

Logică digitală

-Curs 3-

REPREZENTAREA
DATELOR ÎN SISTEME
DE CALCUL

- 2021-

Reprezentarea numerelor în sistemele de calcul

- ❑ Sisteme de numerație poziționale (binar, octal, hexazecimal);
 - ❑ Reprezentarea numerelor în virgulă fixă (SM, C1, C2);
 - ❑ Reprezentarea numerelor de virgulă flotantă;
 - ❑ Coduri binare pentru numere zecimale;
-

Sisteme de numere poziționale

- **Sistem pozițional** - un număr este reprezentat printr-un șir de cifre, unde pt. fiecare poziție a unei cifre este asociată o anumită greutate.

- Valoarea numărului este o sumă a cifrelor înmulțită cu greutatea aferentă:

$$\text{Ex1: } 1734 = 1 \cdot 10^3 + 7 \cdot 10^2 + 3 \cdot 10^1 + 4 \cdot 10^0$$

- Fiecare greutate e o putere a lui 10 corespunzătoare poziției numărului. Pentru numere cu virgulă folosim puteri negative a lui 10.

$$\text{Ex2: } 5186.67 = 5 \cdot 10^3 + 1 \cdot 10^2 + 8 \cdot 10^1 + 6 \cdot 10^0 + 6 \cdot 10^{-1} + 7 \cdot 10^{-2}$$

Binar, octal, hexazecimal ...

Zecimal	Binar	Octal	Hexazecimal
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

Reprezentarea numerelor în virgulă fixă

- Pentru reprezentarea **numerelor cu semn**, se alocă bitul cel mai semnificativ (cel mai din stânga – most significant bit - MSB) semnului:
 - „0” - numere pozitive,
 - „1” - numere negative.
- Uzual în sistemele de calcul se operează fie cu numere întregi, fie cu numere fracționare;
- Din acest motiv, poziția virgulei este considerată implicit după cum urmează:
 - numere întregi poziția virgulei este la dreapta bitului cel mai puțin semnificativ: $b_{n-1}b_{n-2}...b_1b_0.$
 - numere fracționare poziția virgulei este la dreapta bitului cel mai semnificativ, care este și bitul de semn:

$$b_0.b_{-1}...b_{-n+1}b_{-n}$$

Reprezentarea numerelor în virgulă fixă

- Pentru reprezentarea numerelor cu semn există trei formate de reprezentare:
 - semn-mărime - **SM**,
 - complement de 1 - **C1**,
 - complement de 2 - **C2**.
-

Reprezentarea numerelor în virgulă fixă – C2

$$\begin{array}{cccccccc} + & 2 & = & 0 & & 0 & 1 & 0 & + \\ + & 3 & = & 0 & & 0 & 1 & 1 & \\ \hline & & & 0 & & 1 & 0 & 1 & = + 5 \end{array}$$

a.

$$\begin{array}{rcccccccl} - & 2 & = & 1 & 1 & 1 & 0 & + \\ + & 3 & = & 0 & 0 & 1 & 1 & \\ \hline & & & 0 & 0 & 0 & 1 & = + 1 \end{array}$$

c.

$$\begin{array}{r} - \quad 2 \quad = \quad 1 \quad 1 \quad 1 \quad 0 \quad + \\ - \quad 3 \quad = \quad 1 \quad 1 \quad 0 \quad 1 \\ \hline 1 \quad 0 \quad 1 \quad 1 \quad = \quad - \quad 5 \end{array}$$

$$\begin{array}{ccccccccc} + & 2 & = & 0 & & 0 & 1 & 0 & + \\ - & 3 & = & 1 & & 1 & 0 & 1 & \\ \hline & & & 1 & & 1 & 1 & 1 & = - 1 \end{array}$$

d.

- Domeniul valoric pentru numere întregi:
- 2^{n-1} și $2^{n-1} - 1$

Reprezentarea numerelor în virgulă flotantă (mobilă)

- reprezentate folosind notația științifică (care nu este pozițională) → un domeniu valoric foarte mare.
- Pentru a reprezenta un număr în virgulă flotantă folosim trei numere conform relației:

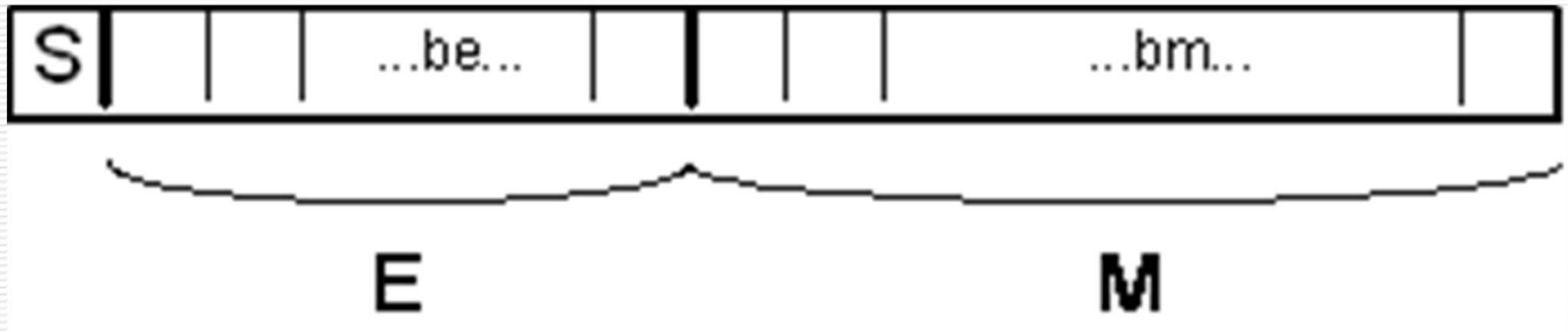
$$N = M * B^E$$

M - mantisa. (M poate fi reprezentată în SM sau C2)

B - baza (de obicei e 2 sau o putere a lui 2)

E - exponent. (E este reprezentat în SM sau cod exces)

Reprezentarea numerelor în virgulă flotantă



M - mantisa. (*M* poate fi reprezentată în SM sau C2)

B - baza (de obicei e 2 sau o putere a lui 2)

E - exponent. (*E* este reprezentat în SM sau cod exces)

Reprezentarea numerelor în virgulă flotantă

□ Reprezentarea mantisei:

■ Reprezentarea lui 18:

$$18 * 10^0 = 1.8 * 10^1 = 0.18 * 10^2 = \dots = 0.0\dots018 * 10^n$$

■ *Obs.:* M , B și E au o infinitate de valori posibile

■ Pentru o tratare unitară și eficientă prin prisma procesării în sistemele de calcul → o reprezentare unică → normalizarea mantisei M

Reprezentarea numerelor în virgulă flotantă – Normalizarea mantisei

- M în SM primul bit din dreapta virgulei trebuie să fie 1.
 - M în C2 și M corespunde la o:
 - valoare negativă atunci primul bit din dreapta virgulei trebuie să fie 0.
 - valoare pozitivă, atunci folosim regula de la SM.
-

Reprezentarea numerelor în virgulă flotantă

☐ Reprezentarea exponentului:

- Problema reprezentării lui 0 în virgulă mobilă.

$$\mathbf{0} = 0 * B^E$$

- ... mai multe variante de reprezentare
 - zero trebuie să fie cât mai ușor de detectat și testat → (?) șir de biți de '0'
 - Dar... în calcule recurgem la aproximări
 - ☐ este posibil ca în urma unor calcule (FP), datorită acestor aproximări successive, să obținem în loc de 0 un număr foarte mic ($M \neq 0$).
-

Reprezentarea numerelor în virgulă flotantă

- ❑ Pentru a minimiza eroarea → exponentul aferent lui 0 să fie cel mai mic posibil.
 - ❑ valoarea min. a oricărui exponent să fie 0.
 - ❑ Toate valorile negative reprezentabile pe N biți sunt deplasate (devin pozitive) prin adunarea unui bias (unui surplus) = valoarea absolută a celui mai mic număr reprezentabil pe N numărul de biți exponent.
 - ❑ Pentru exponent reprezentat în:
 - SM pe 8 biți → valoarea bias-ului este 127
 - C2 pe 8 biți → valoarea bias-ului este 128
-

Reprezentarea numerelor în virgulă flotantă – reprezentarea exp.

Reprezentare binară	Valoare fără semn	Valoare cu semn (reală-cea a numărului reprezentat)	
		Bias = 127	Bias = 128
11111111	255	+128	+127
11111110	254	+127	+126
.	.	.	.
.	.	.	.
.	.	.	.
10000001	129	+2	+1
10000000	128	+1	0
01111111	127	0	-1
01111110	126	-1	-2
.	.	.	.
.	.	.	.
.	.	.	.
00000001	1	-126	-127
00000000	0	-127	-128

Reprezentarea numerelor în virgulă flotantă

□ Cele mai reprezentative standarde pentru virgula mobilă sunt:

■ IEEE 754

■ IBM S360/370

Reprezentarea numerelor în virgulă flotantă

- Standardul IEEE 754/2008- formate:
 - Half precision
 - Simple precision
 - Double precision
 - Double-extended
-

Reprezentarea numerelor în virgulă flotantă: IEEE 754

☐ Caracteristici:

- E și M – format SM
 - Exponentul este reprezentat în exces de:
 - ☐ 127 pentru simplă precizie
 - ☐ 1023 pentru dublă precizie.
 - *Hidden bit.*
 - ☐ Mantisa are un bit de 1 ascuns.
 - ☐ bitul la dreapta virgulei care trebuie să fie 1 (din condiția de normalizare).
 - ☐ (S.**1**M) (unde S este semnul iar M este mantisa)
→ virgula a fost mutată la dreapta bit-ului de 1
cel mai semnificativ: S**1**.M
-

Reprezentarea numerelor în virgulă flotantă: IEEE 754 simplă precizie

- Simplă precizie: 32 biți
 - 1 bit de semn
 - 8 biți de exponent; exponent reprezentat în exces de 127
 - 23 biți de mantisă (significand)
- Formatul de număr este:



Ex.: Să se reprezinte în format IEEE 754 SP numărul 4.625

- Pasul 1: Se convertește numărul în baza 2.

$$4.625 = 100.101 * 2^0$$

- Pasul 2: Normalizare $\rightarrow 1.M$ (mutarea virgulei cu ajustarea corespunzătoare a puterii lui 2)

$$100.101 = 1.00101 * 2^2$$

$$E = 2 + 127 (\text{exces}) = 129 = 128 + 1 = 2^7 + 2^0 = 10000001$$

- Pasul 3:

S	E (8 biți)	M (23 biți)
0	10000001	0010 1000 0000 0000 0000 000

Reprezentarea numerelor în virgulă flotantă: IEEE 754 dublă precizie

□ Reprezentare pe 64 de biți:

- 1 bit de semn
- 11 biți de exponent reprezentați în exces de 1023
- 52 biți de mantisă (significand)



Reprezentarea numerelor în virgulă flotantă: IEEE 754 valori speciale

Nr.	Exponent (E)	Mantisa (M)	Valoare speciala
1.	0	0	± 0
2.	0	$\neq 0$	Denormalized numbers
3.	255	0	$\pm \infty$
4.	255	$\neq 0$	NaN

- Nr. denormalizate: rezultat care este mai mic decât valoarea minimă reprezentabilă
- Infinit: situația în care rezultatul intermediar este infinit sau avem overflow
- 0/0 sau radical din nr. negativ

Coduri binare pentru numere zecimale

<p>(!)</p> <p>Cod (code),</p>	<p>O colecție de șiruri diferite pe n biți, iar fiecare dintre aceste șiruri are o semnificație (reprezintă un număr, caracter, etc.) poartă denumirea de <i>cod</i> (code). Numărul maxim de cuvinte ale unui cod pe n biți este 2^n. Nu întotdeauna însă, toate aceste combinații posibile pe n biți sunt folosite (fac parte din colecția de șiruri care alcătuiesc codul).</p>
<p>Cuvânt al unui cod (code word)</p>	<p>Un șir al colecției care reprezintă o combinație de n valori de 0 sau 1 se numește cuvânt de cod (code word).</p>

Coduri binare pentru numere zecimale - BCD

- ❑ există situații când se dorește afișarea rezultatelor de către interfețele externe ale dispozitivului de calcul într-un format ușor de înțeles (decodificat) de către utilizator – și anume mult întrebuițatul format zecimal;
 - ❑ cel mai la îndemână cod zecimal este BCD (binary-code decimal):
 - reprezentarea unei cifre BCD → înlocuirea cu reprezentarea în binar care îi corespunde → cu un nr. pe 4 biți
-

Coduri binare pentru numere zecimale - BCD

Cifră zecimală	Correspondent BCD
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

Coduri binare pentru numere zecimale - BCD

- conversia unui număr zecimal în BCD prin înlocuirea succesivă a cifrelor zecimale cu tetradele corespunzătoare

$$N_1 = 4\ 5\ 9_{10} = 0100\ 0101\ 1001_{BCD}$$

- operația inversă de transformare a unui număr reprezentat în BCD în omologul zecimal

$$N_2 = 1000\ 0111\ 0000\ 0010_{BCD} = 8\ 7\ 0\ 2_{10}$$

Coduri binare pentru numere zecimale - BCD

□ Avantaje:

- simplitate
- este un cod pozițional, ponderea fiecărei cifre fiind $10^j \cdot 2^i$, unde j reprezintă poziția tetrazei zecimale, iar i poziția bitului în cadrul tetrazei

□ Dezavantaje:

- utilizează un număr mai mare de biți față de reprezentarea binară a numărului respectiv;
 - Fol. 3 cifre BCD - 12 biți (3 cifre zecimale * 4 biți/cifră) se pot reprezenta $10^3 = 1000$ de numere;
 - folosind reprezentarea binară, pe 12 biți se pot reprezenta $2^{12} = 4096$ de numere;
 - implementare anevoioasă a operației de adunare
-

Coduri binare pentru numere zecimale – Exces de 3

- Exces de 3 se obține din codul BCD astfel:
 - la fiecare cifră reprezentată în cod BCD se adună valoarea 3 (0011 în binar).
 - fiecare cifră zecimală se reprezintă cu ajutorul unei combinații de 4 biți (o tetradă de biți)
-

Coduri binare pentru numere zecimale – Exces de 3

Cifră zecimală	Corespondent exces de 3
0	0011
1	0100
2	0101
3	0110
4	0111
5	1000
6	1001
7	1010
8	1011
9	1100

$b_3b_2b_1b_0$	$b_6b_5b_4$									
	000	001	010	011	100	101	110	111		
0000	NUL	DLE	SP	0	@	P	'	p	NUL	Null
0001	SOH	DC1	!	1	A	Q	a	q	SOH	Start of heading
0010	STX	DC2	"	2	B	R	b	r	STX	Start of text
0011	ETX	DC3	#	3	C	S	c	s	ETX	End of text
0100	EOT	DC4	\$	4	D	T	d	t	EOT	End of transmission
0101	ENQ	NAK	%	5	E	U	e	u	ENQ	Enquiry
0110	ACK	SYN	&	6	F	V	f	v	ACK	Acknowledge
0111	BEL	ETB	'	7	G	W	g	w	BEL	Bell
1000	BS	CAN	(8	H	X	h	x	BS	Backspace
1001	HT	EM)	9	I	Y	i	y	HT	Horizontal tab
1010	LF	SUB	*	:	J	Z	j	z	LF	Line feed
1011	VT	ESC	+	;	K	[k	{	VT	Vertical tab
1100	FF	FS	,	<	L	\	l		FF	Form feed
1101	CR	GS	-	=	M]	m	}	CR	Carriage return
1110	SO	RS	.	>	N	^	n	~	SO	Shift out
1111	SI	US	/	?	O	_	o	DEL	SI	Shift in
									SP	Space
									DLE	Data link escape
									DC1	Device control 1
									DC2	Device control 2
									DC3	Device control 3
									DC4	Device control 4
									NAK	Negative acknowledgement
									SYN	Synchronize
									ETB	End transmission block
									CAN	Cancel
									EM	End of medium
									SUB	Substitute
									ESC	Escape
									FS	File separator
									GS	Group separator
									RS	Record separator
									US	Unit separator
									DEL	Delete or rubout

American Standard Code for Information Interchange (ASCII)

$b_3b_2b_1b_0$	$b_6b_5b_4$									
	000	001	010	011	100	101	110	111		
0000	NUL	DLE	SP	0	@	P	'	p		NUL
0001	SOH	DC1	!	1	A	Q	a	q		SOH
0010	STX	DC2	"	2	B	R	b	r		STX
0011										ETX
0100										EOT
0101										ENQ
0110	ACK	SYN	&	6	F	V	f	v		ACK
0111	BEL	ETB	,	7	G	W	g	w		BEL
1000	BS	CAN	(8	H	X	h	x		BS
1001	HT	EM)	9	I	Y	i	y		HT
1010	LF	SUB	*	:	J	Z	j	z		LF
1011	VT	ESC	+	;	K	[k	{		VT
1100	FF	FS	,	<	L	\	l			FF
1101	CR	GS	-	=	M]	m	}		CR
1110	SO	RS	.	>	N	^	n	~		SO
1111	SI	US	/	?	O	_	o	DEL		SI

litera **A** are primele 3 pozitii (765) secventa 100, iar pe urmatoarele 4 pozitii (4321) secventa 0001. Deci A=(1000001) !

NUL Null
 SOH Start of heading
 STX Start of text
 ETX End of text
 EOT End of transmission
 ENQ Enquiry
 ACK Acknowledge
 BEL Bell
 BS Backspace
 HT Horizontal tab
 SP Space
 DLE Data link escape
 DC1 Device control 1
 DC2 Device control 2
 DC3 Device control 3
 DC4 Device control 4
 NAK Negative acknowledgement
 SYN Synchronize
 ETB End transmission block
 CAN Cancel
 EM End of medium
 SUB Substitute
 ESC Escape
 FS File separator
 GS Group separator
 RS Record separator
 US Unit separator
 DEL Delete or rubout

American Standard Code for Information Interchange (ASCII)

Întrebări?

**Enough Talking Let's Get To It
!!Brace Yourselves!!**

