Cap 3. *Bazele logice* CN
(BL ale CN)

3.1. Definiția axiomatică a algebrei Boole

3.2. Reprezentarea funcțiilor booleene (FB): expresie booleană, tabel de adevăr

3.3. Forme normale şi canonice. Complement.

3.4. Funcții logice de bază

3.5. Analiza şi sinteza FB

3.6. Blocuri logice şi funcționale

3.1. Definiția axiomatică a algebrei Boole

Se numeşte **ℓ - latice** – o mulțime nevidă înzestrată cu 2 legi de compoziție:
- „$\bigcup$"**reuniune** (disjuncție) şi
- „$\bigcap$" **intersecție** (conjuncție), care satisfac următoarele axiome:
  1. idempotența: $\forall A \in \ell, A \bigcup A = A,$
     $A \bigcap A = A$
  2. comutativitatea: $(\forall) A, B \in \ell, A \bigcup B = B \bigcup A,$
     $A \bigcap B = B \bigcap A$
  3. asociativitatea: $(\forall) A, B, C \in \ell, A \bigcup (B \bigcup C) = (A \bigcup B) \bigcup C,$
     $(A \bigcap B) \bigcap C = A \bigcap (B \bigcap C)$
  4. absorbția: $(\forall) A, B \in \ell, A \bigcup (A \bigcap B) = A;$
     $A \bigcap (A \bigcup B) = A$
  5. distributivitatea: $(\forall) A, B, C \in \ell, A \bigcup (B \bigcap C) = (A \bigcup B) \bigcap (A \bigcup C),$
     $A \bigcap (B \bigcup C) = (A \bigcap B) \bigcup (A \bigcap C)$

Prim element al laticei – dacă ($\exists$) „0" cu proprietatea: A$\bigcup$0 = A; A $\bigcap$ 0 = 0

Ultim element al laticei – dacă ($\exists$) „1" cu proprietatea: A$\bigcup$1 = 1; A $\bigcap$ 1 = A

3.1. (*)

O a**lgebră BOOLE** este o latice ℓ *distributivă cu prim şi ultim element* pentru care se dă o a III-a lege de compoziţie numită **negaţie** „-” şi pentru care sunt valabile următoarele postulate:

P1. *postulatul dublei negaţii*: $\overline{\overline{A}} = A$

P2. *relaţiile lui de Morgan*: $\overline{A \bigcup B} = \overline{A} \bigcap \overline{B}$; $\overline{A \bigcap B} = \overline{A} \bigcup \overline{B}$

P3. *postulatul contradicţiei*: $A \bigcap \overline{A} = 0$

P4. *postulatul terţului exclus*: $A \bigcup \overline{A} = 1$

3.2. ***Reprezentarea funcţiilor booleene (FB): expresie booleană, tabel de adevăr***

FB (logică) de n variabile $f(x_1,\dots, x_n)$ este o funcţie definită pe produsul cartezian
$f: B_2 \times B_2 \times \dots \times B_2 \to B_2$, *a.î.* $f(i_{1j},\dots, i_{nj}) \in \{0,1\}$, $(\forall)\ j=\overline{1,2^n}$

Dacă o FB depinde de *n* variabile, atunci numărul FB distincte care se pot construi sunt
$N=2^{2^n}$

Pentru a fi **complet definită** o FB de n variabile trebuie cunoscută în $k= 2^n$ *puncte distincte*

FB se pot reprezzenta prin:
- tabele de adevăr
- expresii booleene

Cele 2 sunt echivalente

Exp.: FB de o variabilă

3.3. Forme normale şi canonice. Complement.

Forme normale (FN):
- FND – disjunctivă = FN ai cărei termeni sunt de tip produs logic, uniţi prin sume logice (U, +)
- FNC – conjunctivă = FN ai cărei termeni sunt de tip sumă logică, uniţi prin produs logic ($\cap, *$)

Dacă un termen de tip produs sau sumă logică conţine mai puţine variabile, se numeşte **termen elementar.**

Formele canonice (FC) = FN ai cărei termeni conţin toate variabilele funcţiei
- FCD – disjunctivă = sumă de *mintermeni*, pentru care funcţia ia valoarea logică 1;
  (x=1)
- FCC – conjunctivă = produs de *maxtermeni*, pentru care funcţia ia valoarea logică 0.
  (x=0)

FCD este echivalentă cu FCC.

FCD şi FCC sunt UNICE pentru o funcţie dată.

Forme canonice complete –conţin toţi cei 2^n termeni canonici posibili:
- FCDC = 1
- FCCC = 0

3.4.Funcţii logice de bază

SI
SAU
NU

y / x	0	01
0	m_0 M_0	m_1 M_1
1	m_2 M_2	m_3 M_3

Funcţii logice de bază pentru F.B. de 2 variabile:
FCD: x=1, $\bar{x}$=0

x	*0 0 1 1*	FCD	Funcţia	Denumire funcţie
y	*0 1 0 1*			
	$m_0m_1m_2m_3$			
f_0	0 0 0 0	$f_0=0$	$f_0=0$	ZERO
f_1	0 0 0 1	$f_1= m_3$	$f_1= x \bullet y$	ŞI – are val.1 dacă toate var. au val. 1
f_2	0 0 1 0	$f_2= m_2$	$f_2= x \bullet \bar{y}$	INHIBARE ($x_1 \bullet x_2 \bullet ... \bullet \overline{x_k} \bullet ... \bullet x_n$)
f_3	0 0 1 1	$f_3= m_2+ m_3$	$f_3= x$	IDENTITATE
f_4	0 1 0 0	$f_1= m_1$	$f_4= \bar{x} \bullet y$	INHIBARE
f_5	0 1 0 1	$f_5 = m_1+ m_3$	$f_5 = y$	IDENTITATE
f_6	0 1 1 0	$f_6 = m_1+ m_2$	$f_6 = x \oplus y$	SAU EXCLUSIV
f_7	0 1 1 1	$f_7 = m_1+ m_2+ m_3$	$f_7 = x + y$	SAU – are val. 1, dacă cel puţin o var. are val. 1
f_8	1 0 0 0	$f_8 = m_0$	$f_8 = \bar{x} \bullet \bar{y}$	SAU-NU (NICI, NOR) – areval.0 daca cel puţin o var. are val.1
f_9	1 0 0 1	$f_9 = m_0+ m_3$	$f_9 = \bar{x} \bullet \bar{y} + x \bullet y$	COINCIDENŢA
f_{10}	1 0 1 0	$f_{10} = m_0+ m_2$	$f_{10} = \bar{y}$	NU
f_{11}	1 0 1 1	$f_{11} = m_0+ m_2+ m_3$	$f_{11} = x + \bar{y}$	IMPLICARE ($x_1 + x_2 + ... + \overline{x_k} + ... + x_n$)
f_{12}	1 1 0 0	$f_{12} = m_0+ m_1$	$f_{12} = \bar{x}$	NU
f_{13}	1 1 0 1	$f_{13} = m_0+ m_1+ m_3$	$f_{13} = \bar{x} + y$	IMPLICARE
f_{14}	1 1 1 0	$f_{14} = m_0+ m_1+ m_2$	$f_{14} = \overline{x \bullet y}$	SI-NU (NAND) – are val.1, dacă cel puţin o var. are val.0
f_{15}	1 1 1 1	$f_{15} = m_0+ m_1+ m_2+ m_3$	$f_{15} =1$	UNU

3.5. *Analiza şi sinteza FB*

În utilizarea CN apar 2 situații:

1. se cunosc: - I/ într-o schemă logică + schema logică => se cer /E : *analiza FB*=> dezvoltarea FB, adică *FN=>FC*
2. se cunosc: - I/ + /E => se cere blocul logic care realizează transformarea : *sinteza blocului logic* = minimizarea FB (reducerera nr. total de termeni ai formei analitice la un nr. minim şi care realizeaza aceeaşi operație logică => scade nr. blocurilor logice => scade costul al circuitului)

Sinteza FB

Reprezintă simplificarea FB, adică obținerea unei forme analitice similare cu cea inițială şi care conține mai puține variabile.

Metode:

- Karnaugh
- Veitch
- Haward

Metoda Karnaugh de minimazare a FB

3.5. (*)

Metoda Karnaugh de minimizare a FB

1. Forma analitică se dezvoltă într-o formă canonică: FCC sau FCD.
2. Se împarte numărul de variabile ale funcției în **2 grupe cât mai egale** şi se construieşte diagrama Karnaugh (variabilele sunt dispuse în **codul Gray** – > două căsuțe adiacente diferă printr-o singură poziție).
3. Se completează valorile funcției în diagramă: **cifra 1** pentru termenii din **FCD,** respectiv **cifra 0** pentru termenii din **FCC.** Se vor obține atâtea cifre de 1, respectiv 0, câți termini are FCD, respectiv FCC.
4. Se trece efectiv la minimizare: se grupează **2^k (k=0,1,2,3,....)** suprafețe elementare adiacente care au valoarea logică 1, respectiv 0, astfel încât să formeze linie completă, coloană completă, o suprafața dreptunghiulară, un pătrat (suprafețele extinse pot avea 1 element, 2, 4, 8 etc. elemente adiacente).
5. Corespunzător unei suprafețe extinse se scrie **un produs elementar al variabilelo**r la FCD, respectiv **o sumă elementară a variabilelor** la FCC. Pentru suprafețele care nu pot fi alipite altora, se va scrie expresia mintermenului corespunzător, pentru FCD, respectiv al maxtermenului la FCC.
6. Se scrie forma minimă f_{1min} ca fiind constituită din $\cup$ ecuațiilor suprafețelor elementare în FCD, respectiv $\cap$ ecuațiilor suprafețelor elementare în FCC.

Se pot obține alte 3 forme minime:

- f_{2min} – se obține din minimizarea funcției complementare funcției date (se trec valori în casuțele libere de la funcția inițială);
- f_{3min} – negarea lui f_{1min};
- f_{4min} – negarea lui f_{2min}.

Se face **un tabel al celor 4 forme minime** calculând la fiecare numărul de blocuri SI, SAU, NU.

Foma ***minim minimorum***, adica minimizarea absolută a funcției este dată de funcția cu numărul minim de blocuri utilizate.

3.5. (*)

Exp.: Diagrama Karnaugh pentru o funcție de 4 variabile f(x,y,z,t)

zt / xy	00	01	11	10
00	m_0 M_0	m_1 M_1	m_3 M_3	m_2 M_2
01	m_4 M_4	m_5 M_5	m_7 M_7	m_6 M_6
11	m_{12} M_{12}	m_{13} M_{13}	m_{15} M_{15}	m_{14} M_{14}
10	m_8 M_8	m_9 M_9	m_{11} M_{11}	m_{10} M_{10}