

Prova d'esame di
RETI LOGICHE (Prof. E. Torti)
Lauree in Ing. Elettronica e Informatica DM 270
15 novembre 2024

CANDIDATO:

n. matricola:

- 1) Si progetti una rete combinatoria che funzioni come codificatore con controllo di parità per una tastiera numerica. La rete dovrà codificare ogni ingresso della tastiera numerica (da D0 a D9) in un codice binario BCD ed aggiungere un bit di parità per controllo degli errori. Dovrà inoltre essere presente un meccanismo di priorità degli ingressi che dia la priorità all'ingresso con peso minore. Si ricavino la tabella di verità, le equazioni di costo minimo e il costo totale del circuito, indicando quale tipo di parità è stato implementato.
- 2) Si progetti una rete combinatoria che consenta di selezionare e codificare il segnale proveniente da uno di tre sensori in un impianto di monitoraggio industriale. A seconda di un segnale di controllo, il sistema selezionerà uno dei sensori che invierà il segnale in un formato binario a 2 bit. Il sistema riceve in ingresso i valori in uscita da 3 Sensori S0, S1 ed S2. Ciascuno di questi sensori produce in uscita un valore espresso su 2 bit. Il sistema è inoltre dotato di 2 ingressi di controllo C0 e C1 che determinano quale sensore è attivo (00 seleziona il sensore S0, 01 seleziona S1 e 10 seleziona S3). Il sistema deve riportare sulle uscite Y0 ed Y1 i valori acquisiti dal sensore. È inoltre presente un'uscita A che deve valere 1 solo quando viene selezionata in ingresso una combinazione non valida. Si indichino la tabella di verità e le equazioni booleane che descrivono il sistema.
- 3) Durante la progettazione di un circuito digitale per un dispositivo PLA o PAL, è fondamentale ridurre la complessità delle funzioni logiche per ottimizzare l'uso delle risorse e migliorare le prestazioni. Si indichino le differenze nella progettazione di una funzione su ciascuno di questi due dispositivi, motivando accuratamente la risposta.
- 4) Un circuito sequenziale va riprogettato per ridurre il costo. Il circuito da riprogettare è descritto dalle seguenti equazioni:

$$\begin{aligned}D_A &= AB + BC + ACX \\D_B &= \bar{A}X + CX + A\bar{B} + A\bar{X} + B\bar{C}\bar{X} \\D_C &= BC\bar{X} + AC\bar{X} + \bar{A}\bar{B}\bar{C}\bar{X} + \bar{A}B\bar{C}X + A\bar{B}\bar{C}X \\Z &= \bar{A}B + A\bar{B}\end{aligned}$$

Dove A, B e C sono i bit di stato (A è il bit di peso più significativo), X è l'ingresso e Z l'uscita e la codifiche 111 non è stata utilizzata. Il circuito deve essere riprogettato in modo da essere di costo minimo, utilizzando solo flip-flop di tipo D. Si calcoli il costo degli ingressi del circuito originale e di quello riprogettato, considerando il costo del singolo flip-flop pari a 14.

- 5) Si progetti un contatore in complemento a 2, su un numero minimo di bit, in grado di contare da -2 a 7. Si considerino le combinazioni non utilizzate come stati di non specificazione. Si progetti il contatore utilizzando solo flip-flop JK. Si disegni il diagramma di stato, indicando anche cosa accadrebbe qualora si presentassero in ingresso condizioni considerate come stati di non specificazione in fase di progetto.
- 6) Si descrivano struttura e caratteristiche delle principali categorie di contatori, evidenziando per ciascuna i principali svantaggi e vantaggi.

- 7) Si consideri il seguente codice VHDL che descrive un sommatore/sottrattore asincrono. Commentare sinteticamente gli statement del codice (riportare sul foglio il numero della riga seguito dal relativo commento). Spiegare poi come dovrebbe essere modificato il codice per realizzare un sommatore/sottrattore sincrono. Descrivere poi come andrebbe modificato il codice per includere una porta OVF che indichi il verificarsi della condizione di overflow. Descrivere poi come andrebbe modificato il codice per includere una porta M che indichi la presenza di un multiplo di 4 nel risultato.

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

4: entity addsub is
5:     port(
6:         A,B : in std_logic_vector(3 downto 0);
7:         sel : in std_logic;
8:         S   : out std_logic_vector(3 downto 0)
9:     );

10: end addsub;

11: architecture arch of addsub is
12:     signal res : signed(3 downto 0);

13:     begin
14:         with sel select
15:             res <= signed(A)+signed(B) when '0',
16:                 signed(A)+signed(not(B))+1 when '1',
17:                 "XXXX" when others;

18:         S <= std_logic_vector(res(3 downto 0));

19:     end arch;
```