

Verilog - Scurtă recapitulare

Probleme propuse

Oprîtoiu Flavius
flavius.opritoiu@cs.upt.ro

October 2, 2024

Problema 1

Implementați următoarele funcții Booleene utilizând doar porți NAND în Verilog.

- $f_1 = \sum(0, 2, 3, 4, 5, 7)$
- $f_2 = \sum(1, 5, 7, 8, 9, 10, 11, 13, 15)$
- $f_3 = \sum(0, 4, 5, 15) + \sum_d(2, 7, 8, 10, 12, 13)$
- $f_4 = \sum(1, 4, 5, 13, 14, 15) + \sum_d(7, 8, 9, 12)$
- $f_5 = \sum(1, 3, 4, 5, 11, 12, 14) + \sum_d(6, 7, 9)$

Notă: Pentru a implementa f_1 , descărcați [▶ ex1a.v](#) și [▶ run_ex1a.txt](#), completați codul Verilog în `ex1a.v` din linia 5 și simulați prin comanda `do run_ex1a.txt`.

Similar, pentru f_2 folosiți [▶ ex1b.v](#) și [▶ run_ex1b.txt](#).

Pentru f_3 folosiți [▶ ex1c.v](#) și [▶ run_ex1c.txt](#).

Pentru f_4 folosiți [▶ ex1d.v](#) și [▶ run_ex1d.txt](#).

Pentru f_5 folosiți [▶ ex1e.v](#) și [▶ run_ex1e.txt](#).

Problema 2

Proiectați un modul având o intrare i , pe 3 biti și o ieșire o , pe 1 bit. Valoarea de la intrarea i este un număr întreg, nenegativ iar ieșirea o este definită de relația de mai jos. Implementați unitatea în Verilog.

$$o = \begin{cases} 1 & \text{dacă } i = 4k - 1, k \in \mathbb{N} \\ 0 & \text{altfel} \end{cases}$$

Notă: Pentru implementarea modulului, descărcați [▶ ex2.v](#) și [▶ run_ex2.txt](#), completați codul Verilog în `ex2.v` și simulați codul prin `do run_ex2.txt`.

Problema 3

Implementați convertorul Binary Coded Decimal dat prin tabelul de adevăr de mai jos. Construiți codul Verilog al acestui modul.

Inputs				Outputs			
i_3	i_2	i_1	i_0	o_3	o_2	o_1	o_0
0	0	0	0	0	0	0	0
0	0	0	1	0	1	1	1
0	0	1	0	0	1	1	0
0	0	1	1	0	1	0	1
0	1	0	0	0	1	0	0
0	1	0	1	1	0	1	1
0	1	1	0	1	0	1	0
0	1	1	1	1	0	0	1
1	0	0	0	1	0	0	0
1	0	0	1	1	1	1	1

Problem 3 (contd.)

Notă: Pentru implementarea modulului, descărcați `▶ ex3.v` și `▶ run_ex3.txt`, completați codul Verilog în `ex3.v` și simulați codul prin `do run_ex3.txt`.

Problema 4

Construiți un modul Verilog, care primește la intrarea i , pe 6 biți, numere întregi, nenegative. Ieșirea pe 1 bit, numită $is6$, va fi activă dacă cifra zecilor din reprezentarea zecimală a numărului de la intrare este 6 (Exemplu: $i = 32 \rightarrow is6 = 0$; $i = 60 \rightarrow is6 = 1$; $i = 63 \rightarrow is6 = 1$)

Notă: Pentru implementarea modulului, descărcați [▶ ex4.v](#) și [▶ run_ex4.txt](#), completați codul Verilog în `ex4.v` și simulați codul cu `do run_ex4.txt`.

Problema 5

Construiți un modul Verilog, care primește la intrarea i , pe 6 biți, numere întregi, nenegative. Ieșirea pe 1 bit, numită $is6$, va fi activă dacă cifra zecilor din reprezentarea zecimală a numărului de la intrare este 6 (Exemplu: $i = 32 \rightarrow is6 = 0$; $i = 60 \rightarrow is6 = 1$; $i = 63 \rightarrow is6 = 1$)

Notă: Pentru implementarea modulului, descărcați [▶ ex4.v](#) și [▶ run_ex4.txt](#), completați codul Verilog în `ex4.v` și simulați codul cu `do run_ex4.txt`.