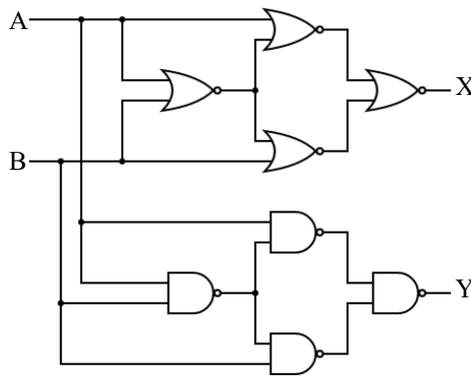


Prova d'esame di
RETI LOGICHE (Prof. E. Torti)
Lauree in Ing. Elettronica e Informatica DM 270 e DM 509
18 luglio 2025

CANDIDATO:

n. matricola:

- 1) Si consideri il circuito in figura. Ricavare, tramite algebra booleana la rete a due livelli in forma minima equivalente per le due funzioni. Quale operazione booleana è realizzata da ciascuna delle due funzioni? Quali applicazioni pratiche hanno? Infine, si implementino le due funzioni utilizzando un opportuno multiplexer.



- 2) Si realizzi un multiplexer 9-1 a 4 vie utilizzando solo multiplexer 8-1 ad una via e multiplexer 2-1 a quattro vie. I multiplexer devono essere connessi in modo che i codici di selezione da 0000 a 1000 possano essere applicati senza bisogno di logica aggiuntiva.
- 3) Si definiscano i termini implicante, primo implicante e primo implicante essenziale. Si indichino poi, per una funzione booleana a scelta, alcuni implicanti, tutti i primi implicanti e tutti i primi implicanti essenziali.
- 4) Si progetti un sistema di temporizzazione per il trasferimento seriale ciclico di un codice binario posto in ingresso ad un opportuno componente. Il sistema dovrà essere dotato di un ingresso di ENABLE che blocchi il trasferimento dati quando $ENABLE = 0$. Per il progetto non possono essere utilizzati registri a caricamento parallelo, ma solo altri componenti di logica sequenziale e/o combinatoria. Il codice binario da trasferire ha lunghezza pari a 6 bit. Realizzare il sistema utilizzando solo flip-flop JK e dovrà essere a costo minimo. Il sistema dovrà inoltre essere dotato di due ulteriori uscite chiamate start of code (SOC) che deve valere 1 unicamente in corrispondenza del primo bit del codice ed end of code (EOC) che deve valere 1 unicamente in corrispondenza dell'ultimo bit del codice. Indicare le equazioni di tutti i componenti del sistema e calcolare il costo degli ingressi assumendo che i singoli flip-flop abbiano un costo pari a 14.
- 5) Si progetti un sistema sequenziale in grado di verificare la corretta ricezione di una parola binaria dotata di bit di parità pari. Il sistema ha due ingressi X ed Y ed un'uscita E. La parola binaria viene posta, un bit alla volta, sull'ingresso X. L'uscita Y serve ad indicare se il valore post sull'ingresso X fa parte della parola binaria ($Y=0$) oppure se è il bit di parità ($Y=1$). Il sistema dovrà porre in uscita, in corrispondenza del bit di parità, il valore $E=1$ quando si è verificato un errore. In tutti gli altri casi il valore di E dovrà essere pari a 0. Si progetti il

sistema adottando la codifica one-hot e si calcoli il costo totale degli ingressi assumendo che il costo del singolo flip-flop sia pari a 14.

- 6) Si discutano e confrontino le varie modalità di descrizione di un'architettura hardware messe a disposizione del linguaggio VHDL, approfondendo i casi nei quali un tipo di descrizione risulti più vantaggiosa rispetto ad un'altra.
- 7) Si consideri il seguente codice VHDL che descrive un contatore a 6 bit. Commentare sinteticamente gli statement del codice (riportare sul foglio il numero della riga seguito dal relativo commento). Si indichi come andrebbe modificato il codice per aggiungere una porta enable attiva bassa. Modificare il codice per effettuare un conteggio all'indietro. Infine, modificare il codice per includere una porta di set attiva bassa con priorità maggiore rispetto al reset che inizializzi il contenuto del registro con la parola a 6 bit contenuta in una nuova porta INIT.

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

4: entity cnt6 is
5:     port(
6:         clk, reset : in std_logic;
7:         Q : out std_logic_vector(5 downto 0);
8:         co : out std_logic
9:     );

10: end cnt6;

11: architecture arch of cnt6 is
12:     signal count : unsigned(5 downto 0);
13:     begin
14:         process(clk,reset)
15:             begin
16:                 if(reset='1') then
17:                     count <= "000000";
18:                 elsif(rising_edge(clk) ) then
19:                     count <= count + "000001";
20:                 end if;
21:             end process;
22:             Q <= std_logic_vector(count(5 downto 0));
23:             co <= '1' when std_logic_vector(count(5 downto 0)) = "111111" else
24:                 '0';

25: end arch;
```