Cursul nr. 1

C1.1. Sistemul de numerație binar

Sistemul binar utilizează pentru reprezentarea numerelor doar două cifre **1** şi **0**. Numărul 1101011 este un exemplu de număr binar.

Conversia binar-zecimală. Baza sistemului de numerație binar fiind **2**, pentru conversia binar-zecimală se atribuie fiecărei cifre binare care intră în componența unui număr ponderea puterii lui **2**, corespunzătoare poziției cifrei respective din numărul binar. Cifra binară (bit) situată pe poziția cea mai din dreapta are ponderea 2^0. Celelalte cifre binare, considerate spre stânga, au ponderile 2^1, 2^2, 2^3 etc.

De exemplu, valoarea în zecimal a numărului binar de mai sus, 1101011 este:

$$1101011 = 1\times 2^6 + 1\times 2^5 + 0\times 2^4 + 1\times 2^3 + 0\times 2^2 + 1\times 2^1 + 1\times 2^0 = 107$$

În cazul numerelor fracționare, la fel ca în sistemul zecimal, se atribuie cifrelor aflate la dreapta virgulei de separație, ponderile fracționare 2^{-1}, 2^{-2}, 2^{-3} etc. De exemplu:

$$101{,}1011 = 1\times 2^2 + 0\times 2^1 + 1\times 2^0 + 1\times 2^{-1} + 0\times 2^{-2} + 1\times 2^{-3} + 1\times 2^{-4} = 5\frac{11}{16} = 5{,}6875$$

Conversia zecimal-binară pentru numere întregi se face după metoda împărțirii repetate prin **2**. La fiecare împărțire se obține un cât şi un rest. Împărțirea se consideră încheiată când se ajunge la un cât egal cu zero. Valorile restului (**0** sau **1**) formează cifrele numărului binar, ordinea de citire a acestor cifre fiind de jos în sus (fig. C1.1). De exemplu, prin conversia în binar a numărului zecimal 217 obținem numărul binar 11011001.

$$\begin{aligned}
217 : 2 &= 108\ \textit{rest}\ 1\\
108 : 2 &= 54\ \textit{rest}\ 0\\
54 : 2 &= 27\ \textit{rest}\ 0\\
27 : 2 &= 13\ \textit{rest}\ 1\\
13 : 2 &= 6\ \textit{rest}\ 1\\
6 : 2 &= 3\ \textit{rest}\ 0\\
3 : 2 &= 1\ \textit{rest}\ 1\\
1 : 2 &= 0\ \textit{rest}\ 1
\end{aligned}$$

SENSUL DE CITIRE

Fig. C1.1 Conversia zecimal-binară a numărului întreg 217

În cazul **numerelor fracționare se efectuează înmulțiri repetate cu 2**. După prima înmulțire rezultă un număr fracționar. Partea întreagă, 0 sau 1, devine prima cifră din numărul binar fracționar. Partea fracționară se înmulțeşte în continuare cu doi şi după acelaşi algoritm de la prima cifră se obțin celelalte cifre binare. Înmulțirile se opresc atunci când partea întreagă devine 1 iar partea fracționară 0.

De exemplu, numărul zecimal fracționar 0,40625 are echivalentul binar 0,01101 (fig. C1.2):

$$\begin{aligned}
&0{,}40625 \times 2 = 0{,}81250\ \textit{parte întreagă} = 0\\
&0{,}81250 \times 2 = 1{,}6250\ \textit{parte întreagă} = 1\\
&0{,}625 \quad \times 2 = 1{,}250\ \textit{parte întreagă} = 1\\
&0{,}25 \quad\ \times 2 = 0{,}5\ \textit{parte întreagă} = 0\\
&0{,}5 \quad\ \ \times 2 = 1\ \textit{parte întreagă} = 1
\end{aligned}$$

SENSUL DE CITIRE

Fig. C1.2 Conversia zecimal-binară a numărului fracționar 0,40625

Operațiile aritmetice cu numere binare, cum ar fi: adunarea, scăderea şi înmulțirea, se prezintă în fig. C1.3.

Adunarea
$0+0=0$
$0+1=1$
$1+0=1$
$1+1=10$

Scăderea
$0-0=0$
$1-0=1$
$1-1=0$
$10-1=1$

Înmulțirea
$0\times 0=0$
$0\times 1=0$
$1\times 0=0$
$1\times 1=1$

Adunarea **Scăderea** **Înmulțirea**

Fig. C1.3 Operații aritmetice cu numere binare

EXEMPLUL C1.1

Să se transforme numărul binar 1011101,1011101 *în echivalentul său zecimal.*
Răspuns: *partea întreagă se converteşte după relația:*

$$1011101 = 1\times 2^6 + 0\times 2^5 + 1\times 2^4 + 1\times 2^3 + 1\times 2^2 + 0\times 2^1 + 1\times 2^0$$

sau

$$1011101 = 64 + 16 + 8 + 4 + 1 = 93$$

Partea fracționară se scrie:

$$0{,}1011101 = 1\times 2^{-1} + 0\times 2^{-2} + 1\times 2^{-3} + 1\times 2^{-4} + 1\times 2^{-5} + 0\times 2^{-6} + 1\times 2^{-7}$$

sau

$$0{,}1011101 = 0{,}5 + 0{,}125 + 0{,}0625 + 0{,}03125 + 0{,}0078125 = 0.7265625\,.$$

Răspunsul final este:

$$1011101{,}1011101\big|_2 = 93{,}7265625\big|_{10}\,.$$

EXEMPLUL C1.2

Să se transforme în binar numărul zecimal 23,30625.
Răspuns: *partea întreagă, aplicând algoritmul de mai jos:*
$23:2=11$ *rest* **1***;* $11:2=5$ *rest* **1***;* $5:2=2$ *rest* **1***;* $2:2=1$ *rest* **0***;* $1:2=0$ *rest* **1***,*
se scrie: $23\big|_{10} = 10111\big|_2$.
Partea fracționară, aplicând algoritmul de mai jos:
$0{,}30625\times 2 = 0{,}6125$ *parte întragă* **0***;* $0{,}6125\times 2 = 1{,}225$ *parte întreagă* **1***;*
$0{,}225\times 2 = 0{,}45$ *parte întragă* **0***;* $0{,}45\times 2 = 0{,}9$ *parte întragă* **0***;*
$0{,}9\times 2 = 1{,}8$ *parte întreagă* **1***;* $0{,}8\times 2 = 1{,}6$ *parte întreagă* **1***;*
$0{,}6\times 2 = 1{,}2$ *parte întreagă* **1** *etc.*
se scrie: $0{,}30625\big|_{10} = 0{,}0100111\ldots\big|_2$

Observație.
În acest caz nu se îndeplinesc condițiile de oprire a înmulțirilor cu **2**. *Considerând numai* 7 *zecimale, numărul binar obținut reprezintă în zecimal* 0,3046875*, abaterea realtivă fiind de* –0,5%. *Pentru o precizie mai bună a conversiei trebuie luate în seamă mai multe zecimale.*

C1.2. Algebra booleană, funcții logice

Algebra logică se mai numeşte şi **booleană** după numele lui George Boole (1815-1864), matematician irlandez, unul din întemeietorii şi inițiatorii studiului logicii matematice.
Algebra booleană operează pe o mulțime binară ***B*** al cărei element generic poate lua doar două valori: **0** şi **1**.
Pe această mulțime binară ***B*** se definesc trei operatori (funcții logice):

- negația (funcția **NU**),
- produsul logic (funcția **ŞI**) şi
- suma logică (funcția **SAU**).

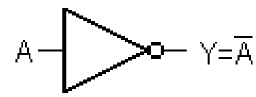
Funcția de negare transformă pe **0** în **1** şi pe **1** în **0**. Semnul operației este o bară trasată deasupra mărimii care se neagă: $\overline{1}=0$; $\overline{0}=1$. Variabila *A* negată se scrie $\overline{A}$ şi se citeşte *A negat* sau *non A*.

Funcția de negare este definită prin tabelul de adevăr din fig. C1.4*a*. Variabila independentă *A* şi funcția rezultantă *Y* se modifică în cadrul operației de negare, după relația:

$$Y = \overline{A} \qquad (1)$$

Elementul logic care realizează funcția (7.1) este reprezentat simbolic în fig. C1.4*b*.

A	Y
0	1
1	0



(a) *(b)*

Fig. C1.4 Funcția de negare: (a) tabelul de adevăr; (b) simbolul operatorului logic

Produsul logic sau conjuncția a două sau mai multe mărimi binare se notează cu semnul înmulțirii (•) sau cu semnul intersecție ($\bigcap$):

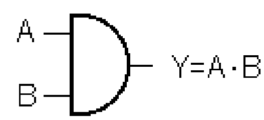
$$Y = A \cdot B \cdot C \cdot ... = A \bigcap B \bigcap C \bigcap ... \qquad (2)$$

şi se citeşte ***A şi B şi C şi …***

Produsul logic este definit prin tabelul de adevăr, care în cazul a două variabile are aspectul din fig. C1.5*a*.

Elementul logic care realizează funcția (7.2) se numeşte element **ŞI**. Reprezentarea simbolică se prezintă în fig. C1.5*b*.

A	B	Y=A·B
0	0	0
0	1	0
1	0	0
1	1	1



(a) *(b)*

Fig. C1.5 Produsul logic: (a) tabelul de adevăr; (b) simbolul operatorului logic

Suma logică sau disjuncția a două sau mai multe mărimi binare se notează cu semnul adunării (+) sau cu semnul reuniune ($\bigcup$):

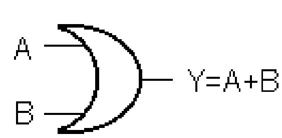
$$Y = A + B + C + ... = A \bigcup B \bigcup C \bigcup ... \qquad (3)$$

şi se citeşte ***A sau B sau C sau …***

Suma logică se defineşte prin tabelul de adevăr, care în cazul a două variabile are aspectul din fig. C1.6*a*.

Elementul logic care realizează funcția (7.3) se numeşte element **SAU** şi este reprezentat simbolic în fig. C1.6*b*.

A	B	Y=A+B
0	0	0
0	1	1
1	0	1
1	1	1



(a) *(b)*

Fig. C1.6 Suma logică: (a) tabelul de adevăr; (b) simbolul operatorului logic

Cu ajutorul celor trei operații logice fundamentale se pot defini **operații binare complexe**. Dintre acestea mai des utilizate sunt:

- negarea funcției ŞI, numită şi funcția **ŞI-NU** (NAND):

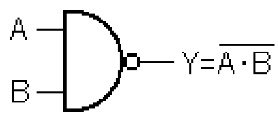
$$Y = \overline{A \cdot B} \qquad (4)$$

- negarea funcției SAU, numită şi funcția **SAU-NU** (NOR):

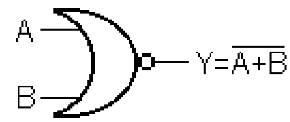
$$Y = \overline{A + B} \qquad (5)$$

Simbolul acestor funcții complexe şi tabelele de adevăr se prezintă în fig. C1.7.

A	B	$Y=\overline{A \cdot B}$
0	0	1
0	1	1
1	0	1
1	1	0



A	B	$Y=\overline{A+B}$
0	0	1
0	1	0
1	0	0
1	1	0



(a) ***(b)***

Fig. C1.7 Funcții logice complexe: (a) ŞI-NU; (b) SAU-NU

Axiomele şi teoremele **algebrei booleene se prezintă în tabelul C1.1:**

Tabelul C1.1

Denumirea	Forma produs	Forma sumă
Axioma 1. **Mulțimea *B* este închisă în raport cu operatorii · şi +**	$x \in B, y \in B \Rightarrow x \cdot y \in B$	$x \in B, y \in B \Rightarrow x + y \in B$
Axioma 2.	$x \cdot 1 = x$	$x + 0 = x$
Axioma 3. **Comutativitatea**	$x \cdot y = y \cdot x$	$x + y = y + x$
Axioma 4. **Distributivitatea**	$x + y \cdot z = (x + y) \cdot (x + z)$	$x \cdot (y + z) = x \cdot y + x \cdot z$
Axioma 5.	$x \cdot \overline{x} = 0$	$x + \overline{x} = 1$
Teorema 1.	$x \cdot x = x$	$x + x = x$
Teorema 2.	$x \cdot 0 = 0$	$x + 1 = 1$
Teorema 3. **Involuția sau dubla negație**	$\overline{\overline{x}} = x$	
Teorema 4. **Asociativitatea**	$x \cdot (y \cdot z) = (x \cdot y) \cdot z$	$x + (y + z) = (x + y) + z$
Teorema 5. **DeMorgan**	$\overline{x \cdot y} = \overline{x} + \overline{y}$	$\overline{x + y} = \overline{x} \cdot \overline{y}$
Teorema 6. **Absorbția**	$x \cdot (x + y) = x$	$x + x \cdot y = x$

EXEMPLUL C1.3

Aplicați axiomele şi teoremele algebrei booleene pentru a simplifica expresia logică $A \cdot B + A \cdot C + B \cdot \overline{C}$

Răspuns

$$A \cdot B + A \cdot C + B \cdot \overline{C} = A \cdot B \cdot (\overline{C} + C) + A \cdot C + B \cdot \overline{C} =$$
$$= A \cdot B \cdot \overline{C} + (A \cdot B \cdot C + A \cdot C) + B \cdot \overline{C} = B \cdot \overline{C} \cdot (A + 1) + A \cdot C \cdot (B + 1) =$$
$$= A \cdot C + B \cdot \overline{C}$$