

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

CANDIDATO:

n. matricola:

- 1) Si progetti un circuito in grado di rilevare un errore all'interno di una cifra BCD. Si proponga un'implementazione a costo minimo del circuito, indicando anche il costo complessivo degli ingressi. Successivamente, implementare il circuito con un opportuno multiplexer e con un opportuno decodificatore.
- 2) Si progetti un circuito per una bilancia elettronica in grado di calcolare automaticamente il peso netto. Si assuma che la tara sia fissa e memorizzata, con un numero opportuno di bit, all'interno del dispositivo. Quando il piatto con il materiale da pesare viene posto sopra alla bilancia, questa deve automaticamente calcolare il peso netto. Si assuma un fondo scala di 110 g con una risoluzione di 1 g ed una tara pari a 12 g.
- 3) Si descriva la struttura base di un blocco logico programmabile, utilizzato all'interno dei dispositivi FPGA, evidenziando in dettaglio quali elementi logici vengono configurati tramite opportuni stream di bit.
- 4) Un circuito sequenziale va riprogettato in modo da ridurre il costo. Il circuito da riprogettare è descritto dalle seguenti equazioni:

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

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

- 5) Si realizzi un circuito in grado di inviare ciclicamente ad un sistema di monitoraggio i segnali acquisiti da 3 diversi sensori, secondo questa sequenza:

$$V_0V_2V_0V_1V_2V_1V_1V_1V_0V_0V_2V_1$$

Il circuito dovrà essere realizzato utilizzando solamente flip-flop JK e componenti combinatori elementari.

- 6) Si descrivano struttura e funzionamento dei due principali tipi di latch. Si modifichi poi il latch SR per includere un segnale di controllo C. Modificare, infine, il circuito per ottenere un latch D.
- 7) Si consideri il seguente codice VHDL che descrive un riconoscitore di sequenza 100. Commentare sinteticamente gli statement del codice (riportare sul foglio il numero della riga seguito dal relativo commento). Spiegare poi come includere un segnale di enable attivo alto.

Modificare il codice per includere un segnale di SET con meno priorità rispetto al RESET per portare il circuito nello stato C. Includere infine un'uscita su 5 bit che conti il numero di sequenze riconosciute dal circuito.

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

3: entity rec is
4:     port(
5:         x,clk,reset : in std_logic;
6:         y : out std_logic
7:     );

8: end rec;

9: architecture arch of rec is

10:     type state is (A, B, C);
11:     signal actual_state, next_state : state;

12:     begin

13:     stare_register : process(clk, reset)
14:     begin
15:         if(reset='1') then
16:             actual_state <= A;
17:         else if(rising_edge(clk)) then
18:             actual_state <= next_state;
19:         end if;
20:     end if;
21: end process;

22:     next_state_proc : process(x, actual_state)
23:     begin
24:         case actual_state is
25:             when A =>
26:                 if(x='1') then
27:                     next_state <= B;
28:                 else
29:                     next_state <= A;
30:                 end if;
31:                 y <= '0';
32:             when B =>
33:                 if(x='1') then
34:                     next_state <= B;
35:                 else
36:                     next_state <= C;
37:                 end if;
38:                 y <= '0';
39:             when C =>
40:                 if(x='0') then
41:                     next_state <= A;
42:                     y <= '1';
43:                 else
44:                     next_state <= B;
45:                     y <= '0';
46:                 end if;
47:             end case;
48:         end process;
49: end arch;
```