

# Test AC

## Varianta 4

- Proiectați un circuit combinațional cu o intrare  $i$ , pe 4 biți, și o ieșire  $o$ , pe 3 biți. Când valoarea binară de la intrarea  $i$  este 0, 1, 2 sau 3, ieșirea are valoarea  $(2 \cdot i + 7) \bmod 6$  (obținând deci valorile 1, 3, 5 și 1). Când intrarea este 4, 5, 6, 7, 8, 9 sau 10 ieșirea are valoarea  $(3 \cdot i + 2) \bmod 8$ . Când intrarea  $i$  este 11, 12, 13, 14 sau 15 ieșirea va fi câtul împărțirii lui  $i$  la 3.
- Se consideră următorul tabel de adevăr:

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

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

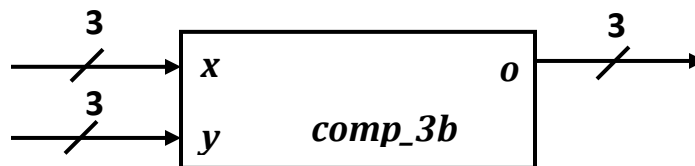
Pe baza configurațiilor binare date în tabel să se :

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

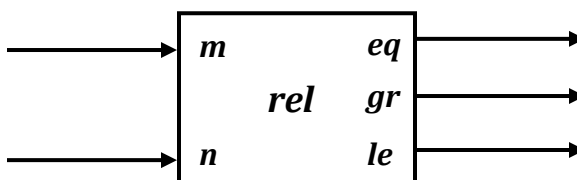
Data:

Nume:

3. Construiți, folosind limbajul Verilog, un modul pentru compararea a două numere pe 3 biți, fără a folosi operatorii Verilog relaționali.



- a) Desenați arhitectura completă a comparatorului folosind instanțe ale unității **rel** pentru două numere pe 1 bit dată mai jos:



Obs: Pentru intrările/ ieșirile menționate au loc relațiile:

$$\left\{ \begin{array}{l} eq=1 \text{ dacă } m = n \\ gr=1 \text{ dacă } m > n \\ le=1 \text{ dacă } m < n \end{array} \right.$$

- b) Să se implementeze în limbajul Verilog arhitectura **comp\_3b** desenată la punctul a).
- c) Să se redacteze folosind limbajul Verilog un **testbench** pentru verificarea exhaustivă a modulului Verilog implementat anterior.