



Bevezetés a számítástechnikába

#02 – Adatábrázolás és logikai áramkörök

2023. szeptember 18–22.

Siklósi Bálint <siklosi.balint@itk.ppke.hu>
Naszlady Márton Bese <naszlady@itk.ppke.hu>

#02/1 – Számábrázolás

Számrendszerek

- ▶ Numeral system (nem numeric)
- ▶ A matematikai fogalom írott formában történő megjelenítésére használható módszer
- ▶ Helyiértéken alapuló számrendszerek
- ▶ Más, pl. sorrendiségen alapuló rendszer (római)

Számrendszer alapja és a számjegyek

- ▶ Fő paraméterek:
 - ▶ Számrendszer alapja
 - ▶ Pozíciókba írható számjegyek

Alap	Számjegyek	Jelölés
10	0,1,2,3,4,5,6,7,8,9	$27_{[10]}$, 27

Számrendszer alapja és a számjegyek

► Fő paraméterek:

- Számrendszer alapja
- Pozíciókba írható számjegyek

Alap	Számjegyek	Jelölés
10	0,1,2,3,4,5,6,7,8,9	$27_{[10]}$, 27
8	0,1,2,3,4,5,6,7	$33_{[8]}$, 033

Számrendszer alapja és a számjegyek

► Fő paraméterek:

- Számrendszer alapja
- Pozíciókba írható számjegyek

Alap	Számjegyek	Jelölés
10	0,1,2,3,4,5,6,7,8,9	$27_{[10]}$, 27
8	0,1,2,3,4,5,6,7	$33_{[8]}$, 033
2	0,1	$11011_{[2]}$, 11011 b

Számrendszer alapja és a számjegyek

- ▶ Fő paraméterek:
 - ▶ Számrendszer alapja
 - ▶ Pozíciókba írható számjegyek

Alap	Számjegyek	Jelölés
10	0,1,2,3,4,5,6,7,8,9	$27_{[10]}$, 27
8	0,1,2,3,4,5,6,7	$33_{[8]}$, 033
2	0,1	$11011_{[2]}$, 11011 b
16	0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F	$1B_{[16]}$, 0x1B

Számrendszer alapja és a számjegyek

- ▶ Fő paraméterek:
 - ▶ Számrendszer alapja
 - ▶ Pozíciókba írható számjegyek

Alap	Számjegyek	Jelölés
10	0,1,2,3,4,5,6,7,8,9	$27_{[10]}$, 27
8	0,1,2,3,4,5,6,7	$33_{[8]}$, 033
2	0,1	$11011_{[2]}$, 11011 b
16	0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F	$1B_{[16]}$, 0x1B
A	0,1,...,A-1	$???_A$

Szám értéke

A szám értékét a számjegyek értéke és a helyiérték adja meg:

$$x = (a_n \cdot A^n) + (a_{n-1} \cdot A^{n-1}) + \dots + (a_1 \cdot A^1) + (a_0 \cdot A^0)$$

Matematikailag „szebb” jelöléssel:

$$x = \sum_{k=0}^n a_k \cdot A^k$$

ahol:

- ▶ x az ábrázolt szám értéke
- ▶ n a számjegyek száma
- ▶ a_k a k -adik helyiértéken lévő számjegy
- ▶ A a számrendszer alapja

Fun fact: faktoriális számrendszer

- ▶ vegyes alapú számrendszer
- ▶ kiértékelés szabálya:

$$x = \sum_{k=0}^n a_k * k!$$

- ▶ pl. $719_{[10]} = 5 * 5! + 4 * 4! + 3 * 3! + 2 * 2! + 1 * 1! + 0 * 0! = 543210_!$
- ▶ pl. $5_{[10]} = 210_!$
- ▶ pl. $3628799_{[10]} = 9876543210_!$

Mértékegységek

- ▶ bit (**binary digit**): legkisebb egység, értéke 0 vagy 1
 - ▶ byte: 8 bit
 - ▶ prefixumok (1000-esével): k (kilo), M (mega), G (giga), T (tera), P (Peta)
 - ▶ prefixumok (1024-esével): Ki (kibi), Mi (mebi), Gi (gibi), Ti (tebi), Pi (pebi)
-
- ▶ $1 \text{ TB} = 1\,000 \text{ GB} = 1\,000\,000\,000 \text{ byte} = 0,909 \text{ TiB}$

#02/2 – Számító áramkörök

Bináris összeadás

Adjuk össze az alábbi két számot: 4 és 6. Az eredmény természetesen 10.

Binárisan: $100_2 + 110_2 = 1010_2$

Hogyan kell ezt a bináris összeadást helyiértékenként elvégezni?

Összeadó áramkörök

A bináris összeadás igazságtáblája:

A	B	S
0	0	0
0	1	1
1	0	1
1	1	0

Összeadó áramkörök

A bináris összeadás igazságtáblája:

A	B	S
0	0	0
0	1	1
1	0	1
1	1	0

Ez pontosan a XOR művelet!

Összeadó áramkörök

A bináris összeadás igazságtáblája:

A	B	S
0	0	0
0	1	1
1	0	1
1	1	0

Ez pontosan a XOR művelet!

Viszont mi a helyzet a továbbviendő maradékkal?

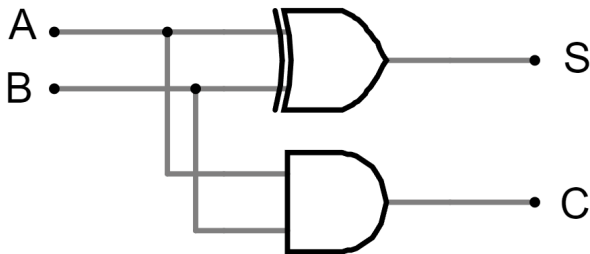
Összeadó áramkörök

A bináris összeadás igazságtáblája, maradékkal:

A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Összeadó áramkörök

A bináris, maradékos összeadást generáló áramkör neve **half-adder**.



Mi van akkor, ha már a bemenetkor is figyelembe vennénk a maradékot?

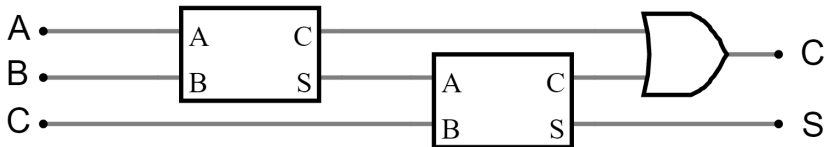
Összeadó áramkörök

A hozott maradékot is figyelembe vevő összeadás igazságtáblája:

C	A	B	S	C
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

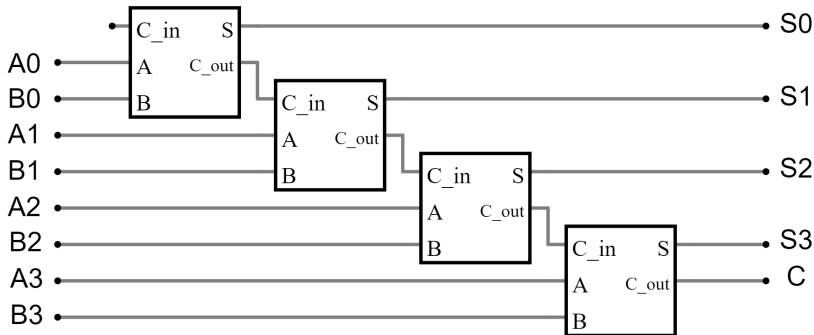
Összeadó áramkörök

A hozott maradékot is figyelembe vevő összeadó áramkör neve **full-adder**.



Összeadó áramkörök

Full-adderek egymás után kapcsolásával több helyiértékes bináris összeadó áramkört is építhetünk:



#02/3 – Bonyolultabb adatok ábrázolása

Nemnegatív egész számok (unsigned integer)

- ▶ Bináris ábrázolás, olvasás miatt hexadecimális
- ▶ Fix hosszon (általában n -szer 8 bites egységekben (8, 16, 32, 64 bites számok))

Kivonás és negatív számok

Vonjuk ki a 4-es számból a 6-ot. Az eredmény természetesen -2 .

Binárisan: $100_2 - 110_2 = -10_2$

Hogyan kell ezt a bináris kivonást helyiértékenként elvégezni?
Egyáltalán, hogyan lehet a negatív számokat binárisan ábrázolni?

Negatív számábrázolás

- ▶ Előjelbites ábrázolás
 - ▶ Első bit: előjel. $0 \rightarrow +$, $1 \rightarrow -$
 - ▶ *nulla* = 00000000 (+0) vagy 10000000 (−0)

Negatív számábrázolás

- ▶ Előjelbites ábrázolás
 - ▶ Első bit: előjel. $0 \rightarrow +, 1 \rightarrow -$
 - ▶ *nulla* = 00000000 (+0) vagy 10000000 (−0)
- ▶ Kettes komplementum ábrázolás – signed integer
 - ▶ $\neg |neg.szám + 1|$
 - ▶ egy *nulla* van
 - ▶ +, − könnyen elvégezhető
 - ▶ konvertálás nagyobb méretre: az első bitet ismétljük

Negatív számábrázolás

- ▶ Előjelbites ábrázolás
 - ▶ Első bit: előjel. $0 \rightarrow +, 1 \rightarrow -$
 - ▶ *nulla* = 00000000 (+0) vagy 10000000 (−0)
- ▶ Kettes komplementum ábrázolás – signed integer
 - ▶ $\neg |neg.szám + 1|$
 - ▶ egy *nulla* van
 - ▶ +, − könnyen elvégezhető
 - ▶ konvertálás nagyobb méretre: az első bitet ismételjük
- ▶ Eltolás ábrázolás – excess-M
 - ▶ Eltoljuk az ábrázolandó értékeket *M*-mel

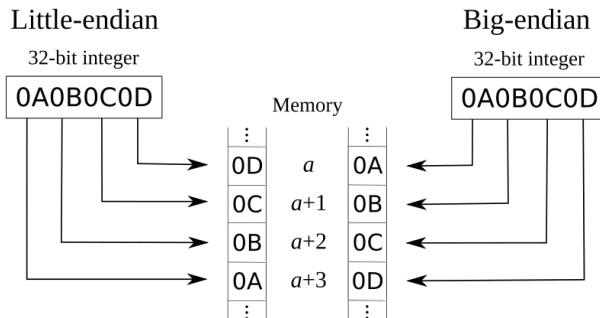
Egész számok ábrázolási határai és pontossága

- ▶ Ábrázolható számok mennyisége N biten: 2^N
- ▶ Ábrázolási határok N biten:

Ábrázolás	legkisebb szám	legnagyobb szám
Előjel nélküli egész	0	$2^N - 1$
Kettes komplement	-2^{N-1}	$2^{N-1} - 1$
Eltolt-M	$-M$	$-M + 2^N - 1$

- ▶ Túlcsordulás
 - ▶ levágás (Fun fact: km óra)
 - ▶ szaturáció (A legnagyobb/kisebb elemet tartjuk meg)
- ▶ Fun fact: Y2K probléma

Bytesorrend (Endianness)



Például:

- ▶ német számok – *huszon_öt* = *Fünf-und-zwanzig*
- ▶ dátumok – *2023. szeptember 14.* = *14th September 2023*
- ▶ adatkommunikációban (melyik byte-ot küldöm át előbb?)

Karakterek és kódolásuk

- ▶ Karakter: ABC betűi, számok, írásjelek (szóköz is!), vezérlő karakterek stb...
- ▶ Karakterkészlet: Karakterek kiválasztott csoportja. (pl. nyelv, régió alapján)
- ▶ Karakterek kódolása: A "karakter" fogalomhoz rendeljünk valamit, amivel tárolni és megjeleníteni tudjuk azt
 - ▶ Morze-kód: karakter → rövid/hosszú jelek, pl. rádió
 - ▶ Braille-kód: karakter → 3D objektumok
 - ▶ "Informatika": karakter → szám ⇒ kódtáblák

ASCII kódtábla

ASCII Code Chart

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2		!	"	#	\$	%	&	'	()	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{		}	~	DEL

Latin1 - ISO 8859-1 kódtábla

Codepage 819 - Latin 1 - ISO 8859-1

	-0	-1	-2	-3	-4	-5	-6	-7	-8	-9	-A	-B	-C	-D	-E	-F
0-		0001	0002	0003	0004	0005	0006	0007	0008	0009	000A	000B	000C	000D	000E	000F
1-	0010	0011	0012	0013	0014	0015	0016	0017	0018	0019	001A	001B	001C	001D	001E	001F
2-	0020	0021	0022	0023	0024	0025	0026	0027	0028	0029	002A	002B	002C	002D	002E	002F
3-	0030	0031	0032	0033	0034	0035	0036	0037	0038	0039	003A	003B	003C	003D	003E	003F
4-	0040	0041	0042	0043	0044	0045	0046	0047	0048	0049	004A	004B	004C	004D	004E	004F
5-	0050	0051	0052	0053	0054	0055	0056	0057	0058	0059	005A	005B	005C	005D	005E	005F
6-	0060	0061	0062	0063	0064	0065	0066	0067	0068	0069	006A	006B	006C	006D	006E	006F
7-	0070	0071	0072	0073	0074	0075	0076	0077	0078	0079	007A	007B	007C	007D	007E	007F
8-	0080	0081	0082	0083	0084	0085	0086	0087	0088	0089	008A	008B	008C	008D	008E	008F
9-	0090	0091	0092	0093	0094	0095	0096	0097	0098	0099	009A	009B	009C	009D	009E	009F
A-	00A0	00A1	00A2	00A3	00A4	00A5	00A6	00A7	00A8	00A9	00AA	00AB	00AC	00AD	00AE	00AF
B-	00B0	00B1	00B2	00B3	00B4	00B5	00B6	00B7	00B8	00B9	00BA	00BB	00BC	00BD	00BE	00BF
C-	00C0	00C1	00C2	00C3	00C4	00C5	00C6	00C7	00C8	00C9	00CA	00CB	00CC	00CD	00CE	00CF
D-	00D0	00D1	00D2	00D3	00D4	00D5	00D6	00D7	00D8	00D9	00DA	00DB	00DC	00DD	00DE	00DF
E-	00E0	00E1	00E2	00E3	00E4	00E5	00E6	00E7	00E8	00E9	00EA	00EB	00EC	00ED	00EE	00EF
F-	00F0	00F1	00F2	00F3	00F4	00F5	00F6	00F7	00F8	00F9	00FA	00FB	00FC	00FD	00FE	00FF

Fun fact: UTF8

Character		Code point		UTF-8		
		Octal	Binary	Binary	Octal	Hexadecimal
\$	U+0024	044	010 0100	00100100	044	24
¢	U+00A2	0242	000 1010 0010	11000010 10100010	302 242	C2 A2
₹	U+0939	004471	0000 1001 0011 1001	11100000 10100100 10111001	340 244 271	E0 A4 B9
€	U+20AC	020254	0010 0000 1010 1100	11100010 10000010 10101100	342 202 254	E2 82 AC
©	U+10348	0201510	0 0001 0000 0011 0100 1000	11110000 10010000 10001101 10001000	360 220 215 210	F0 90 8D 88

Mit tanultunk?

1100001110000001 =?

Mit tanultunk?

1100001110000001 =?

- ▶ Előjel nélküli egész 2 byte-on: 50049
- ▶ Előjel nélküli egész 1 byte-on: 195; 129
- ▶ Előjeles egész 2 byte-on: -17281
- ▶ Kettes komplement 2 byte-on: -15487
- ▶ 128-as eltolás 1 byte-on: 67; 1
- ▶ Lebegőpontos ábrázolás. előjel, 10 bit mantissza, 5 bit karakterisztika 15-ös eltolással: -0.032958984375
- ▶ ASCII kódolással: nem értelmezhető
- ▶ Latin-1 kódolással: Ā_i
- ▶ UTF-8 kódolással: Á

De akár:

- ▶ 16-bit high color: RGB(24,28,1)
- ▶ sakklépés: sötét gyalog lép a 14-es mezőről az 1-esre.

Szorgalmi feladat

Töltsd le az alábbi bináris adatot és kódold ki a *megfelelő* kódolás szerint az egyes sorait.

A különböző sorokban különböző kódrendszerek szerint van tárolva az adat.

Milyen kódolás szerint és mit ábrázol a 3. és 4. sor?

`https://moodle.ppke.hu/pluginfile.php/86072/mod_page/content/2/szorgalmi.txt`

VÉGE



PÁZMÁNY

Pázmány Péter Katolikus Egyetem
Információs Technológiai és Bionikai Kar