

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

CANDIDATO:

n. matricola:

- 1) Si semplifichi la seguente espressione booleana indicando chiaramente identità/proprietà/teoremi utilizzati. Si proponga poi un'implementazione della funzione semplificata andando ad utilizzare un opportuno decodificatore.

$$F = \bar{A}(A + B) + \bar{C} + BC + \overline{(A + B)C} + BC\bar{D} + CD + \overline{A\bar{B}C\bar{D}} + \bar{A}\bar{B}C$$

- 2) Si riportino la tabella di verità, le mappe di Karnaugh e le equazioni di un half adder e di un full adder. Si progetti poi un sottrattore ad n bit utilizzando il full adder descritto precedentemente. Si utilizzi poi la tecnica della contrazione per costruire un opportuno circuito in grado di sottrarre la costante 9 al dato in ingresso, senza considerare il riporto in uscita ed utilizzando il numero minimo di bit.
- 3) Si descriva il funzionamento di un buffer a tre stati. Indicare poi come sia possibile utilizzare questo componente per realizzare un multiplexer 2-1, descrivendone anche il funzionamento.
- 4) Si progetti un circuito logico sequenziale con due ingressi X e Y e un'uscita Z. L'ingresso X riceve una sequenza binaria di lunghezza arbitraria a partire dal bit più significativo. Per indicare che il bit posto nell'ingresso X è l'ultimo della sequenza, l'ingresso Y viene posto pari ad 1. In tutti gli altri casi Y vale 0. L'uscita Z deve valere 1 quando la sequenza binaria completa risulta essere un multiplo di 8, in tutti gli altri casi l'uscita Z deve valere 0. Costruire il diagramma di stato con il formalismo di Mealy, la tabella di stato e realizzare il circuito adottando la codifica Gray. Ricavare le equazioni booleane del circuito sequenziale e il costo degli ingressi, assumendo che ogni flip-flop abbia un costo pari a 14.
- 5) Si progetti un contatore all'indietro modulo 9 utilizzando solo flip-flop di tipo D. Si calcoli poi il costo degli ingressi, considerando pari a 14 il costo di un singolo flip-flop. Come andrebbe modificato il circuito per includere un segnale di ENABLE attivo basso? Questo segnale deve bloccare il conteggio per ENABLE =1, mentre per ENABLE=0 il conteggio deve riprendere dal valore immagazzinato nel contatore. Ricavare inoltre il diagramma di stato andando ad indicare cosa succedrebbe qualora si presentassero in uscita combinazioni considerate come condizioni di non specificazione in fase di progetto.
- 6) Si confrontino funzionamento e caratteristiche dei flip-flop pulse triggered ed edge triggered, evidenziando eventuali problematiche legate all'utilizzo di questi dispositivi.
- 7) Si consideri il seguente codice VHDL che descrive un sommatore/sottrattore sincrono a 4 bit. 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 includere una porta di ENABLE attivo basso. Descrivere poi come andrebbe modificato il codice per includere una porta OVF che indichi il verificarsi della condizione di overflow.

Modificare infine il codice per includere una porta W che permetta di cambiare l'ordine degli operandi quando W=1.

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

4: entity as is
5:     port(
6:         clk, sel, reset in std_logic;
7:         A,B : in std_logic_vector(4 downto 0);
8:         RIS : out std_logic_vector(4 downto 0)
9:     );

10: end sub;

11: architecture arch of as is
12:     signal res : signed(4 downto 0);

13:     begin
14:         process (clk, reset)
15:             begin
16:                 if (reset = '0') then
17:                     res<="00000";
18:                 elsif (rising_edge(clk))
19:                     if (sel = '0') then
20:                         res <= signed(A)+signed(B);
21:                     else
22:                         res <= signed(A)+signed(not(B))+1;
23:                     end if;
24:                 end if;
25:             end process;
26:             RIS <= std_logic_vector(res(4 downto 0));
27: end arch;
```