

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

CANDIDATO:

n. matricola:

- 1) Si progetti un compressore di dati che riceva in input una parola a 4 bit (A3 A2 A1 A0) e produca in uscita una parola su 2 bit (C1 e C0), secondo la seguente specifica. C1 dipende dai bit A3 ed A2, mentre C0 dai bit A1 ed A0. Ogni uscita Ci deve valere 0 quando la coppia di bit assegnata è uguale, 1 altrimenti. Si indichino tabella di verità, mappe di Karnaugh ed equazioni in forma minima. Si implementi poi il circuito utilizzando solo porte NAND e solo porte NOR, indicando quale delle due soluzioni risulti a costo minimo.
- 2) Si descriva la struttura di una Look-Up-Table (LUT) a 2 ingressi, indicandone l'utilità applicativa. Si progetti poi una LUT a 4 ingressi utilizzando solo LUT a 2 ingressi e multiplexer 2-1. Con la LUT progettata si implementi poi la seguente funzione booleana:

$$F = AB + ACD + \bar{B}C + A\bar{D}$$

- 3) Si descriva la struttura ed il funzionamento di un codificatore, discutendo in dettaglio il meccanismo di priorità. Ricavare poi tabella di verità, mappe di Karnaugh ed equazioni booleane per un encoder a 4 ingressi con priorità all'ingresso di peso minore, prevedendo anche un'uscita che indichi quando almeno uno degli ingressi assume il valore 1.
- 4) Si progetti un circuito in grado di riconoscere la sequenza 1001 all'interno di una sequenza arbitraria di bit. Il circuito dovrà essere dotato di un ingresso X e di un'uscita Z. L'uscita Z dovrà assumere il valore 1 tutte le volte che all'ingresso X sono già stati applicati i valori 100 e l'ingresso corrente vale 1. In tutti gli altri casi l'uscita Z deve valere 0. Si preveda anche la presenza di un segnale di Reset asincrono che riporti il circuito nello stato iniziale. Si utilizzi un'opportuna codifica di costo minimo per assegnare codici agli stati e si utilizzino solo flip-flop di tipo D. Quanto vale il costo degli ingressi del circuito, considerando il costo di ogni singolo flip-flop pari a 14 (ignorare il costo delle porte NOT)? Esempio di sequenza da riconoscere:

X: 0 0 0 1 0 0 1 0 0 1 1 0  
uscita Z: 0 0 0 0 0 0 1 0 0 1 0 0

- 5) Si progetti un contatore in complemento a 2 in grado di contare da -4 a 5 su un numero minimo di bit. Si considerino le combinazioni non utilizzate in fase di progetto come condizioni di non specificazione. Si realizzi il progetto utilizzando solo flip-flop di tipo D. Si indichi come andrebbe modificato il progetto per includere un segnale di ENABLE che permetta il normale conteggio quando ENABLE = 0 e blocchi il conteggio per ENABLE = 1; quando il circuito viene abilitato nuovamente dopo che il conteggio è stato bloccato, il conteggio deve ripartire dal valore al quale il conteggio era stato bloccato. Indicare infine il diagramma di stato, evidenziando cosa accadrebbe qualora si presentassero in uscita combinazioni considerate come condizioni di non specificazione in fase di progetto.
- 6) Si descrivano struttura e funzionamento di un registro a caricamento parallelo con possibilità di scorrimento dati in entrambe le direzioni.

- 7) Si consideri il seguente codice VHDL che descrive un contatore 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 un segnale di ENABLE attivo basso. Si indichi come andrebbe modificato il codice per aggiungere una porta SET con priorità minore rispetto al RESET che inizializzi il contenuto del registro con il valore contenuto in una porta INIT. Infine, includere un'uscita co che indichi il raggiungimento della massima dinamica di conteggio.

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

4: entity cnt4 is
5:     port(
6:         clk, reset : in std_logic;
7:         Q : out std_logic_vector(3 downto 0)
8:     );

9: end cnt4;

10: architecture arch of cnt4 is
11:     signal count : unsigned(3 downto 0);
12:     begin
13:         process(clk,reset)
14:             begin
15:                 if(reset='1') then
16:                     count <= "0000";
17:                 elsif(rising_edge(clk)) then
18:                     count <= count + "0001";
19:                 end if;
20:             end process;
21:             Q <= std_logic_vector(count(3 downto 0));

22: end arch;
```