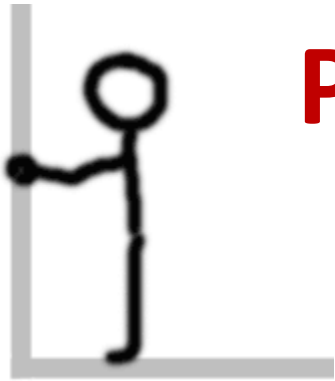
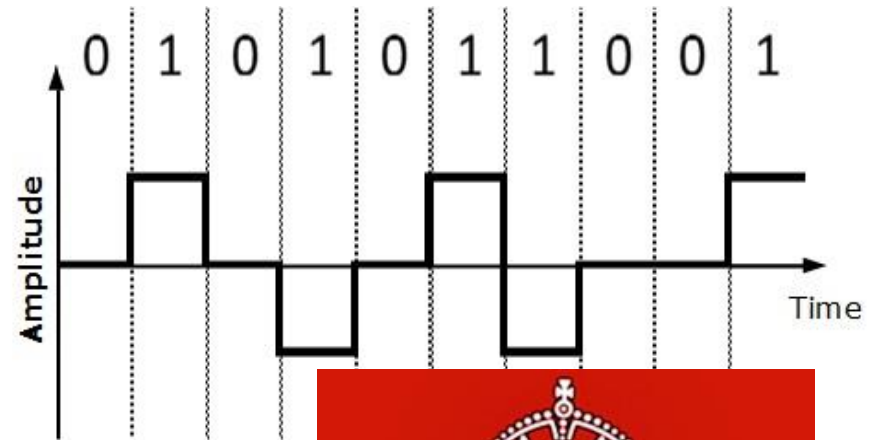
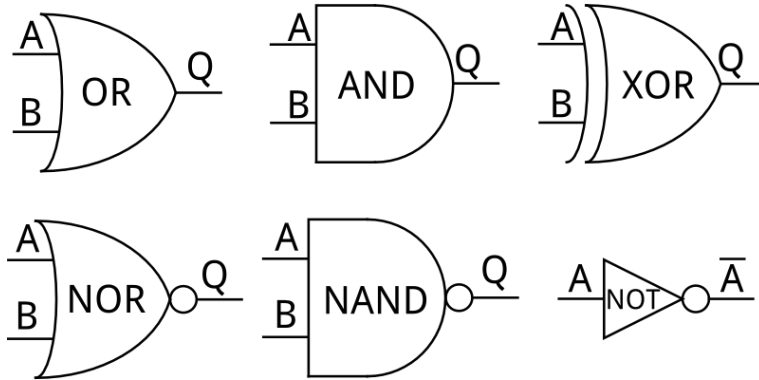




Proiectare Logica

Digital Logic Design

Curs 01



Date contact



- **Lector:** Cornel Mironel NICULAE
- **email:** cornel.nicolae@unibuc.ro
- **www:** cnic.ro/pl/
- **Logic grid (simulator logic in HTML5):**
cnic.ro/pl/lg/
- **Carte de referinta:** B. Holdsworth and R.C. Woods,
Digital Logic Design, 4th Edition, 1999

Calculul notei la *Proiectare Logica*

➤ **Nota in catalog:**

$$= \text{ROUND}(0.3 * \text{Partial} + 0.3 * \text{NotaLab} + 0.4 * \text{Examen}, 2)$$

adica

30% Partial; (se sustine in intervalul de timp alocat unui curs pe la mijlocul semestrului) **NU ESTE OBLIGATORIE** sustinerea sa.

30% Colocviu laborator [≥ 5]

Nota jurnal de laborator minus (–) o corectie de max

3p la temele curs si lab trimise in format electronic)

40% Examen [≥ 5] Daca partialul nu s-a dat acesta se va da pe subiect separat in cadrul examenului.

➤ **Prezenta la laborator obligatorie:**

- 1 lab lipsa nu afecteaza nota la colocviu
- 2 lab lipsa ➔ se scad 2 p din nota de colocviu
- 3 lab lipsa ➔ se scad 4p din nota la colocviu
- 4 lab lipsa se repeta laboratorul in anul urmator

➤ **Prezenta la curs** nu este obligatorie dar se contorizeaza. Afecteaza nota finala (in minus) cu maxim 0.5 p.

Baze de numeratie

Reprezentarea araba a numerelor

➤ Baza 10:

- $213.46_{10} = 2 \cdot 10^2 + 1 \cdot 10^1 + 3 \cdot 10^0 + 4 \cdot 10^{-1} + 6 \cdot 10^{-2}$

$$N_{10} = a_n 10^n + a_{n-1} 10^{n-1} + \dots + a_0 10^0 \\ + a_{-1} 10^{-1} + \dots + a_{-m} 10^{-m}$$

➤ Baza b:

$$N_b = a_n b^n + a_{n-1} b^{n-1} + \dots + a_0 b^0 + a_{-1} b^{-1} \\ + \dots + a_{-m} b^{-m}$$

Schimbarea bazei de numeratie

Numere intregi:

Deimpartit=Impartitor* Cat +Rest: $D = I * C + R$

➤ **Baza 10:**

- $213_{10} = 2 \cdot 10^2 + 1 \cdot 10^1 + 3$

$$\frac{213_{10}}{10} = 2 \cdot 10^1 + 1 + \frac{3}{10}$$

➔ ultima cifra a unui numar in baza 10 este restul impartirii acestui numar la 10

$$\frac{21_{10}}{10} = 2 + \frac{1}{10}$$

➔ A doua cifra se obtine calculand restul impartirii la 10 a catului obtinut anterior

Schimbarea bazei de numeratie (2)

numere intregi:

Deimpartit=Impartitor* Cat +Rest: $D = I * C + R$

Numar (Cat anterior)	Impartitor (baza)	Rest
213	10	3
21	10	1
2	10	2

Cifra cea mai putin
semnificativa



Cifra
cea mai semnificativa

Least Significant Digit : **LSD**

Most Significat Digit : **MSD**

Schimbarea bazei de numeratie (3)

Numere intregi:

Numar (Cat anterior)	Impartitor (baza)	Rest
213	2	1
106	2	0
53	2	1
26	2	0
13	2	1
6	2	0
3	2	1
1	2	1

LSB

MSB

$$213_{10} = 11010101_2$$

Schimbarea bazei de numeratie (4)

Numere fractionare: Baza 2

Numar (Cat anterior)	Partea intreaga
0.403	-
$0.403 \times 2 = 0.806$	0
$0.806 \times 2 = 1.612$	1
$0.612 \times 2 = 1.224$	1
$0.224 \times 2 = 0.448$	0
$0.448 \times 2 = 0.896$	0
$0.896 \times 2 = 1.792$	1
$0.792 \times 2 = 1.584$	1
$0.584 \times 2 = 1.168$	1


MSD

LSD

2^{-n}		Aprox
-		
$0 \times 1/2$	0.5	0
$1 \times 1/4$	0.25	0.25
$1 \times 1/8$	0.125	0.375
$0 \times 1/16$	0.0625	0.375
$0 \times 1/32$	0.03125	0.375
$1 \times 1/64$	0.015625	0.390625
$1 \times 1/128$	0.0078125	0.3984375
$1 \times 1/256$	0.00390625	0.40234375

$$0.403_{10} = 0.01100111 \dots_2$$

Schimbarea bazei de numeratie (5)

Numere fractionare:

$.265 \times 2$	$.265 \times 8$	$.265 \times 16$
0.530×2	2.120×8	4.240×16
1.060×2	0.960×8	3.840×16
0.120×2	7.680×8	$D.440 \times 16$
0.240×2	5.440×8	7.040×16
0.480	3.520	0.640
$(0.265)_{10} = (0.0100)_2$	$(0.20753)_8$	$(0.43D70)_{16}$

Bazele de numeratie 2, 8 si 16

In bazele de numeratie mai mari ca 10 se folosesc litere pentru desemnarea cifrelor mai mari de 10.

Octal	Binary
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

HD	Binary	HD	Binary
0	000	8	1000
1	001	9	1001
2	010	A	1010
3	011	B	1011
4	100	C	1100
5	101	D	1101
6	110	E	1110
7	111	F	1111

Conversii binar <-->octal si invers

$$\begin{array}{cccc} (110 & 001 & 011 & 100)_2 \\ \downarrow & \downarrow & \downarrow & \downarrow \\ = (6 & 1 & 3 & 4)_8 \end{array}$$

$$\begin{array}{cccc} (4 & 3 & 2 & .7)_8 \\ \downarrow & \downarrow & \downarrow & \downarrow \\ = (100 & 011 & 010 & .111)_2 \end{array}$$

Octal	Binary
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

Conversii binar <--> hexazecimal

In bazele de numeratie mai mari ca 10 se folosesc litere pentru desemnarea cifrelor mai mari de 10.

$$(1011 \ 1010 \ 0011 \ .0010)_2 \\ = (B \ A \ 3 \ .2)_{16}$$

$$(4 \ F \ C \ 2)_{16} \\ = (0100 \ 1111 \ 1100 \ 0010)_2$$

HD	Binary	HD	Binary
0	000	8	1000
1	001	9	1001
2	010	A	1010
3	011	B	1011
4	100	C	1100
5	101	D	1101
6	110	E	1110
7	111	F	1111

Adunarea Binara

$$\begin{array}{rcccc}
 2^3 & 2^2 & 2^1 & 2^0 & \\
 1 & 0 & 1 & 1 & 11 \\
 0 & 1 & 1 & 1 & \underline{+7} \\
 \hline
 \end{array}$$

Suma

1 0 0 1 0 18

**Biti depasire
(de transport)**

1 1 1 1

Carry bits)

Scaderea Binara – rezultat pozitiv

Diferenta
Biti de imprumut

2^3	2^2	2^1	2^0	
1	1	0	0	12
0	0	1	1	-3
1	0	0	1	+9
	1	1		



Scaderea Binara – rezultat negativ

Daca apare bit de imprumut la cel mai semnificativ bit → rezultatul este negativ.

	2^3	2^2	2^1	2^0	
	0	0	1	1	3
	1	1	0	0	-12
Diferenta	0	1	1	1	-9
Biti de imprumut	1	1			

Diagram illustrating binary subtraction (3 - 12) resulting in a negative value (-9). The subtraction is performed using 4-bit binary numbers. The minuend (3) is 0011 and the subtrahend (12) is 1100. The result is 0111, which is -9 in two's complement. The borrow bits (Biti de imprumut) are shown as 1 for the most significant bit and 1 for the second bit from the left.

Intregi cu semn

- Prin conventie s-a ales ca cel mai semnificativ bit (MSB – Most Significant Bit) sa reprezinte semnul:

0 → + si 1 → -

- $-23_{10} = 10010111_2$ $+23_{10} = 00010111_2$

- $-0_{10} = 10000000_2$ $+0_{10} = 00000000_2$

- In aceasta conventie, un registru de 8 biti, ar putea reprezenta numerele intre -127 si +127, cu deficianta ca exista 2 reprezentari ale numarului 0 (zero)

Conceptul de registru

- In aplicatiile practice, de exemplu in calculatoare, avem limitari privind numarul de cifre alocate pentru un numar.
- De exemplu, putem alege sa reprezentam numere in baza 10 pe un spatiu ce poate memora MAXIM 4 cifre zecimale.
- Acest spatiu format din 4 locuri pentru cifrele din baza 10 reprezinta un **registru de 4 cifre zecimale**. Putem memora in acest registru numere intre 0000 si 9999. Numarul **3956** poate fi reprezentat in registru:

3	9	5	6
---	---	---	---

Complementul aritmetic 1

- A fost introdus pentru reducerea numarului de operatii ce trebuiesc implementate intr-o unitate de calcul.
- **Scop initial:** transformare operatiei de scadere intr-o operatie de adunare folosind registrii de memorare cu numar fix de cifre.

De exemplu dorim sa facem operatia $17-13 = 4$ folosind registrii de 2 cifre in baza 10. Vrem sa facem $17+C(13)=104$

	1	7		+
	A	B		C(13)
1	0	4		

$7+B=14 \Rightarrow B=7$ 1 este cifra de transport

$1+A+1=10 \Rightarrow A=8$. Prin urmare $C(13)=87$

Complementul aritmetic 2

- numărul care adunat cu 17 da rezultatul corect pe ultimele 2 cifre, cu pretul apariției unei cifre de transport este complementul aritmetic. Dorim să efectuăm $17-13=4$ și facem $17+87=104$
- Complementul aritmetic al unui număr este acel număr care adunat cu primul da rezultatul corect la care s-ar adăuga, eventual, o cifră unu, pe poziția cea mai semnificativă, dacă rezultatul s-ar memora pe un registru cu o cifră în plus (adică 3 cifre în loc de 2).

Adică $17+C(13)=4+100=104 \rightarrow C(13)=104-17=87$.

Observăm că $13+C(13)=13+87=100 \rightarrow C(13)=100-13$

Pe registrii de 3 cifre $C(13)=1000-13=987$. Într-adevăr $987+17=1004 \rightarrow$ trunchiat la 3 cifre da rezultatul corect.

Complementul aritmetic 3

In baza 10:

Complementul fata de 9(10-1) pe un registru de 4 cifre.

8694 → **numar initial**

1305 → **complementul fata de 9** (suma tuturor cifrelor face 9: cel mai mare numar din 4 cifre din baza 10)

Complement fata de 10

Se obtine din complementul fata de 9 la care se adauga o unitate.

Dorim sa calculam pe reg. de **3** cifre: **127-95=32**; →

C(95)=904+1=905; 905+127=1032 → Trunchiat la 3 cifre

Complementul aritmetic 3

- In fiecare baza de numeratie exista 2 complementi:
 - Complementul fata de **cea mai mare cifra** (De ex. fata de 9 in baza 10) dintre cifrele $0,1,\dots,9$
 - Complementul fata de **baza de numeratie** (De ex. fata de 10 in baza 10)
- In baza 2 vor exista 2 complementi:
 - Complementul fata de 1 si
 - Complementul fata de 2.
- **Ce complementi vor exista in baza 16?**

Complementii aritmetici in baza 2

➤ **Complementul fata de 1** inseamna **inversarea** bitilor
(0 $\leftrightarrow$ 1)

- Ex: $C1(1010) = 0101$, in cazul unui **registru de 4 biti**

➤ **Complementul fata de 2**

- **Prima metoda:** $C2(1010) = C1(1010) + 1 = 0110$
- **A 2-a metoda:** **a)** se pastreaza bitii de la dreapta celui mai putin semnificativ bit = 1 si **b)** se inverseaza ceilalti.
 - Ex.1. **1010** → **0110**
 - Ex.2. **0100** → **1100**
 - Ex.3. **101000** → **011000**

Reprezentarea binara a numerelor negative folosind C2

➤ Reprezentarea numerelor negative o constituie complementul fata de 2 a numarului pozitiv corespunzator.

- De ex. folosind registrii de 8 biti $19_{10} = 00010011_2$
- $-19_{10} = C2(19) = 11101101_2$
- Verificare:

$$\begin{array}{r} 00010011 + \\ \underline{11101101} \\ 10000000 \end{array} \quad \begin{array}{r} +19 \\ -19 \\ 0 \end{array}$$

- Bitul de semn al numerelor negative este 1

➤ In aceasta reprezentare, un registru cu 8 biti poate memora numere intregi intre -128 si $+127$. Intr-adevar $-128_{10} = 10000000_2$; $+127_{10} = 01111111_2$; $-1_{10} = 11111111_2$. Verificarea se face prin adunarea:

$$\begin{array}{r} 10000000_2 + \\ 01111111_2 \\ \underline{11111111_2} \\ 11111111_2 \end{array} \quad \begin{array}{r} -128_{10} \\ +127_{10} \\ -1_{10} \end{array}$$

Validitatea Complementului fata de 2

- Complementele fata de 2 si 1 ale numarului X , se mai noteaza cu $[X]_2$ si respectiv $[X]_1$ (notat uneori si cu $\bar{X}$).
- Daca ne referim la **registrii pe n biti**, din cele prezentate anterior rezulta ca:

$$[X]_2 = 2^n - X \text{ si } [X]_1 + 1 = [X]_2.$$

Prin urmare daca dorim sa facem diferenta $Y - X$ atunci calculam $Y + [X]_2 - 2^n$.

Stim ca $X - X = 0$. Intr-o reprezentare pe n biti

$$0 = X - X = X + [X]_2 - 2^n = X + [X]_1 + 1 - 2^n = X + \bar{X} + 1 - 2^n.$$

Prin urmare ar trebui ca $X + \bar{X} + 1 = 2^n$.

Caz particular:

1010	X
0101	$\bar{X}$
<u> 1 </u>	
1000	

Reprezentare in C2

Pe **un registru de 4 biti** avem in aceasta reprezentare urmatoarele numere binare:

Nr zecimal	Nr. binar in C2	Nr zecimal	Nr. binar in C2
-1	1111	7	0111
-2	1110	6	0110
-3	1101	5	0101
-4	1100	4	0100
-5	1011	3	0011
-6	1010	2	0010
-7	1001	1	0001
-8	1000	0	0000

Reprezentarea Offset Binary

La convertoarele Analog digitale mai intalnim si aceasta reprezentarea Offset Binary (OB):

Nr zecimal	Nr. binar in OB	Nr zecimal	Nr. binar in OB
7	1111	-1	0111
6	1110	-2	0110
5	1101	-3	0101
4	1100	-4	0100
3	1011	-5	0011
2	1010	-6	0010
1	1001	-7	0001
0	1000	-8	0000

Formula Pentru Offset Binary

Pe un registru de n biti

$$\text{OB}(X,n)=\text{uint}(X,n) + \text{uint}(2^{n-1},n)$$

Exemplu numarul 120 pe un registru de 8 biti:

$$\begin{aligned}\text{OB}(120,8) &= 0111\ 1000_2 + 0111\ 1111_2 \\ &= 1111\ 0111_2\end{aligned}$$

Inmultirea numerelor intregi in binar

- Inmultirea numerelor intregi pozitive in reprezentare binara este analoga inmultirii in baza 10.

1011		11 ×
1101		13
1011	} Produce Partiale	
0000		
1011		
1011		
10001111		143

Reprezentari binare pentru numerele zecimale

Exista mai multe reprezentari binare pentru cifrele zecimale (0-9).

NBCD –Naturally **Binary Coded Decimal**

Decimal digit	NBCD 8,4,2,1	BCD 7,4,2,1	BCD 2,4,2,1	BCD 8,4,-2,-1	Excess 3 XS3
0	0000	0000	0000	0000	0011
1	0001	0001	0001	0111	0100
2	0010	0010	0010	0110	0101
3	0011	0011	0011	0101	0110
4	0100	0100	0100	0100	0111
5	0101	0101	1011	1011	1000
6	0110	0110	1100	1010	1001
7	0111	1000	1101	1001	1010
8	1000	1001	1110	1000	1011
9	1001	1010	1111	1111	1100