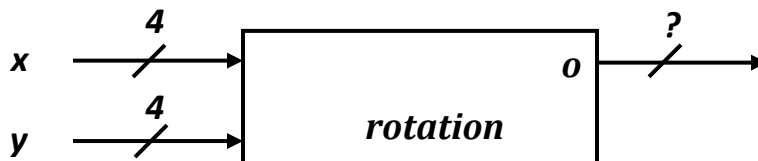


# Test AC

## Varianta 9

1. Se consideră un modul de prelucrare a două intrări pe 4 biți, așa cum este ilustrat mai jos.



- a) Dacă bitul cel mai semnificativ al intrării  $x$  coincide cu bitul cel mai puțin semnificativ al intrării  $y$  atunci se va roti conținutul intrării  $x$  cu două poziții la dreapta și va fi livrat la ieșirea  $o$ . În caz contrar se va roti conținutul intrării  $y$  cu două poziții la stânga și va fi furnizat 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	1	0	1
0	0	0	1	0	0	1
0	0	1	0	0	0	0
0	0	1	1	1	1	1
0	1	0	0	0	0	1
0	1	0	1	0	0	0
0	1	1	0	1	0	1
0	1	1	1	0	1	0
1	0	0	0	0	0	0
1	0	0	1	1	0	0

Obs: Mintermii de la 10 → 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**).

Data:

Nume:

3. Se consideră un Linear Feedback Shift Register (**LFSR**) de 6 ranguri, construit conform secvenței de ieșire:  $q[0] \leftarrow q[5]; q[1] \leftarrow q[0]; q[2] \leftarrow q[1] \oplus q[5]; q[3] \leftarrow q[2]; q[4] \leftarrow q[3]; q[5] \leftarrow q[4]$ .
- 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 modulului Verilog implementat anterior.