

Nume:

Test AC

Varianta 13

- Se consideră o unitate combinațională care acceptă la intrările sale numere pe 6 biți reprezentate în $C1$ (1 bit de semn și 5 biți de mărime) și generează la ieșirile sale cea mai semnificativă cifră zecimală a numărului cu semn schimbat. Ex: dacă la intrare avem numărul 27 ieșirea va fi -2, în timp ce pentru configurația -8 ieșirea va fi 8.
 - Implementați, folosind limbajul Verilog, modulul ***msd_c1***.
 - Redactați un modul testbench pentru testarea modulului ***msd_c1***, implementat la punctul a).
- Se consideră un dispozitiv combinațional având o intrare i și o ieșire o pe 4 biți, care codifică un număr pe 4 biți în formatul Gray, codificare descrisă în tabelul de mai jos.

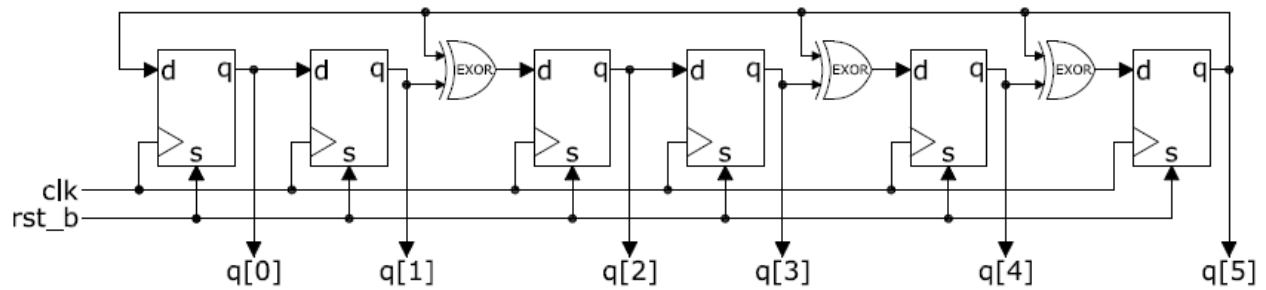
Inputs				Outputs			
I_3	I_2	I_1	I_0	O_3	O_2	O_1	O_0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	0
0	1	0	1	0	1	1	1
0	1	1	0	0	1	0	1
0	1	1	1	0	1	0	0
1	0	1	0	1	1	1	1
1	0	1	1	1	1	1	0

Obs: Restul valorilor nelistate sunt considerate elemente don't care

- Minimizați pe foaie funcțiile de la ieșire folosind metoda diagramelor Karnaugh.
- Redactați un modul care implementează funcția booleană rezultată după minimizare. Modulului i se va atribui un nume sugestiv (ex. ***gray_encoder***).

Nume:

3. Se consideră arhitectura unui Linear Feedback Shift Register (**LFSR**) de 6 ranguri:



- Să se determine periodicitatea secvenței de ieșire generând toți vectorii nenuli pe 6 biți.
- Să se implementeze, folosind **metoda instanțierii multiple**, structura **LFSR** prezentată mai sus.
- Să se redacteze un modul **testbench** care să verifice funcționalitatea modului **LFSR** implementat anterior.