



BUDAPESTI MŰSZAKI ÉS GAZDASÁGTUDOMÁNYI EGYETEM
VILLAMOSMÉRNÖKI ÉS INFORMATIKAI KAR
MÉRÉSTECHNIKA ÉS INFORMÁCIÓS RENDSZEREK TANSZÉK

Digitális technika

VIMIAA02

1. EA

Fehér Béla
BME MIT

Vetítés előadások

- Ebben az évben az előadások további egy (IB027) teremben csak kivetítőről követhető, a kapcsolat így nem elég közvetlen
- Ennek javítására beiktatunk visszajelzési lehetőséget
- Bármelyik teremből kérdések, megjegyzések emailben küldhetők a benes@mit.bme.hu címre
- Résztémák végén a kérdéseket megnyitjuk és próbálunk válaszolni

Digitális Rendszerek

- Számítógépek

- Számítógép központok
- Asztali számítógépek
- Hordozható számítógépek

- ~ Az adatfeldolgozó egység neve **CPU**

- Beágyazott rendszerek

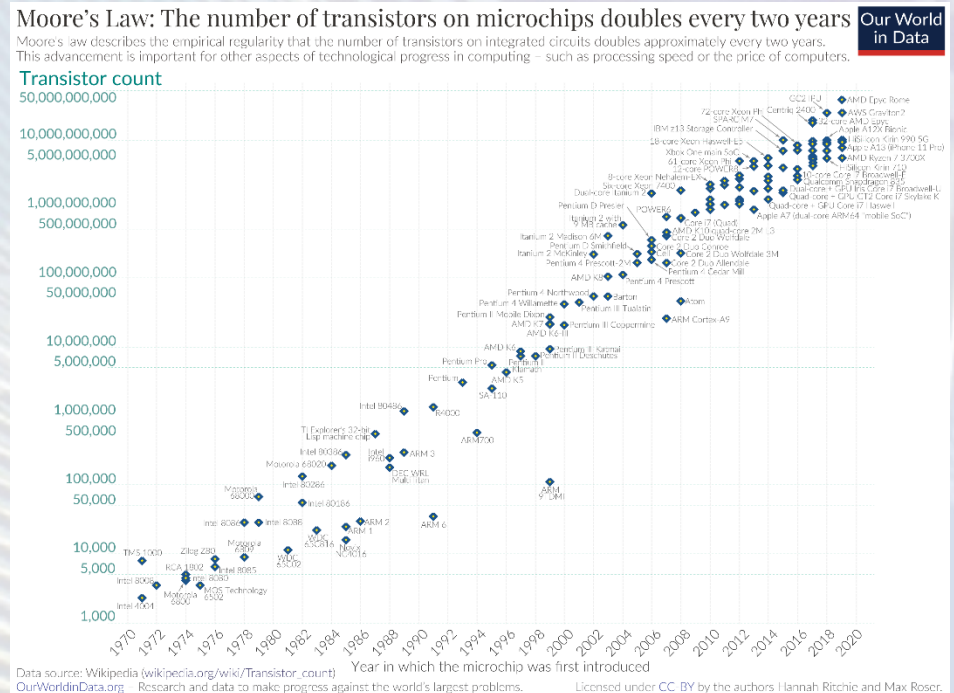
- Autó ECU
- Kapu kódzár
- Vérnyomásmérő

- ~ 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: évente 2x
 - 1975: 2 évente 2x
 - Órajelsebesség
 - Energiafogyasztás

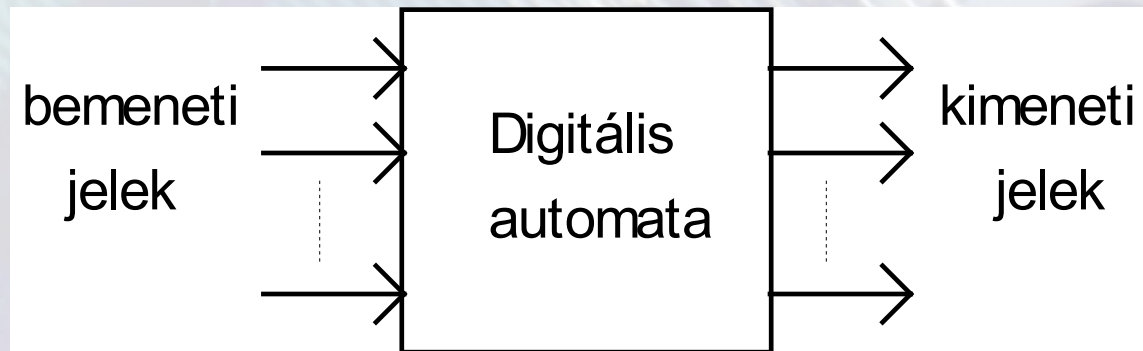


Digitális Rendszerek

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

Digitális technika

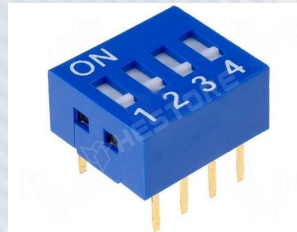
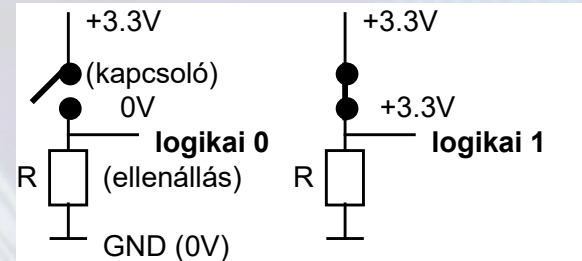
- **A digitális technika tárgyban** olyan elektronikus információ feldolgozó gépek tervezésének, belső felépítésének, megvalósításának, működésének alapjaival foglalkozunk, amelyek be és kimenetein az információ 2 értékű logikai jelek formájában jelenik meg (0, 1).



Digitális technika

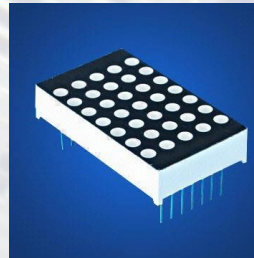
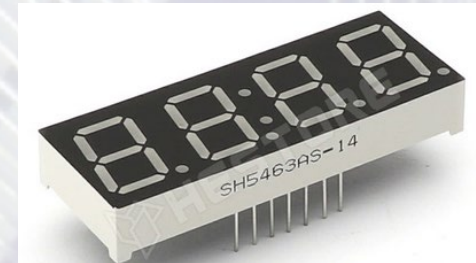
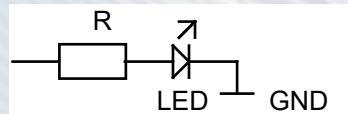
- **Közvetlen digitális bemeneti jelek**

- Nyomógomb
- Kapcsolók stb.



- **Közvetlen digitális kimeneti jelek**

- LED-ek,
- 7 szegmenses LED kijelzők
- LED mátrix kijelzők stb.



Digitális technika

- **Adatábrázolás kódokkal**
 - Aritmetikai kódok (számok)
 - Ha a kódolt információnak nagysága van. Pl. Fizikai mennyiség (feszültség, hőmérséklet...) kódolása.
 - Ilyenkor összehasonlíthatók és aritmetikai műveletek végezhetők közöttük.
 - Analóg jelből analóg/digitális konverzióval állítható elő
 - Egyéb jelek, kódok
 - Halmaz elemeinek megkülönböztetése. Pl. alma:00, körte:01, sárgabarack:10, őszibarack:11
 - Közlekedési lámpa állapotai (PSZ: 100,110,001,010)
 - Karakterek kódolása (Pl. ASCII kód)
 - Speciális kódok: Pl. pozíció kód, hiba javító kódok, változó hosszúságú kódok (tömörítés) stb.

Digitális technika

- Számábrázolási módszerek

- Pozícionális számábrázolás, N helyiértéken, tetszőleges számrendszerben

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

- ahol r a számrendszer alapja (radix)
- d_i a számrendszer egy számjegye (digit)
- Akár tekinthetjük egy polinomnak is, r hatványaival

$$D = d_{N-1} * r^{N-1} + d_{N-2} * r^{N-2} + \dots + d_2 * r^2 + d_1 * r^1 + d_0 * r^0$$

- 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 a számjegy sorozat a 8-as számrendszerben is egy érvényes szám, de más számértéket jelent (kb. a fele)

- $2014_8 = 2*8^3 + 0*8^2 + 1*8^1 + 4*8^0 =$
 $= 2*512 + 0 + 1*8 + 4*1 = 1036_{10}$

Digitális technika

- **Digitális technikában fontos számrendszerek**
- **Tízés/Decimális/Dekadikus** $r = 10$
 - $d_i = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9,$
- **Kettes/Bináris** $r = 2$
 - $d_i = 0, 1,$ (a nevük bit, **binary digit == bit**)
- **Nyolcas/Oktális** $r = 8$
 - $d_i = 0, 1, 2, 3, 4, 5, 6, 7,$
- **Tizenhatos/Hexadecimális** $r = 16$
 - $d_i = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F$
 - $d_i = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, a, b, c, d, e, f$
 - A számjegyek fenti szimbólumait a gépek bináris bitsorozatokkal reprezentálják

Digitális technika

Konverzió számrendszerek között

- **A Bináris $\rightarrow$ Decimális konverzió**
 - Táblázat alapján, minden aktív d_i bináris digit numerikus értékét összegezzük
- $11111011110_2 = 2014_{10}$, mert
$$\begin{aligned} &= 1*2^{10} + 1*2^9 + 1*2^8 + 1*2^7 + 1*2^6 + 0*2^5 + 1*2^4 + 1*2^3 + 1*2^2 + 1*2^1 + 0*2^0, \\ &= 1024 + 512 + 256 + 128 + 64 + 0 + 16 + 8 + 4 + 2 + 0 \\ &= 2014 \end{aligned}$$

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:

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

$$= b_{N-1} * 2^{N-1} + b_{N-2} * 2^{N-2} + \dots + b_2 * 2^2 + b_1 * 2^1 + b_0 * 2^0$$

$$= ((((((b_{N-1} * 2) + b_{N-2}) * 2 + \dots + b_2) * 2 + b_1) * 2 + b_0$$

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

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

Digitális technika

- **Bináris $\leftrightarrow$ Hexadecimális, egyszerű csoportosítás**
 - $16 = 2^4$, 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

_0100₂

- **Bináris $\leftrightarrow$ 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 lesz a hányados
 - Példa decimális jelöléssel
 - Eredmény:
Visszafelé kiolvasva,
az első bit a legkisebb helyiértékű bit, (LSB Least Significant Bit)
 $2014_{10} = 11111011110_2$

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 $\rightarrow$ Bináris konverzió, másik algoritmus**

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

- Ha igen, akkor a bináris alakban $d_N = 1$ és a kivonás után újabb feltétel vizsgálat következik a következő (kisebb) hatvánnyal
- Az első bit a legnagyobb helyiértékű bit (MSB, Most Significant Bit)
 $2014_{10} = 11111011110_2$

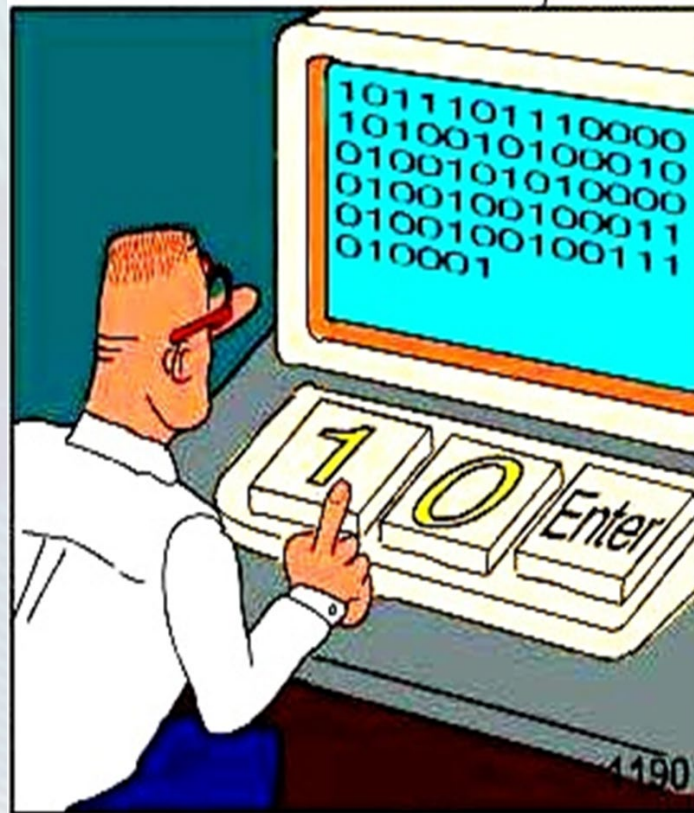
Dec. Szám	2^N	Szám $> 2^N$?	Különbség	Bin. Digit
2014	2048	nem	2014	0
2014	1024	igen	990	1
990	512	igen	478	1
478	256	igen	222	1
222	128	igen	94	1
94	64	igen	30	1
30	32	nem	30	0
30	16	igen	14	1
14	8	igen	6	1
6	4	igen	2	1
2	2	igen	0	1
0	1	nem	0	0

Digitális technika

- **Számrendszerek és konverziók összefoglalása**
 - Fontos számrendszerek: bináris, hexadecimális és decimális
 - A bináris az elsődleges, minden új ismeretünk majd erre épül, azonban nagyobb értéktartománynál mérete miatt kezelhetetlen, áttekinthetetlen, kényelmetlen
 - 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 bináris kezelése bonyolult. Mindkét irányban (DEC→BIN, BIN→DEC) algoritmikus megoldások szükségesek, amelyek speciális aritmetikai műveletek elvégzése után adják meg a konverzió eredményét.

Kérdések, észrevételek

- A kérdéseket a benes@mit.bme.hu címre várjuk



Digitális technika

- Néhány fontosabb bináris érték, fejben számoláshoz

2^7	2^8	2^{10}	2^{16}	2^{20}	2^{30}	2^{32}
128	256	1024	65536	1048576	1073741824	4294967296
~száz		~ezer		~millió	~milliárd	

- Apró kellemetlenség, $1000 \neq 1024$ ($10^3 \neq 2^{10}$)
- A korábban elterjedt k, M, G, T nagyságrendi 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, de egy informatikusnak illik tudni róla

Bináris számábrázolás tulajdonságai

- **Eddig pozitív egészek**
 - N bit, 0-tól 2^N-1 terjedő pozitív egész értéktartomány
 - Pozíció függő súlytényező: helyiérték (1,2,4,8...)
- **Aritmetikai műveletek:**
 - **Bináris összeadás szabályai** (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. és a 3. pozíción
 - Eredmény esetleg 4 + 1 jegy, pl. $9 + 8 = 17$

		1	1				
		0	1	1	0		6
+		0	0	1	1	+	3
		1	0	0	1		9

Bináris számábrázolás tulajdonságai

- **Bináris szorzás szabályai**

- Egy bites operandusokra:

$$0 * 0 = 0, 1 * 0 = 0, 0 * 1 = 0, 1 * 1 = 1$$

- Bit szorzásnál nem keletkezik átvitel, de lesznek részszorzatok, amiket páronként összegezni kell (vagy esetleg több bemenetű összeadással?)

- Példák: $6 * 3 = 18$

	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		

$$14 * 11 = 154$$

				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 ($4+4 = 8$)

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 van – (elő)jel, egyedi szimbólum
 - De itt csak „0” és „1” van, nincs több szimbólum
 - Más szabály kell (az előjel is egy új bit):
 - Előjel + abs. érték (pl. lebegőpontos formátum mantissza)
 - Eltolt (offset) bináris (pl. lebegőpontos form. exponens)
 - Egyes komplement
 - Kettes komplement – Csak ezzel foglalkozunk

Előjeles számábrázolás

- **Komplement kódok: A kettes komplement fontos!!**
- Egyes komplement (1's C):
 - Képzési szabálya (egy szám -1 -szerese): Minden bináris számjegyet invertálunk ($0 \rightarrow 1, 1 \rightarrow 0$)
- **Kettes komplement (2's C):**
 - Képzési szabálya (egy szám -1 -szerese):
Minden bitet invertálunk és az így kapott számhoz hozzáadunk 1-et és csak az eredeti számú bitet őrizzük meg
 - Más módszer: A szám értékét 2^N -ből binárisan kivonva megkapjuk a negatívjának 2's C kódját.
Pl. 4 bitre -5 képzése: $2^4 - 5 = 16 - 5 \rightarrow 10000 - 0101 = 1011$,
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 komplement számábrázolás**

- A pozicionális számábrázolás definíciója alapján

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

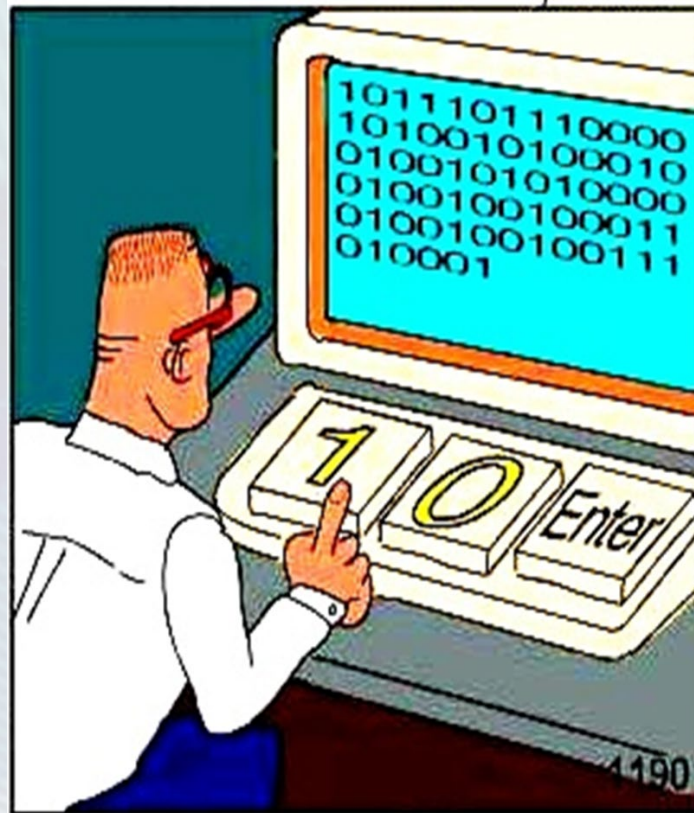
- b_{N-1} a legnagyobb helyiértékű bit (neve: MSB), b_i pedig a többi bit. Az MSB negatív értékű, ha nem nulla
- A 2's C előjeles számokkal az összeadás szabályai megegyeznek a normál pozitív számokra vonatkozókkal (az eredmény is 2's C)
- Egyetlen 0 kód, önmaga kettes komplement kódja
- Könnyű aritmetikai tesztek ($=$, $\neq$, $>$, $<$, $\leq$, $\geq$)

Előjeles számábrázolás

- **Kettes komplement (2's C) méretkonverzió**
 - **Előjel kiterjesztés:** Számjegyek számának növelése
 - Pozitív számokra egyértelmű, bal oldalon kiegészítés 0-kal, a numerikus érték 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 ennek bitjeit invertálva + 1, azaz $0000_0000_0100 + 1 = 0000_0000_0101$
 - 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)

Kérdések, észrevételek

- A kérdéseket a benes@mit.bme.hu címre várjuk



Valós számok

- Az eddigi pozícionális számrendszer, a törtrészre is kiegészíthető, csak negatív kitevőkkel
 - $r^{-1}, r^{-2}, \dots, r^{-n}$, tört helyiértékek, r^0 -tól jobbra ($1/2, 1/4, \dots$)
- Bináris előjeles (2-es komplement) számrendszer 8 biten valós számokra, pl. 3 db tört számjegy esetére:

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 (nem létező) „kettedes” pont a ↑ megfelelő helyen

- Tehát ebben a számformátumban pl. előjelesen a
 - $00110\ 101 = 6 + 0.5 + 0.125 = 6,625$
 - az $11111\ 111$ negatív, tehát 2-es kompl. Képzés kell az abszolút értékhez: $00000001 = 0.125$ tehát $11111111 = -0.125$
- Tetszőleges pontosság, bitszám növelésével elérhető, DE
 - Probléma: $0,1_{10} = 0,0001100110011001100\dots_2$

Fixpontos számábrázolás tulajdonságai

- A teljes értéktartományt (FSR, Full Scale Range) a legnagyobb helyiértékű bit (MSB) értéke határozza meg, a példában az előjeles számokra $\sim \pm 2^4 = \sim \pm 16$

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

- Két érték közötti min. eltérést (felbontás, pontosság) a legkisebb helyiértékű bit (LSB) értéke határozza meg, a példában $\sim \pm 2^{-3} = \sim \pm 0,125$
 - Nagy értéktartomány $\rightarrow$ sok egész bit
 - Nagy pontosság $\rightarrow$ sok törtrész bit
 - Rögzített bitszámnál kompromisszum kell
- Megoldás $\rightarrow$ Skálázó tényező alkalmazása

Lebegőpontos számformátum

- A számok normál alakját modellezi, a választott számrendszer szerinti skálázó tényező használatával

$$D = (-1)^e * m * r^k$$

- ahol **e** az előjel, **m** a mantissza, **r** a radix (2 vagy 10), **k** a kitevő. A szabvány több méretet definiál (32/64/128 bit).
- Pl. az IEEE754 szabvány szerint, 32 biten a formátum a következő: e=1 bit, m=24 (23+1) bit, k=8 bit, és az érték
$$(-1)^e * (1+m) * 2^{(k-127)}$$
- 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 (kijelzés) vagy akár decimális aritmetikára (pontosabb banki számítások)**
- **Két megoldás lehetséges, feladattól függ a választás**
 - Decimális adatok tárolása (nem hatékony), decimális műveletvégzés (bonyolultabb), de közvetlen eredmény
 - Bináris adatok tárolása (hatékony), bináris műveletek (egyszerűbb), **kijelzés előtt BIN → DEC konverzió**

Decimális számábrázolás

- Decimális számjegyek kódolása, ábrázolása
- A binárisan kódolt decimális (**BCD**, Binary Coded Decimal) kódban a bitek **számjegyenként 4 biten**, természetes 8-4-2-1 súlyozással szerepelnek
- Érvényes **NBCD** kódok értéktartmánya: **0000-1001** (Pl. 1010 nem érvényes kód)
- Pl. 1986 NBCD kódja:
0001100110000110

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

Kódolási technikák

- A numerikus értékeknél fixpontos (előjeles) egész, tört, lebegőpontos
- Nemcsak számokkal dolgozunk:
 - Szöveg, hang, kép, stb.
- A továbbiakban megvizsgáljuk a kódolási technikák néhány egyszerűbb területét
- Feladat: Adott célra legkedvezőbb kódolás elérése

Kódolási technikák

- **A bináris kódolási szimbólumkészlet 2 elemű $\{0,1\}$**
- **A legegyszerűbb esetekben**
 - k db bittel 2^k db kódszó képezhető, ill.
 - N darab kódszót minimum $n \geq \lceil \log_2 N \rceil$ bittel tudunk képezni (pl. 10 db kódszó minimum $\lceil \log_2 N \rceil = 3,33 \rightarrow 4$)
- **Kódkészlet osztályozása**
 - Fix vagy változó hosszúságú
 - Pozíció kód (szomszédos kódolású)
 - Hibajelző és/vagy hiba javító



Kódolási technikák

- **Fix hosszúságú kódok**

- Minimális bitszám igény, min. $n \geq \lceil \log_2 N \rceil$

- Bináris, vagy bármely, tetszőleges sorrendű

- Nem minimális bitszám mellett

- **k-az-n-ből** (k db 1-es az n db bitből),
pl. 1-a-6-ból

- Könnyen kezelhető, értelmezhető,
digitális hardverrel generálható, dekódolható

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

- Eredeti ASCII (American Standard Code for Information Interchange) karaktertáblázat 7 bites, 128 db kódszó, pl.

& = 010_0110

A = 100_0001

a = 110_0001

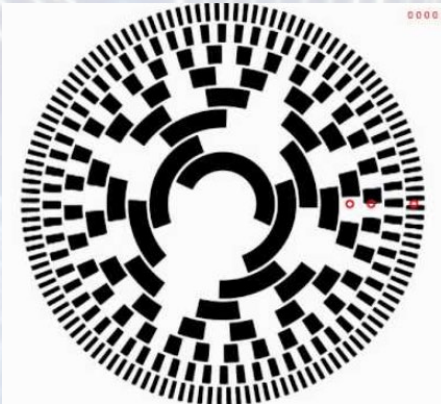
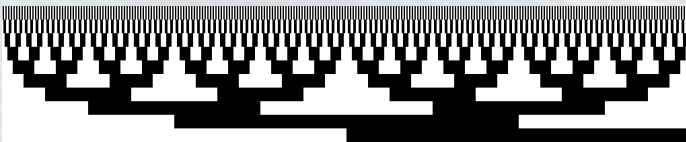
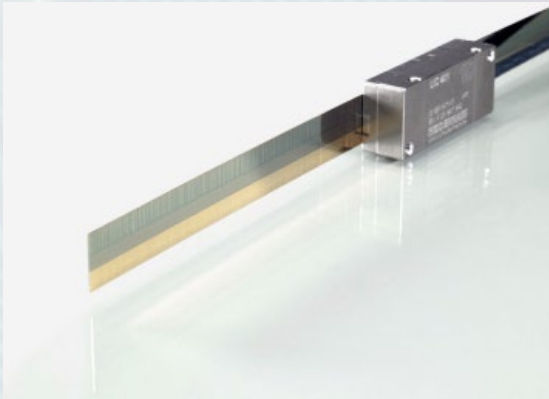
0 = 011_0000

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 csak 1 bit változás (forgóadónál a végértéken is)



Kódolási technikák

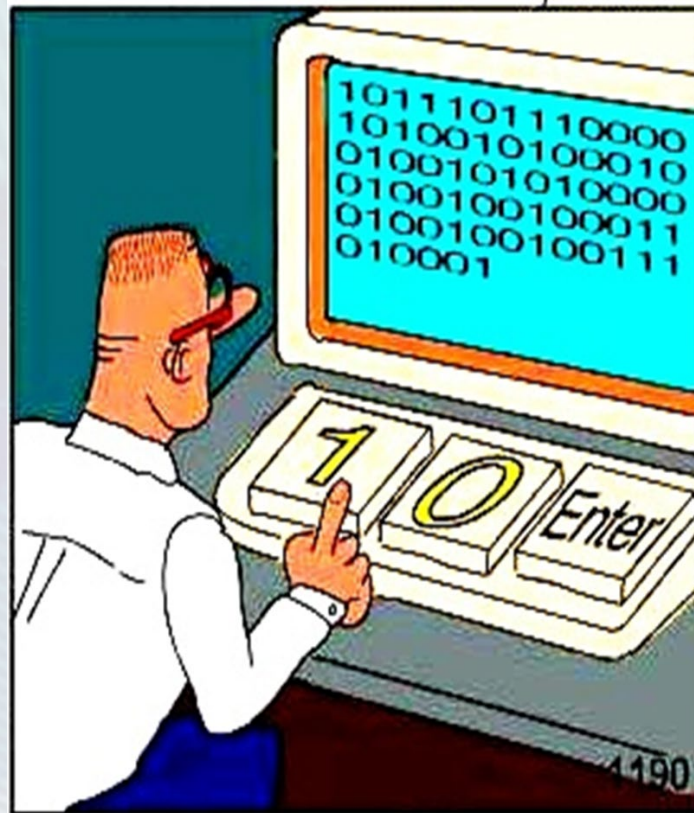
- Gray, tükrözött kód bináris kód
- n bitből $N=2^n$ méretű kódszókészlet generálható

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

- A bal alsó saroktól kezdve kiolvashatók a 2, 3, 4 bites kódtáblák (2 bit 4 db, 3 bit 8 db, 4 bit 16 db)
- Lehet kevesebb, de páros kódszó számot is használni, az aktuális tábla középszimmetrikus oszlopaival (pl. 10 db kódszó 4 biten, ha éppen erre lenne szükség)
- Ha már megismertük majd a XOR (Exclusive OR, kizáró VAGY) logikai függvényeket, látni fogjuk, hogy a Gray kód generálása viszonylag könnyű

Kérdések, észrevételek

- A kérdéseket a benes@mit.bme.hu címre várjuk



Verilog

Verilog 1-14

1. EA vége