

Prova d'esame di
RETI LOGICHE (Prof. E. Torti)
Lauree in Ing. Elettronica e Informatica DM 270
27 giugno 2023

CANDIDATO:

n. matricola:

- 1) Si progetti un sistema automatico di irrigazione che rispetti le seguenti specifiche. Il sistema è dotato di un ingresso di abilitazione A (A=1 sistema abilitato, A=0 sistema non abilitato), un sensore di livello del serbatoio dell'acqua S (S=1 acqua disponibile, S=0 serbatoio vuoto), 5 sensori di umidità U_i con $i=0, \dots, 4$ ($U_i=0$ umidità corretta, $U_i=1$ umidità bassa) 4 sensori di acidità del terreno C_y con $y=0, \dots, 3$ ($C_y=1$ acidità corretta, $C_y=0$ acidità anomala). Il sistema presenta come uscite un irrigatore I (I=1 irrigatore acceso, I=0 irrigatore spento), un segnalatore acustico E (E=0 segnale attivo, E=1 segnale non attivo) ed un indicatore L che indica quando riempire il serbatoio dell'acqua (L=1 serbatoio da riempire, L=0 altrimenti). Se il sistema non è abilitato, allora tutte le uscite non sono attive (irrigatore spento, segnalatore acustico non attivo, serbatoio OK). Se il sistema è abilitato e l'acidità è anomala, I ed L non sono attivi, mentre E è attivo. Se il sistema è abilitato ed acidità ed umidità sono corrette, tutte le uscite sono inattive. Se invece l'umidità è bassa, viene attivato l'irrigatore a patto che il serbatoio abbia acqua disponibile. Se invece il serbatoio non ha acqua disponibile, l'irrigatore non viene attivato e viene indicato di riempire il serbatoio. Si proponga un'implementazione in forma minima del sistema di controllo, calcolando anche il costo degli ingressi.

- 2) Si semplifichi la seguente funzione logica, indicando quali teoremi/proprietà e/o identità sono stati utilizzati. Si implementi poi la funzione semplificata con un opportuno multiplexer di costo minimo.

$$Y = \bar{A} \bar{B} \bar{C} \bar{D} + A \bar{B} \bar{C} + A \bar{B} C \bar{D} + A B D + \bar{A} \bar{B} C \bar{D} + B \bar{C} D + \bar{A}$$

- 3) Si descriva la struttura di un dispositivo FPGA. Si descrivano e confrontino poi le varie tecniche utilizzabili per produrre i bit di configurazione del dispositivo.

- 4) Un circuito sequenziale va riprogettato per ridurre il costo. Il circuito da riprogettare è descritto dalle seguenti equazioni:

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

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 codifica 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 a 3 bit che segue la sequenza di conteggio 4-1-3-7-5 utilizzando solo flip-flop JK. Si ricavino le equazioni di costo minimo e si disegni il diagramma di stato, indicando anche cosa succederebbero qualora si presentassero in ingresso combinazioni considerate come condizioni di non specificazione in fase di progetto.

- 6) Si consideri un registro a caricamento parallelo. Si discuta la tecnica di gating del clock, descrivendo come viene implementata e quali problematiche comporta. Si descriva poi nel dettaglio una soluzione alternativa a tale tecnica.

- 7) Si consideri il seguente codice VHDL che descrive un sommatore/sottrattore sincrono. 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 un segnale di ENABLE attivo basso. Si indichi come andrebbe modificato il codice per aggiungere un'uscita che indichi il verificarsi della condizione di overflow. Infine, includere un ingresso SET con priorità maggiore rispetto all'ingresso di RESET.

```
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, clk, reset : 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:         process (clk, reset)
15:             if (reset='1') then
16:                 res<="0000";
17:             elsif (rising_edge(clk)) then
18:                 if (sel='0') then
19:                     res <= signed(A)+signed(B);
20:                 else
21:                     res<=signed(A)+signed(not(B))+1;
22:                 end if;
23:             end if;
24:         end process;
25:         S <= std_logic_vector(res(3 downto 0));

26: end arch;
```