: library ieee;
2: use ieee.std_logic_1164.all;
3: use ieee.numeric_std.all;

4: entity adder4 is
5:     port(
6:         A,B : in std_logic_vector(3 downto 0);
7:         S   : out std_logic_vector(3 downto 0)
8:     );

9: end adder4;

10: architecture arch of adder4 is
11:     signal sum : signed(3 downto 0);

12:     begin
13:         sum <= signed(A)+signed(B);
14:         S <= std_logic_vector(sum(3 downto 0));

15: end arch;

```