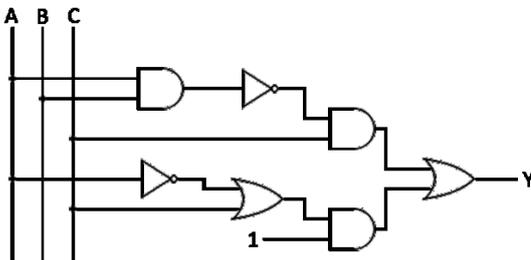


Prova d'esame di
RETI LOGICHE (Prof. E. Torti)
 Lauree in Ing. Elettronica e Informatica DM 270
 11 novembre 2020

CANDIDATO:

n. matricola:

- 1) Si calcoli il ritardo di propagazione del seguente circuito logico (considerando anche il ritardo prodotto dalle porte NOT sugli ingressi) conoscendo le caratteristiche delle porte logiche indicate in tabella (t_{PHL} è il ritardo di propagazione per la transizione alto-basso, mentre t_{PLH} è il ritardo di propagazione per la transizione basso-alto). Ricavare poi la tabella di verità e minimizzarla tramite mappa di Karnaugh. Considerando il costo degli ingressi, è più conveniente implementare la rete semplificata con solo porte NAND o con sole porte NOR?



Porta	t_{PHL}	t_{PLH}
AND	0,4 ns	0,2 ns
OR	0,1 ns	0,6 ns
NOT	0,1 ns	0,1 ns

- 2) Si progetti un full-adder, ricavando la tabella di verità, le mappe di Karnaugh e le equazioni algebriche. Utilizzare poi il full-adder come blocco elementare per progettare un sottrattore a 4 bit con riporto in uscita. Si applichi poi la tecnica della contrazione per progettare un circuito “decrementa di 5” per numeri a 4 bit senza riporto in uscita.
- 3) Si descrivano i diversi metodi utilizzabili per generare i bit di configurazione di una FPGA, discutendo per ciascuno di questi le principali caratteristiche.
- 4) Si progetti un circuito in grado di riconoscere la sequenza 1011 all’interno di una sequenza arbitraria di bit. Il circuito è dotato di un ingresso X e di un’uscita Z. L’uscita Z dovrà valere 1 tutte le volte che all’ingresso X sono già stati applicati i valori 101 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 attivo alto che riporti il circuito nello stato iniziale. Si utilizzi la codifica Gray per assegnare i codici agli stati e si proponga l’implementazione a costo minimo del circuito utilizzando solo flip-flop di tipo D. Come cambierebbe il circuito se il sistema dovesse essere inizializzato dal segnale di Reset asincrono allo stato finale invece che allo stato iniziale?
 Esempio di sequenza da riconoscere: X: 0000010110010110110
 Z: 0000000010000010010
- 5) Si descrivano l’utilità, il funzionamento e le diverse tipologie di ingressi diretti considerando un generico circuito sequenziale.
- 6) Si progetti un contatore, con un numero opportuno di bit, che segua la sequenza di conteggio 0-2-4-6-8 utilizzando solo flip-flop di tipo D. Considerare gli stati non utilizzati come condizioni di non determinazione. Calcolare il costo degli ingressi assumendo che ogni flip-flop abbia un costo pari a 14. Disegnare il diagramma di stato indicando chiaramente cosa accadrebbe qualora si presentassero in uscita gli stati considerati come condizioni di non specificazione. Discutere come andrebbe modificato il progetto nel caso in cui questo

contatore dovesse essere utilizzato in diversi tipi di applicazioni (ad esempio in un giocattolo per bambini oppure per il controllo del motore di un veicolo – non è necessario rifare il progetto per i due casi, basta la spiegazione).

- 7) Si consideri il seguente codice VHDL che descrive un codificatore a 4 bit con priorità all'ingresso con peso maggiore. 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 una porta di ENABLE attivo basso. Indicare come dovrebbe essere modificato il codice per dare la priorità all'ingresso con peso inferiore invece che all'ingresso con peso maggiore. Infine indicare come andrebbe aggiunta una porta di output M che assume valore logico alto se e solo se il valore posto in ingresso D è un multiplo intero di 2.

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

3:entity cod_p is
4:    port(
5:        D : in std_logic_vector(3 downto 0);
6:        X : out std_logic_vector(1 downto 0);
7:        V : out std_logic
8:    );
9:end cod_p;

10:architecture arch of cod_p is
11:begin
12:    process(D)
13:    begin
14:        if D(3) = '1' then
15:            X <= "11";
16:            V <= '1';
17:        elsif D(2) = '1' then
18:            X <= "10";
19:            V <= '1';
20:        elsif D(1) = '1' then
21:            X <= "01";
22:            V <= '1';
23:        elsif D(0) = '1' then
24:            X <= "00";
25:            V <= '1';
26:        else
27:            X <= "--";
28:            V <= '0';
29:        end if;
30:    end process;
31:end arch;
```