Budapesti Műszaki és Gazdaságtudományi Egyetem
Villamosmérnöki és Informatikai Kar
Méréstechnika és Információs Rendszerek Tanszék

Digitális technika
VIMIAA01

Fehér Béla
BME MIT

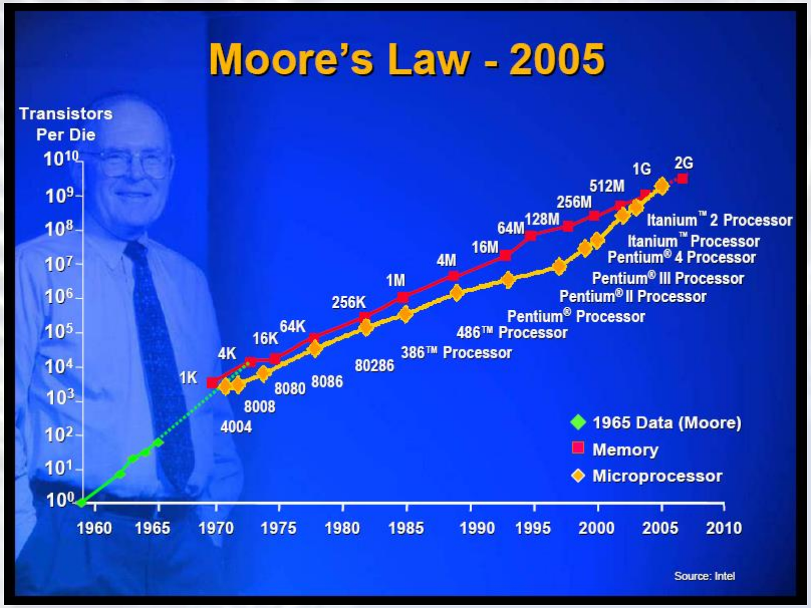
Digitális Rendszerek

- **Számítógépek**
  - Számítógép központok
  - Asztali számítógépek
  - Hordozható számítógépek
    - ~ Az adatfeldolgozó egység neve **CPU**

- **Beágyazott rendszerek**
  - Autó ECU
  - Kapu kódzár
  - Vérnyomásmérő
    - ~ Az adatfeldolgozó egység neve **mikrovezérlő**

Digitális Rendszerek

- **CPU↔ MIKROPROCESSZOR ↔ Mikrovezérlő**
  - Széles teljesítményskála, szinte folytonos átmenet
    - Méret, műveletvégzési képesség, magok száma
  - A technológiai háttér közös: Félvezető technológia
    - Óriási fejlődési ütem, Moore törvény, tranzisztorok száma
      - 1965: 2x/év , 1975 2x/2év
    - 2014-es rekord tr. szám adatok:
      - Intel Xeon IvyBridge 4,3mrd
      - IBM SyNAPTIC 5,4mrd
      - Xilinx UltraSCale 20mrd
- **Tervezhetetlen komplexitás!**

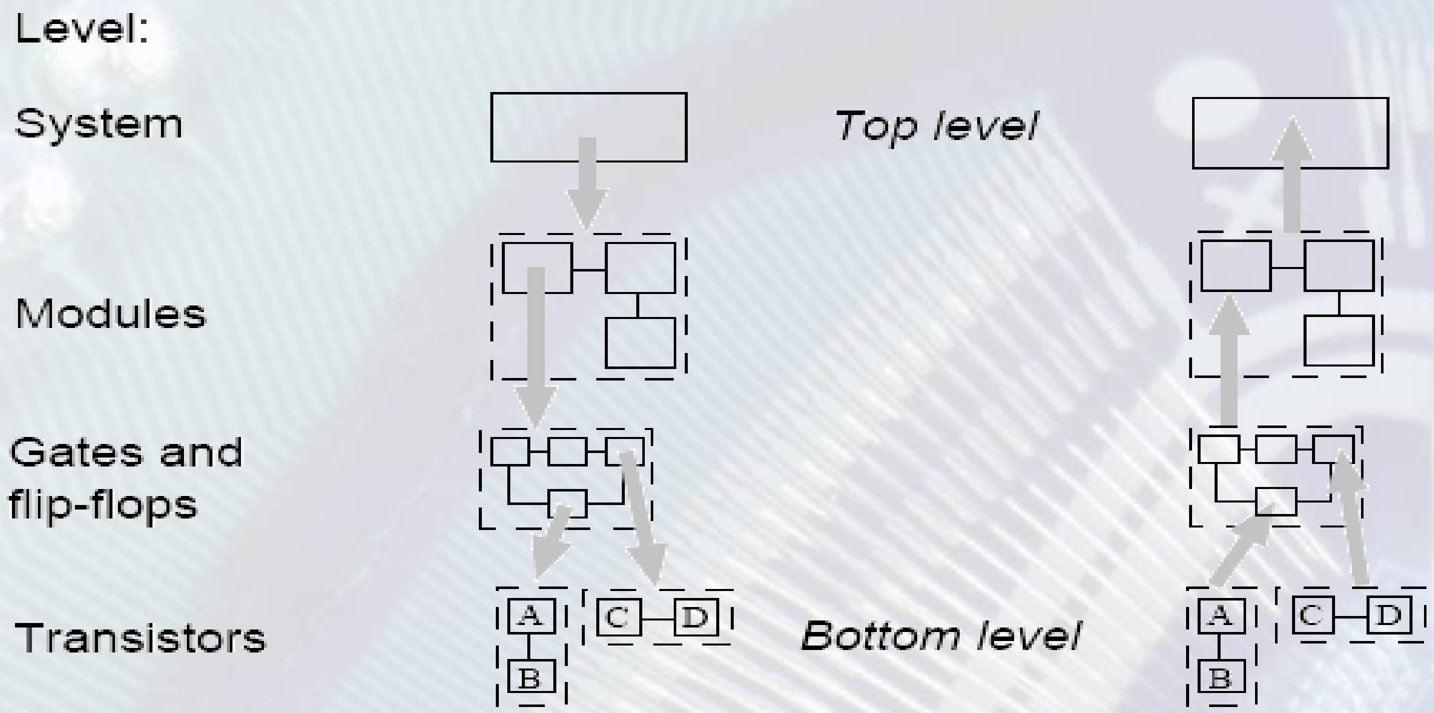


Digitális Rendszerek

- **Összetett rendszerek tervezése**
  - **Hierarchia**
    - Részekre osztás, majd újabb szintek bevezetés
  - **Modularitás**
    - Jól definiált funkciók és interfészek, építkezhetőség
  - **Egységesítés, szabványosítás**
    - Közös funkciók uniformizálása
    - Erőteljes újrahasznosítás
  - **A digitális technika tárgyban a tervezési feladatok során is ezeket az elveket fogjuk felhasználni, alkalmazni**

Hierarchikus tervezési módszerek

- **Felülről lefelé (top-down)**
- **Alulról felfelé (bottom-up)**



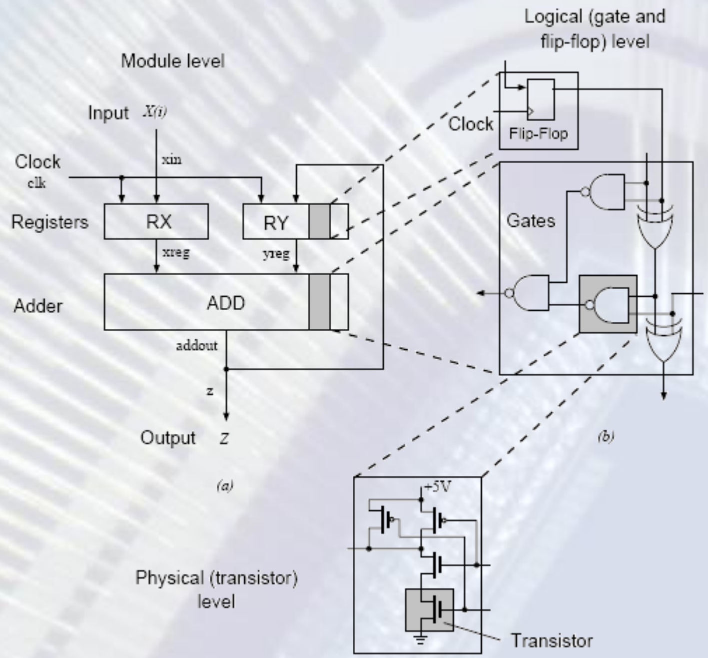
– Léteznek a kívánt típusú komponensek?

– Megfelel a rendszer a specifikációnak?

Tervezési szintek

- **A hierarchikus tervezési szintek szétválaszthatók**
  - A fizikai szinttel (félvezető tranzisztor) mi már nem foglalkozunk
- **Gyakran egyedi tervezési megközelítést igényelnek**

$$Z = \sum_{i=0}^{N-1} X(i)$$



Specifikáció finomítás

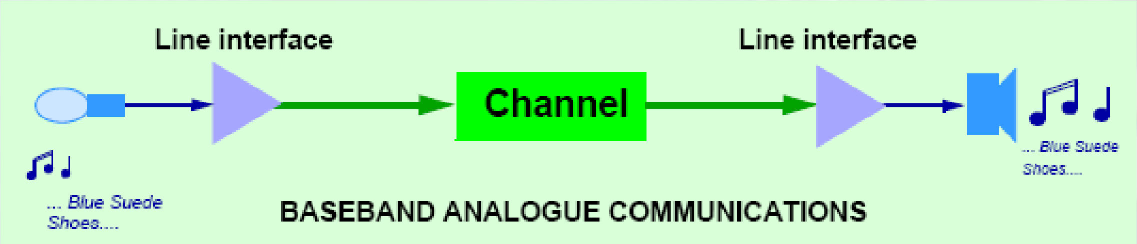
- **A tervezés, megvalósítás egy iteratív folyamat**

Specifikáció finomítása

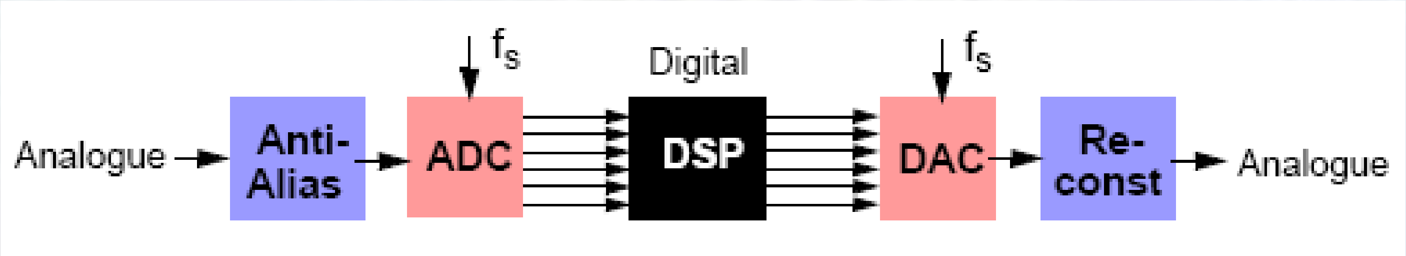
- **Felhasználói specifikáció**
  - Általában szöveges formában
  - Jellemzően nem műszaki paraméterek
- **Előzetes rendszerterv**
  - Követelmények lefordítása
  - Főbb paraméterek meghatározása
- **Funkcionális rendszerterv**
  - Globális döntések a megvalósításról
  - Modul funkciók specifikálása
- **Logikai/digitális tervezés, ellenőrzés**

Digitális technika

- **Beágyazott rendszerek**
  - A környezetből analóg és digitális jelek
  - Hagyományos feldolgozás analóg elemekkel

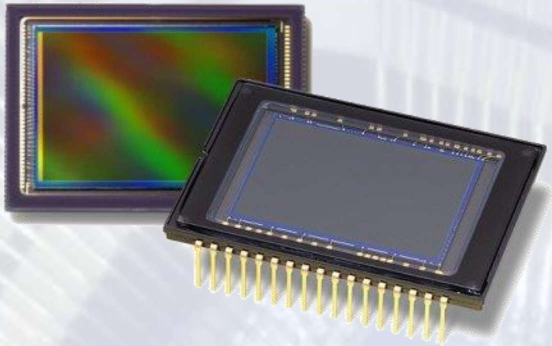


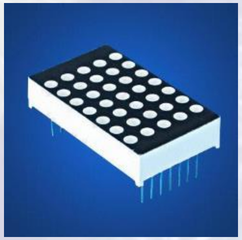
  - Korszerű feldolgozás digitális módon



Digitális technika

- **Analóg jelek feldolgozása konverzió után**
- **Közvetlen digitális jelek**
  - Nyomógomb
  - Billentyűzetek
    - Kódolás
    - Leolvasás
  - Képérzékelők
  - Léptető motor
  - Kijelzők





Digitális technika

- **Adatábrázolás**
  - Numerikus értékek
    - Külső jelek A/D konverzió után Temp = 26,5 °C
    - Belső adatok reprezentációja π = 3,1415
    - Memória cím 0x8000_FA14
  - Egyéb jelek, kódok
    - ON-OFF, egyéb diszkrét állapotok
    - Karakterek, kódtáblák
    - Speciális kódok (pozíció, tömörített, stb.)

Digitális technika

- **Számábrázolási módszerek**
  - Pozicionális számábrázolás, tetszőleges számrendszerben

$$\mathbf{D} = \sum_{i=0}^{n-1} d_i * r^i$$

  - ahol **r** a számrendszer alapja (radix),
  - $\mathbf{d_i}$ a számrendszer egy számjegye (digit)
  - Akár tekinthetjük egy polinomnak is

$$\mathbf{D=d_{n-1}*r^{n-1}+d_{n-2}*r^{n-2}+.....+d_2*r^2+d_1*r^1+d_0*r^0}$$

  - Például ismerjük az r = 10-es számrendszert
  - Ebben a decimális digitek ismert szimbólumai: 0, 1, 2 ,3 ,4 ,5 ,6, 7, 8, 9, (0…r-1)

Digitális technika

- **Számábrázolási módszerek**
- **Példa:**
  - A 2014_{10} jelentése értelemszerűen:
  - $2014_{10} = 2*10^3 + 0*10^2 + 1*10^1 + 4*10^0 =$
    $= 2000 + 0 + 10 + 4 = 2014_{10}$
  - Ugyanez 8-as számrendszerben is egy érvényes szám, de más numerikus értéket jelent (kb. a fele)
  - $2014_8 = 2*8^3 + 0*8^2 + 1*8^1 + 4*8^0 =$
    $= 2*512 + 0 + 1*8 + 4*1 = 1036_8$

Digitális technika

- **Digitális technikában fontos számrendszerek**
- **Dekadikus r = 10**
  - di = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9,
- **Bináris r = 2**
  - di = 0, 1, (a nevük bit, **bi**nary digi**t == bit)**
- **Oktális r = 8**
  - di = 0, 1, 2, 3, 4, 5, 6, 7,
- **Hexadecimális r = 16**
  - di = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F
- A számjegyek fenti szimbólumait a gépek bináris bitsorozatokkal reprezentálják

Digitális technika

- **Számjegyek bitkódjai → természtes kódkép**

  $D = \sum_{i=0}^{n-1} d_i * ri$ alapján

Érték	BIN	OKT	DEC	HEX	HEXDIG
0	0	000	0000	0000	0
1	1	001	0001	0001	1
2		010	0010	0010	2
3		011	0011	0011	3
4		100	0100	0100	4
5		101	0101	0101	5
6		110	0110	0110	6
7		111	0111	0111	7
8			1000	1000	8
9			1001	1001	9
10				1010	A, a
11				1011	B, b
12				1100	C, c
13				1101	D, d
14				1110	E, e
15				1111	F, f

- **$X_8 = b_2*2^2 + b_1*2^1 + b_0*2^0$**
- **$X_{16} = b_3*2^3 + b_2*2^2 + b_1*2^1 + b_0*2^0$**
- **$X_{10} = b_3*2^3 + b_2*2^2 + b_1*2^1 + b_0*2^0$**
  - X_{16}, X_{10} felírása formailag azonos, értelmezési tartományuk eltérő

Digitális technika

- **Konverzió számrendszerek között**
- **Bináris ↔ Hexadecimális egyszerű**
  - $16 = 2^4$, 1 hexadecimális digit 4 bináris digit (bit)
  - $2014_{16} = 10000000010100_2$, csoportosítás jobbról kezdve, és bal oldalon 4 bitre kiegészítve

2				0				1				4			
0	0	1	0	0	0	0	0	0	0	0	1	0	1	0	0

  - Szokásos írásmód $2014_{16} = 0010\_0000\_0001\_0100_2$
- **Bináris ↔ Oktális hasonlóan, 3 bites csoportokkal**
  - $8 = 2^3$, 1 oktális digit 3 bináris digit (bit)
  - $2014_8 = 010\_000\_001\_100_2$

Digitális technika

- **A Decimális → Bináris konverzió bonyolultabb, valódi számítási algoritmust kíván**
  - Egészosztás 2-vel, a maradék az új bit, a legkisebb helyiértéktől kezdve, amíg 0 a hányados
  - Példa decimális jelöléssel
  - Eredmény: (visszafelé kiolvasva) $2014_{10} = 11111011110_2$

Osztandó	Osztó	Hányados	Maradék
2014	:2	1007	0
1007	:2	503	1
503	:2	251	1
251	:2	125	1
125	:2	62	1
62	:2	31	0
31	:2	15	1
15	:2	7	1
7	:2	3	1
3	:2	1	1
1	:2	0	1

Digitális technika

- **Decimális → Bináris konverzió, másik algoritmus**

$$2^{N+1} >= \text{Decimális szám} > 2^N$$

- Ha igen, akkor a bináris alakban $d_N = 1$ és kivonás után újabb feltétel-vizsgálat következik a következő (kisebb) hatvánnyal

Dec. Szám	2^N	Szám> 2^N ?	Különbség	Bin. Digit
2014	2048	nem	2014	0
2014	1024	igen	990	1
990	512	igen	478	1
478	256	igen	222	1
222	128	igen	94	1
94	64	igen	30	1
30	32	nem	30	0
30	16	igen	14	1
14	8	igen	6	1
6	4	igen	2	1
2	2	igen	0	1
0	1	nem	0	0

Digitális technika

- **A Bináris → Decimális konverzió fontosabb**
  - Előző algoritmus inverze: Táblázat alapján, minden aktív d_i bináris digit numerikus értékét összegezzük

- $\mathbf{11111011110_2 = 2014_{10}}$, mert

  $\mathbf{= 1024 + 512 + 256 + 128 + 64 + 16 + 8 + 4 + 2 = 2014}$

Digitális technika

- **Bináris → Decimális konverzió, másik algoritmus**
  - Az osztó/hányados algoritmus inverze: Legnagyobb helyiértékű bittől kezdve duplázás és következő bit hozzáadása lépésről-lépésre
  - Alapja a számpolinom felírása Horner formulával:

$$\mathbf{D} = \sum_{i=0}^{n-1} b_i * 2^i$$

$$= b_{n-1}*2^{n-1}+b_{n-2}*2^{n-2}+\ldots..+b_2*2^2+b_1*2^1+b_0*2^0$$

$$= ((((((b_{n-1}*2)+b_{n-2})*2+\ldots..+b_2*2+b_1)*2+b_0$$

– **Példa: $2014_{10} = 11111011110_2 =$**

$$= ((((((((((1*2+1)*2+1)*2+1)*2+1)*2+0)*2+1)*2+1)*2+1)*2+1)*2+0)$$

- **Számrendszerek és konverziók összefoglalása**
  - Fontos számrendszerek: bináris, hexadecimális és decimális
  - A bináris az elsődleges, minden új ismeretünk majd erre épül, azonban nagyobb értéktartománynál mérete kezelhetetlen, áttekinthetetlen
  - A hexadecimális formátum ennek egy tömörített formája, nincs szükség algoritmikus konverzióra, a többjegyes hexa számokat számjegyenként bináris sorozattá alakítva közvetlenül a teljes bináris formát kapjuk. Az {A,B,C,D,E,F} szimbólumokat használjuk a {10,11,12,13,14,15} számértékek jelölésére
  - A többjegyes decimális számok kezelése bonyolult. Mindkét irányban (DEC→BIN, BIN→DEC) algoritmikus megoldások szükségesek, amelyek lépésenkénti végrehajtása adja meg a konverzió eredményét.

Digitális technika

- **Néhány fontosabb bináris érték, fejben számoláshoz**

2^7	2^8	2^{10}	2^{16}	2^{20}	2^{30}
128	256	1024	65536	1048576	1073741824
~száz		~ezer		~millió	~milliárd

- **Apró kellemetlenség, 1000 ≠ 1024**
- A korábban elterjedt k, M, G, T jelölések nem teljesen precízek

SI (decimális)				IEC (bináris)			
jel	név	érték		jel	név	érték	
k	kilo	10^3	1000^1	Ki	kibi	2^{10}	1024^1
M	mega	10^6	1000^2	Mi	mebi	2^{20}	1024^2
G	giga	10^9	1000^3	Gi	gibi	2^{30}	1024^3
T	tera	10^{12}	1000^4	Ti	tebi	2^{40}	1024^4

- Az új szabványos jelölés lassan terjed, mi is nehezen tanuljuk...

Bináris számábrázolás tulajdonságai

- **Eddig pozitív egészek**
  - N bit, 0-tól 2^{N-1} értéktartomány
  - Pozíció függő súlytényező: helyiérték
- **Műveletek:**
  - Összeadás szabályai (általában 2 operandus között): $0 + 0 = 0$, $1 + 0 = 1$, $0 + 1 = 1$, $1 + 1 = 10$, ahol az 1 az átvitel a következő, eggyel magasabb helyiértékre
  - Példa $6 + 3 = 9$, 4 biten
  - Átvitel a 2. pozíción
  - Eredmény esetleg 4 + 1 jegy, pl. $9 + 8 = 17$

		1		
	0	1	1	0
+	0	0	1	1
	1	0	0	1

Bináris számábrázolás tulajdonságai

- **Szorzás**
  - Bináris szorzás szabályai:
    0 * 0 = 0, 1 * 0 = 0, 0 * 1 = 0, 1 * 1 = 1
  - Nincs átvitel, de vannak részszorzatok és részszorzat összegek (több bemenetű összeadás?)
  - Példák:
  - 6*3 = 18 14*11 = 154

	0	1	1	0	*	0	0	1	1
	0	0	0	0					
		0	0	0	0				
			0	1	1	0			
				0	1	1	0		
	0	0	1	0	0	1	0		

				1	1	1	0	*	1	0	1	1
				1	1	1	0					
			1	1	1	0						
		0	0	0	0							
	1	1	1	0								
1	0	0	1	1	0	1	0					

  - Az eredmény alapvetően 2N bites (4+4 = 8)

Bináris számábrázolás tulajdonságai

- **Osztás**
  - Bináris egész osztás szabályai:
    1 / 1 = 1, 0 / 1 = 0, az osztás 0-val nem értelmezett
  - Pozitív számokra jól értelmezhető
  - Példák:
  - 11 : 3 = 3, maradék 2

1	0	1	1	:	1	1	=	0	1	1
1	0									
1	0	1								
0	1	0	1							
	0	1	0							

Előjeles számábrázolás

- **Eddig: Összeadás, szorzás, maradékos osztás → Egyik sem vezet ki a pozitív számok halmazából, bár a számtartományt esetleg növelni kell!**
- **Kivonás? Negatív hozzáadása? Mi a negatív?**
- **Előjeles számok:**
  - Normál jelölésben – (elő)jel
  - De itt csak „0” és „1” van, nincs több szimbólum
  - Más szabály kell (az előjel is egy új bit):
    - Előjel + érték
    - Eltolt (offset) bináris
    - Egyes komplemens
    - **Kettes komplemens – Csak ezzel foglaljozunk**

Előjeles számábrázolás

- **Komplemens kódok: A kettes komplemens fontos!!**
- **Egyes komplemens (1’s C):**
  - Képzési szabálya: Negatív értékhez minden bináris számjegyet invertálunk ($0 \rightarrow 1, 1 \rightarrow 0$)
- **Kettes komplemens (2’s C):**
  - Képzési szabálya:
    Negatív értéknél minden bitet invertálunk és az így kapott számhoz hozzáadunk 1-et
  - Más módszer: A szám értékét 2^N-ből binárisan kivonva megkapjuk a negatívjának 2’s C kódját.
    Pl. 4 bitre − 5 képzése: 16−5 → 10000 – 0101= 1011,
    mert igaz, hogy 0101 + 1011 = 10000, ami viszont 4 biten 0.

Bináris	2's C
0111	+7
0110	+6
0101	+5
0100	+4
0011	+3
0010	+2
0001	+1
0000	0
1111	-1
1110	-2
1101	-3
1100	-4
1011	-5
1010	-6
1001	-7
1000	-8

Digitális technika

- **Kettes komplemens számábrázolás**
  - A pozícionális számábrázolás definíciója alapján

$$D = -b_{n-1} * 2^{n-1} + \sum_{i=0}^{n-2} b_i * 2^i$$

  - b_{n-1} a legnagyobb helyiértékű bit (MSb), b_i pedig a többi bit. Az MSb negatív értékű, ha nem nulla
  - A 2'sC előjeles számokkal végzett műveletvégzési szabályok megegyeznek a normál pozitív számokra vonatkozókkal
  - Egyetlen 0 kód, önmaga 2's C komplemens kódja
  - Könnyű aritmetikai tesztek (=, ≠, >, <, ≤, ≥)

Előjeles számábrázolás

- **Kettes komplemens (2’s C) méretkonverzió**
  - Előjel kiterjesztés: Számjegyek számának növelése
  - Pozitív számokra egyértelmű, bal oldalon kiegészítés 0-kal, a szám értéke természetesen nem változik
  - A **+5** érték 4 biten 0101 és 12 biten 0000_0000_0101
  - A –5 érték 4 biten 1011 és 12 biten 1111_1111_1011
  - Mert a 2’s C szabályai szerint bitjeit invertálva + 1
    - 0000_0000_0100 + 1 = 0000_0000_0101
  - Általánosan, ha kevesebb bitről előjel kiterjesztéssel méretet növelünk több bitre, az **érték** nem változik
    - Jelentősége: pl. konverzió különböző méretű adatformátumok között (8 bites bájt → 32 bites szó)

Valós számok

- **Pozicionális számrendszer, negatív kitevők,**
  - r^{-1}, r^{-2}, ….. r^{-n}, tört helyiértékek, r^0 –tól jobbra
- **Bináris számrendszer valós számokra**

2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}
16	8	4	2	1	0,5	0,25	0,125

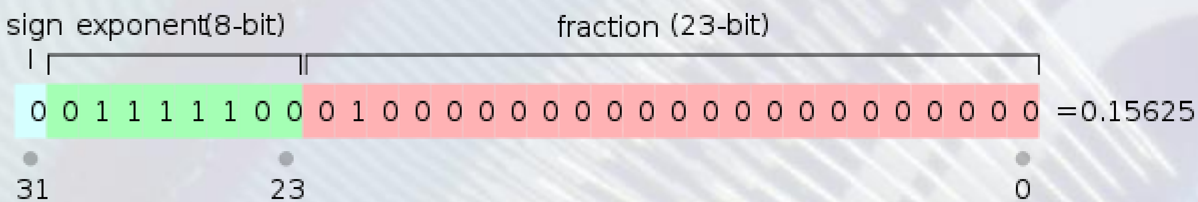
  - Implicit „kettedes” pont a ↑ megfelelő helyen
- **Tehát ebben a számformátumban pl. előjelesen a**
  - 00110101 = 6,625 illetve az 11111111 = -0,125
- **Tetszőleges pontosság, bitszám növelésével**
- **Probléma: $0{,}1_{10} = 0{,}00011001100110011 00\ldots_2$**

Lebegőpontos számformátum

- A számok normál alakját modellezi

$$\mathbf{D = (-1)^{e} * m * r^{k}}$$

- ahol **e** az előjel, **r** a radix (2 vagy 10), **m** a mantissza, **k** a kitevő. A szabvány több méretet definiál (32/64/128 bit).
- Pl. 32 biten: e=1 bit, m=24 (23+1) bit, k=8 bit



- Értéktartománya széles: 32 biten maximum $\pm 3{,}4*10^{38}$
- Tartalmazza a 0-t, és a legkisebb értékei $\pm 1{,}4*10^{-45}$
- Egyenletes relatív pontosság, a mantissza pontossága, 2^{-23}

Decimális számábrázolás

- **Digitális hardver → bináris számábrázolás**
- **„Könnyű” a műveletvégzők tervezése**
  - ADD, SUB, MULT, DIV, SQRT
- **Azonban szükség lehet a decimális értékre vagy akár decimális aritmetikára**
  - Pl. numerikus kijelzés esetén
- **Két megoldás lehetséges, feladattól függ a választás**
  - Decimális adatok tárolása (nem hatékony), decimális műveletvégzés (bonyolultabb), közvetlen eredmény
  - Bináris adatok tárolása (hatékony), bináris műveletek (egyszerűbb), kijelzés előtt BIN2BCD konverzió

Decimális számábrázolás

- **Decimális számjegyek kódolása, ábrázolása**
- **Csak az BCD kód lényeges, a 8421 bites súlyozással**
- A 2421 és EXC3 kódokra igaz, hogy 9-es komplemensük negáltjai pl. 2 és 7
- Az 5-ből-2 kód képes bithibák jelzésére (ahol nem 2 bit aktív, az hibás BCD számjegy)

Érték	BCD	2421	EXC3	5-ből2	Gray
0	0000	0000	0011	00011	0110
1	0001	0001	0100	11000	0010
2	0010	0010	0101	10100	0011
3	0011	0011	0110	01100	0001
4	0100	0100	0111	10010	0000
5	0101	1011	1000	01010	1000
6	0110	1100	1001	00110	1001
7	0111	1101	1010	10001	1011
8	1000	1110	1011	01001	1010
9	1010	1111	1100	00101	1110

- A Gray kód speciális un. pozíciókód, minden egymást követő kódszava között 1 bit változik, még a 9-0 átmenetnél is. Ez kedvező lehet bizonyos esetekben.

Kódolási technikák

- **Láttuk az eddigiekben (pl. a decimális számjegyek kódolásánál), hogy különféle lehetőségek vannak**
  - A numerikus értékeknél fixpontos, lebegőpontos
  - A decimális számjegyeknél BCD-EXC3-GRAY
  - Nemcsak számokkal dolgozunk: Szöveg, hang, kép, mérési adatok
  - Tetszőleges egyedi események, állapotok
- **A továbbiakban megvizsgáljuk a kódolási technikák néhány egyszerűbb területét**
- **Feladat: Adott célra legkedvezőbb kódolás elérése**

Kódolási technikák

- **A kódolási ABC 2 elemű {0,1}**
- **A legegyszerűbb esetekben**
  - k bittel 2^k kódszó képezhető, ill.
  - N darab kódszót minimum $n \geq \lceil \log_2 N \rceil$ bittel tudunk képezni
- **A kódkészlet osztályozása**
  - Fix vagy változó hosszúságú
  - Numerikus, alfanumerikus, grafikus
  - Pozíció kód vagy szomszédos kódolású
  - Redundáns biteket tartalmazó hibajelző és/vagy javító

Kódolási technikák

- **Fix hosszúságú kódok**
  - Minimális bitszám igény, min. n≥ ⌈$\log_2$ N⌉
    - Bináris, vagy bármely, tetszőleges sorrendű (pl. Gray)
  - Nem minimális bitszám mellett
    - k-az-n-ből, pl. 1-az-N-ből, 2-az-5-ből (láttuk)
    - Könnyen kezelhető, értelmezhető, dekódolható, digitális hardverrel generálható

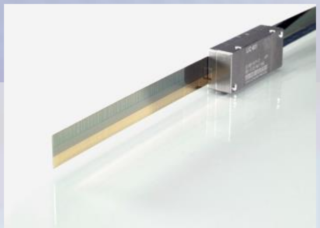
1-a-6-ból
100000
010000
001000
000100
000010
000001

- Eredeti ASCII karaktertáblázat 7 bites, 128 kódszó

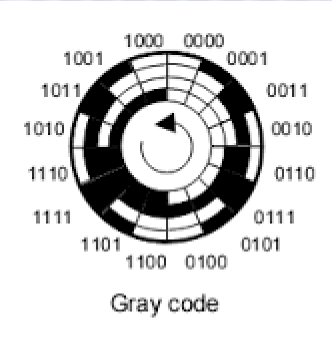
ASCII Code Chart

	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

Kódolási technikák



- **Pozíció kódok**
  - A lineáris ill. forgó abszolút pozíció jeladóknál a kód megbízható adatátvitelt ad, a **szomszédos** kódszavak között mindig csak1 bit változás (forgóadónál a végértéken is)



- **Gray, reflektíve kód (generálása könnyű)**
  - n bitből $N=2^n$ méretű kódszókészlet generálható
  - Lehet kevesebb, de páros kódszó számot is használni

0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0
0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0
0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0

  - Ha már megismertük a XOR logikai függvényeket, látni fogjuk, hogy generálása viszonylag könnyű

Első előadás vége