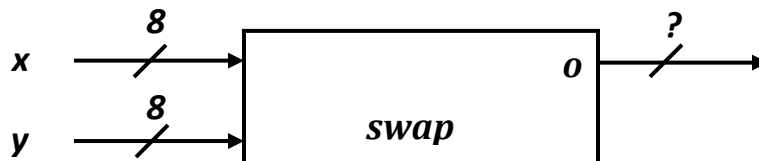


Test AC

Varianta 7

1. Se consideră un modul de prelucrare a două intrări pe 8 biți, așa cum este ilustrat mai jos.



- a) Dacă bitul de paritate ai intrării x coincide cu bitul de paritate ai intrării y atunci se va inversa jumătatea mai semnificativă ai intrării x cu jumătatea mai puțin semnificativă ai intrării y , iar intrările combinate vor fi direcționate spre ieșirea o . În caz contrar se va inversa jumătatea mai puțin semnificativă ai intrării x cu jumătatea mai semnificativă ai intrării y , intrările combinate fiind furnizate la ieșirea o .
- b) Redactați un **testbench** care să verifice funcționalitatea modului Verilog implementat anterior.

2. Se consideră următorul tabel de adevăr:

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

Obs: Mintermii de la 10 \rightarrow 15 sunt considerați elemente don't care

Pe baza configurațiilor binare date în tabel să se :

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

Punctaj: 1. a) 2p; b) 2p | 2. a) 1p; b) 1p | 3. a) 1p; b) 1p; c) 1p | **1 punct din oficiu** | Total: 10p

Data:

Nume:

3. Se consideră un Linear Feedback Shift Register (**LFSR**) de 4 ranguri, construit conform secvenței de ieșire: $q[0] \leftarrow q[3]; q[1] \leftarrow q[0] \oplus q[3]; q[2] \leftarrow q[1]; q[3] \leftarrow q[2] \oplus q[3]$.
- a) Să se deseneze pe o foaie arhitectura completă a LFSR-ului conform secvenței de propagare menționată mai sus.
 - b) Să se implementeze, folosind **vectori de instanțe**, structura LFSR proiectată la subpunctul a).
 - c) Să se redacteze, folosind limbajul Verilog, un **testbench** care să verifice funcționalitatea modului Verilog implementat anterior.