

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

CANDIDATO:

n. matricola:

- 1) Si descrivano struttura e caratteristiche di una memoria ROM. Data una ROM 64 X 6 con ingresso di abilitazione, si progetti una memoria ROM 192 X 18 utilizzando un numero opportuno di chip ed un decodificatore.
- 2) Si progetti un circuito combinatorio che confronti due numeri A e B di 4 bit ciascuno rappresentati in complemento a 2 per verificare se B è maggiore di A. Il circuito ha una sola uscita L, per cui dovrà essere $L=1$ se $A < B$, $L=0$ se $A \geq B$. Si utilizzi poi la tecnica della contrazione per realizzare un circuito in cui $L=1$ se $A < 4$, si calcoli il costo degli ingressi del circuito realizzato e si determini il ritardo di propagazione totale, assumendo che ciascuna porta logica utilizzata abbia un ritardo di propagazione di 2 ns.
- 3) Si descrivano le diverse tecniche utilizzabili per generare i bit di configurazione di un dispositivo FPGA, evidenziando per ciascuna le caratteristiche e le eventuali problematiche.
- 4) Si progetti il sistema di navigazione di un robot autonomo. Il robot è dotato di due ingressi e due uscite. Gli ingressi sono un segnale E che indica se il robot è attivo o meno ($E=1$ robot attivo, $E=0$ robot non attivo) ed un sensore di ostacoli S ($S=1$ se è presente un ostacolo, 0 altrimenti). Le uscite sono l'attivazione del motore M per far muovere il robot ($M=1$ robot in movimento $M=0$ robot fermo) e l'attivazione del sistema di aggiramento ostacoli A ($A=1$ aggira ostacolo, 0 altrimenti). Inizialmente il robot si trova nello stato "spento" dove il robot non si muove e non aggira ostacoli. Rimane in questo stato finché non viene attivato ($E=1$). A questo punto il robot si muove (stato "in movimento") finché è attivo e non sono presenti ostacoli. Nell'ipotesi che il robot sia attivo e siano presenti ostacoli, si raggiunge lo stato di "aggiramento" ponendo $A=1$ e si resta in questo stato finché $E=1$ e $S=1$ (è possibile raggiungere questo stato sia da "spento" che da "in movimento"). Il robot mantiene il motore attivo negli stati "aggiramento" "ed in movimento". Da questi due stati si torna allo stato "spento" non appena $E=0$. Si disegni il diagramma di stato usando il formalismo di Moore e si proponga un'implementazione a costo minimo del sistema utilizzando la codifica one-hot. Si calcoli il costo degli ingressi del sistema assumendo che ogni singolo flip-flop D abbia un costo pari a 14.
- 5) Si progetti un contatore che esegua il conteggio da -3 a 4, rappresentando i numeri in complemento a 2 su un numero minimo di bit. Si esegua il progetto utilizzando solamente flip-flop D, considerando gli stati non utilizzati in fase di progetto come condizioni di non specificazione. Si disegni poi il diagramma di stato indicando anche cosa accadrebbe qualora si presentassero in ingresso combinazioni considerate come condizioni di non specificazione in fase di progetto. Si discuta come andrebbero gestiti questi casi in base al tipo di applicazione in cui verrà utilizzato il contatore.
- 6) Nel progetto di un contatore binario sincrono è possibile utilizzare la tecnica del gating seriale oppure del gating parallelo. Si descrivano e confrontino le principali caratteristiche di queste due tecniche.

- 7) Si consideri il seguente codice VHDL che descrive un sommatore a 4 bit. Commentare sinteticamente gli statement del codice (riportare sul foglio il numero della riga seguito dal relativo commento). Si modifichi il codice per includere una porta OVF che indichi il verificarsi della condizione di overflow. Spiegare poi come dovrebbe essere modificato il codice per includere una porta OP che indichi se effettuare una somma (OP=0) oppure una sottrazione (OP=1). Si modifichi infine il codice per lavorare con notazione in valore assoluto invece che in complemento a 2.

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

4: entity adder4v2 is
5:     port(
6:         A,B : in std_logic_vector(3 downto 0);
7:         S   : out std_logic_vector(3 downto 0)
8:     );
9:
10: end adder4v2;

11: architecture arch of adder4v2 is
12:     signal sum : signed(3 downto 0);

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

16: end arch;
```