

IT ALAPISMERETEK

Előadás vázlat

1

Tartalomjegyzék - Témakörök:

1. Számítástechnikai alapfogalmak
2. Számrendszerek: a bináris és a hexadecimális számrendszer, számok ábrázolása
3. Adatok ábrázolása a számítógépben
4. Logikai műveletek, logikai operátorok, igazságtáblák, De Morgan azonosságok

A hardver:

5. A számítógép felépítése, hardver blokk-vázlat, kapcsolódások

2

6. A processzor
7. A gépi kódú utasítások
8. A megszakítások, a megszakítási rendszer
9. A memória
10. Háttértárak
11. Egyéb perifériák
12. A szoftver

3

1. Számítástechnikai alapfogalmak

1.1 A számítógép feladata:

adatok önálló, emberi beavatkozás nélküli feldolgozása.

a feldolgozást irányító, a gép számára végrehajtható formában adott utasítások sorozatát, nevezzük programnak, amit képes egyértelműen végrehajtani

számítógépes program: valamilyen számítógép-rendszernek egységként átadható utasításhalmaz, mely irányítja annak működését.

4

Egy feladat számítógépes megoldása folyamán három terület együttműködését tételezzük fel

Hardver: A számítógépek fizikai, kézzel megfogható elemeit (az elektromos, elektronikai, elektromechanikai részeket) jelenti, amelyeket a feldolgozás folyamán használunk.

Szoftver: A számítógép hardverét működtető, a feladatok megoldására alkalmassá tevő, nem megfogható rész, azaz a hardveren futtatható programok és adatok ill. ezek dokumentációit, általában a gép használatához szükséges szellemi terméket jelenti.

5

Orgver: A hardver és a szoftver összehangolása a felhasználói igényeknek megfelelően, valamint magába foglalja a felhasználó képzettségét ismereteit is.

1.2. A számítástechnika története

1.2.1. Mechanikus eszközök

Ókorban: ABAKUSZ ez a golyós számológép az első számolóeszköz. Ennek megfelelője napjainkban a szoroban, vagy a golyós számoló tábla.

1623 Wilhelm Schickard az első mechanikus, fogaskerekekkel működő összegző gépe.

6

1625 Pascal mechanikus működésű számológépe. Megvalósítja az automatikus átvitelképzést az egyes helyiértékek között a fogaskerekes összeadó gépén.

1671 Leibniz a négy alpműveletes gép elkészítése. Mechanikus összeadó és szorzó gépet készít ahol a szorzás műveletét ismételt összeadással valósítja meg.

1679 Leibniz javasolja elsőnek a kettes számrendszert a mechanikus összeadó és szorzó gépeken.

1786 Muller bevezette a regiszterek közötti műveletvégzést.

7

1833 Babbage megkonstruálja az un. analitikus gépet jellemzője: működtetése mechanikus lyukkártyával történt (a gép programozása és az adatok bevitelére lyukkártyán található)

1847 George Boole „A logika matematikai analízise” munkája, mely elméleti alapot adott a számítógép logikai tervezéséhez és programozásához.

1.2.2. Elektromechanikus eszközök

1880- as amerikai népszámlálás hatalmas adatait még nyolc évvel később sem tudták feldolgozni.

8

1889 Hermann Hollerith kidolgozza a lyukkártya mai formáját, és a lyukkártyás adatfeldolgozás elvét.

1890- es népszámlálás adatfeldolgozása 6 hétig tartott Hollerith gépén. A gép lyukkártyával működött, de a lyukkártyák olvasását villamos működtetésű egység végezte.

1936 Konrad Zuse bináris aritmetikai egységet épített jelfogók felhasználásával, majd

1941- ben megépíti az első lebegőpontos aritmetikával rendelkező jelfogós külső programvezérlésű gépet

9

1.2.3. Elektronikus eszközök

1943-45 első elektronikus számítógép megépítése a Pensylvániai Egyetemen. A gép neve: ENIAC jellemzője: nagyon sok ellenállást, reléket, kapcsolókat, és elektroncsövet tartalmazott.

1948 Neumann János az USA kormányának készített jelentésében (amely a tárolt programú elektronikus számítógép alapdokumentumának tekinthető) felvázolja azokat az elveket, amelyek alapján a mai számítógépek is működnek.

10

1949 Neumann elvű EDVAC típusú számítógép elkészítése.

Jellemzője:- általános célú
- elektronikus
- tárolt programú
- digitális
- binárisan működő } számítógép

1951 elkészül az UNIVAC az első sorozatban gyártott számítógép.

11

1.2.3. NEUMANN- elv

- az utasításoknak az adatokkal azonos módon, azaz a memóriában, numerikus formában történő tárolása,
- kettes számrendszer alkalmazása
- teljes elektronikus működés
- a vezérlőegység az utasításokat emberi beavatkozás nélkül értelmezi és hajtja végre,
- a számítógép tartalmazzon olyan számológépművet, amely képes elvégezni az alapvető logikai műveleteket,
- perifériák alkalmazása, a külvilággal történő kapcsolattartásra

12

2.1.Számrendszerek: a bináris és a hexadecimális számrendszer, számok ábrázolása

A kétállapotú fizikai elemek mint az adattárolás eszközei. Egyik állapotnak a nulla szám másik állapotnak az egyes feleltethető meg.

Elektromos áramköröknél a „bekapcsolt” ill. „kikapcsolt” állapotot ily módon stabilan lehet regisztrálni.

Az ilyen áramkörökből felépíthető digitális számítógépeknek az a megfelelő számrendszer amelynek csak két diszkrét állapota van.

A megoldást a kettes (bináris) számrendszer jelenti. Így egyetlen kétállapotú fizikai elem (pl. tranzisztor) alkalmas egy egyjegyű, kettes számrendszerbeli szám értékének tárolására.

2.1. A kettes (bináris) számrendszer

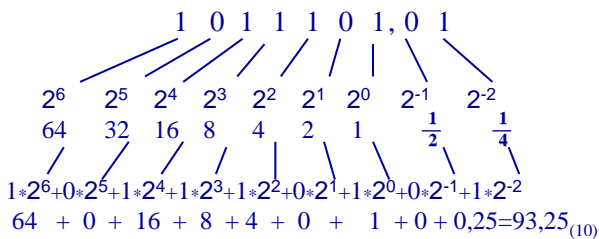
jellemzője: - a számrendszer alapszáma a kettő,

- két számjegyet tartalmaz: 0-át és 1-et,

-Az egyes pozíciókon az alapszám egész kitevőjű hatványainak az értéke szerepel,

tizes	0	1	2	3	4	5	6	7	8	9
kettes	0	1	10	11	100	101	110	111	1000	1001

$$1011101,01_{(2)} = \dots\dots? \dots\dots_{(10)}$$



Megállapítható, hogy a decimálisan (tizes számrendszerben) négy helyiértéken felírható szám binárisan 9 pozíciót igényelt, azaz lényegesen hosszabban lehetett ugyanazt az értéket kifejezni.

2.3. A Tizenhatos (Hexadecimális) számrendszer

jellemzője: - a számrendszer alapja a tizenhat,

- egy helyiértéken 16 különböző állapotot tudunk kifejezni,

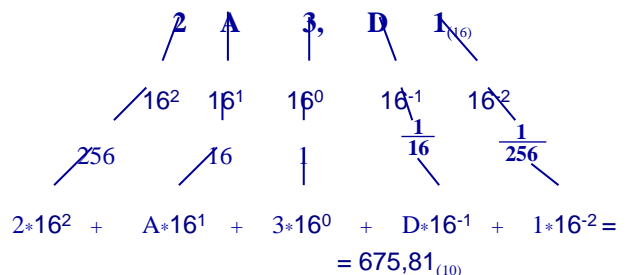
- a 16 számjegyet a decimális számrendszer számjegyein kívül a Latin abc első 6 nagybetűje szimbolizálja

A hexadecimális számrendszer számjegyei:

Hexadecimális	Decimális	Bináris	Hexadecimális	Decimális	Bináris
0	0	0	8	8	1000
1	1	1	9	9	1001
2	2	01	A	10	1010
3	3	11	B	11	1011
4	4	100	C	12	1100
5	5	101	D	13	1101
6	6	110	E	14	1110
7	7	111	F	15	1111

Az eddigiekben alkalmazott helyiértékek szerinti kifejezést felhasználva, írjuk fel decimálisan az alábbi hexadecimális szám értékét

$$2A3D1_{(16)} = \dots\dots? \dots\dots_{(10)}$$



$$2 * 16^2 \text{ azaz } 2 * 256 = 512$$

$$A * 16^1 \text{ azaz } 10 * 16 = 160$$

$$3 * 16^0 \text{ azaz } 3 * 1 = 3$$

$$D * 16^{-1} \text{ azaz } 13 * \frac{1}{16} = \frac{13}{16}$$

$$1 * 16^{-2} \text{ azaz } 1 * \frac{1}{16} = \frac{1}{256}$$

$$\text{összesen: } 675 \frac{208}{256}_{(10)}$$

$$2A3, D1_{(16)} = 675,81_{(10)}$$

Ha egy tetszőleges bináris számot négyes csoportokra osztunk a bináris vesszőtől balra és jobbra a négyes csoportoknak megfeleltethetünk egy - egy hexadecimális számjegyet

19

$$\text{pl. } \underbrace{\dots 11001}_{1 \ 9} \underbrace{101101}_{11 \ 6} \underbrace{10011}_{9 \ 12} \underbrace{100111}_{14}$$

$$1 \ 9 \ B \ 6 \ , \ 9 \ C \ E$$

Mint hogy $2^4 = 16$, a négy számjeggyel leírható bináris számok és az egy számjeggyel leírható hexadecimális számok között közvetlen a megfeleltetés.

A hexadecimálisan egy helyi értéken kifejezhető számok bináris leírásához négy pozíció szükséges. A 16-os számrendszert a számítógépek nagy része használja ugyanis a számjegyeket 4 biten tárolja.

20

Bináris	Hexadecimális	Bináris	Hexadecimális
0000	0	1000	8
0001	1	1001	9
0010	2	1010	A
0011	3	1011	B
0100	4	1100	C
0101	5	1101	D
0110	6	1110	E
0111	7	1111	F

2.4. Számrendszerek közti átváltás

A különböző számrendszerben levő számok egymásba átszámíthatók. (Konverzió)

21

Bármely számrendszerben azonos módon állítjuk össze a számot.

Legyen „Z” a számrendszer alapszáma. Az adott szám, ha háromjegyű (d_0, d_1, d_2 , az alacsony helyi értéktől kezdve), a szám általános alakja:

$$d_2 * Z^2 + d_1 * Z^1 + d_0 * Z^0$$

$$\text{pl. } 169_{(10)} = 1 * 10^2 + 6 * 10^1 + 9 * 10^0$$

$$1011_{(2)} = 1 * 2^3 + 0 * 2^2 + 1 * 2^1 + 1 * 2^0$$

A decimális szám egészrésze minden számrendszerben egész, a tört része pedig tört.

22

Ez lehetővé teszi az egész- és tört rész konverzióját külön-külön. A részeredményeket egyesítjük, így kapjuk meg a decimális szám, új számrendszerbeli megfelelőjét.

2.4.1. Decimális számrendszerből, bináris számrendszerbe való átalakítás

2.4.1.1. Egész számok esetén

Egész számok esetén az átváltás algoritmus a számrendszer alapszámával való osztás, míg nulla nem lesz a hányados. Minden osztás után az eredményt leírjuk és feljegyezzük a maradékot, amit egymástól függőleges vonallal választunk el.

23

A bináris számot úgy kapjuk, hogy a függőleges vonal jobb oldalán vezetett maradékot alulról fölfelé összeolvassuk.

Példa: $169_{(10)} = ?_{(2)}$

$$169:2 = 84 \longrightarrow 1$$

$$84:2 = 42 \longrightarrow 0$$

$$42:2 = 21 \longrightarrow 0$$

$$21:2 = 10 \longrightarrow 1$$

$$10:2 = 5 \longrightarrow 0$$

$$5:2 = 2 \longrightarrow 1$$

$$2:2 = 1 \longrightarrow 0$$

$$1:2 = 0 \longrightarrow 1$$

$$169_{(10)} = 10101001_{(2)}$$

$$\begin{array}{r|l} 169_{(10)} & :2 \\ \hline 84 & 1 \\ 42 & 0 \\ 21 & 0 \\ 10 & 1 \\ 5 & 0 \\ 2 & 1 \\ 1 & 0 \\ 0 & 1 \end{array}$$

Visszaírás: alulról ↑

24

2.4.1.2. Törtrész átalakítása

A törtrész átalakítása az alábbi algoritmus szerint történik: - a törtrészt szorozzuk kettővel,

- külön írjuk egymás alá a szorzat egész és törtrészét,
- az előzőleg kapott törtrészt szorozzuk tovább kettővel,

Mindezt addig folytatjuk amíg a szorzandó törtrésze "0" nem lesz, vagy a kívánt pontosságot el nem érjük, esetleg a számjegyek ismétlődnek

pl. $0,79_{(10)} = ?_{(2)}$ Az átalakítást 8 bináris jegyig végezzük.

$0,79 * 2 = 1,58$	→	1		$0,79_{(10)} * 2$
$0,58 * 2 = 1,16$	→	1		$1 \quad 58 * 2$
$0,16 * 2 = 0,32$	→	0		$1 \quad 16 * 2$
$0,32 * 2 = 0,64$	→	0		$0 \quad 32 * 2$
$0,64 * 2 = 1,28$	→	1		$0 \quad 64 * 2$
$0,28 * 2 = 0,56$	→	0		$1 \quad 28 * 2$
$0,56 * 2 = 1,12$	→	1		$0 \quad 56 * 2$
$0,12 * 2 = 0,24$	→	0		$1 \quad 12 * 2$
$0,79_{(10)} = 0,11001010_{(2)}$				$0 \quad 24$

kiolvasás

kiolvasás

2.4.2. A decimális számrendszerből-hexadecimális számrendszerbe történő konvertálás

Az átszámítás menete megegyezik a binárisra történő konvertálásnál megismertekkel. De itt az alapszám: 16

Először az egészrészt számítjuk át úgy, hogy mindig 16- tal osztjuk a decimális számot.

A törtrész átalakításánál fordítva járunk el, ugyanis a decimális szám törtrészét folyamatosan 16- tal szorozzuk

pl. $12438,964_{(10)} = ?_{(16)}$

A szám egész részét elosztjuk 16- tal: $12438 : 16$

$12438 : 16 = 777$ és marad 6,	12438	$:16$
A 777- et elosztjuk 16- tal:	777	6
$777 : 16 = 48$ és marad 9,	48	9
A 48- at elosztjuk 16- tal:	3	0
$48 : 16 = 3$ és marad 0	0	3

A hexadecimális szám egész része: $3096_{(16)}$
A decimális szám törtrészét megszorozzuk 16- tal

$0,964 * 16 = 15,424$	$0,964 * 16$	15	$424 * 16$
A 0,424- et szorozzuk 16- tal:	$F \leftarrow 15$	6	$784 * 16$
$0,424 * 16 = 6,784$	6	$784 * 16$	$544 * 16$
A 0,784- et szorozzuk 16- tal:	$C \leftarrow 12$	$544 * 16$	$12,544$
$0,784 * 16 = 12,544$			

Az eljárás tetszőlegesen sok lépésben folytatható a kívánt pontosságig

Tehát a $12438,964_{(10)} = 3096,F6C_{(16)}$

Az átalakítás könnyebben megoldható ha, a decimális számot minden esetben átalakítjuk binárisra, és ezt alakítjuk át hexadecimálissá.

$(10) \rightarrow (2) \rightarrow (16)$
 $(16) \rightarrow (2) \rightarrow (10)$

2.4.3. Bináris számrendszerbeli szám átalakítása hexadecimálissá és viszont

Mivel a 16 a 2^4 - ik hatványa ezért 4 bináris jegy egy hexadecimális számjeggyel helyettesíthető. Az átalakítás során a bináris vesszőtől kiindulva négyes csoportokat (tetradokat) kell létrehozni. A csoportoknak megfelelő hexadecimális számjegyeket csak egymás mellé írjuk.

pl. $1110111010,1010111_{(2)} = ?_{(16)}$

0011	1011	1010	1010	1110
3	B	A	A	E

$1110111010,1010111_{(2)} = 3BA,AE$

2.4.4. Hexadecimálisból-binárisra történő konvertálás

A hexadecimális alakból - binárisra történő áttérésnél minden hexadecimális jegyet egy tetrádon kell felírni és a bitnégyeseket visszaolvasni.

pl. $A81,FC_{(16)} = ?_{(2)}$

$$\begin{array}{ccccccc} \text{A} & 8 & 1 & , & F & C_{(16)} & \\ \hline \underbrace{1010} & \underbrace{1000} & \underbrace{0001} & , & \underbrace{1111} & \underbrace{1100}_{(2)} & \\ \hline A81,FC_{(16)} = 101010000001,11111100_{(2)} \end{array}$$

31

2.5. Aritmetikai műveletek a különböző számrendszerekben

Bináris aritmetika

2.5.1. Bináris összeadás menete:

- jobbról balra haladva összeadjuk az azonos helyiértékű számjegyeket,
- ha az összeg nem haladja meg a számrendszerben leírható maximális számjegyet, akkor leírjuk, (kettes számrendszerben 1; 16-ban 15)
- ha meghaladja kivonjuk belőle a számrendszer alapját, a maradékot leírjuk,

32

- az eggyel nagyobb helyiértéken álló számok összegéhez pedig 1-et hozzáadunk, azaz átvitel történik az eggyel magasabb helyiértékre,

- a műveletet jobbról balra haladva addig folytatjuk, amíg a számjegyek el nem fogynak.

Összeadáskor az alábbi esetek fordulnak elő, és az erre vonatkozó szabályok a következők:

$$\begin{array}{cccc} 0 & 0 & 1 & 1 \\ +0 & +1 & +0 & +1 \\ \hline 0 & 1 & 1 & 10 \end{array}$$

átvitel az eggyel nagyobb helyi értékre
érdemjegy számjegye

33

2.5.2. Bináris kivonás

A lehetséges négy bitpárosítás esetén a kivonásra az alábbi szabályok érvényesek:

Át hozat az eggyel magasabb helyiértékről
ami 2-öt „ér”

$$\begin{array}{cccc} 0 & 1 & 1 & 00 \\ -0 & -0 & -1 & -1 \\ \hline 0 & 1 & 0 & 1 \end{array}$$

- ha a kisebbbitendő számjegy nem kisebb mint a kivonandó, a kivonást elvégezzük, az eredményt leírjuk,
- ha kisebb, „kölcsonvesszük” az eggyel magasabb helyiértékű oszlopból egy kettest,

34

_____ A műveletet addig folytatjuk, amíg a számjegyek el nem fogynak.

Példa bináris összeadásra és kivonásra:

$$\begin{array}{r} 11 \\ \curvearrowright \quad \curvearrowright \\ 1101101,101 \\ - 111101,010 \\ \hline 0110000,011 \end{array} \quad \begin{array}{r} 1 \\ \curvearrowleft \quad \curvearrowleft \quad \curvearrowleft \quad \curvearrowleft \\ 1001,11 \\ + 110,11 \\ \hline 10000,10 \end{array}$$

A bináris összeadás és kivonás műveleti táblája:

+	0	1
0	0	1
1	1	0*

-	0	1
0	0	1
1	1*	0

*- al jelölt esetben átvitel ill. áthozatal történik.

35

2.5.3. Kivonás (komplementálással)

A negatív számok speciális ábrázolásával a kivonás az alábbiak szerint vezethető vissza összeadásra:

- negatív számnál képeznünk kell a szám kettes komplementjét.
- egy bináris szám egyes komplementjét úgy kapjuk, hogy a szám minden egyesének helyére 0-át, minden nullája helyére 1-et írunk.
- A kettes komplement képzése egy 1-es hozzáadásával valósul meg.

A számítógépek többnyire a kettes komplement képzésén alapuló eljárást használják kivonásra.

36

pl. $11001110 \rightarrow$ nyolcpozíciós bináris szám
 $00110001 \rightarrow$ a szám egyes komplementere
 $+00000001$
 $\hline 00110010 \rightarrow$ a szám kettes komplementese

ell. $11001110 \rightarrow$ eredeti szám
 $+00110010 \rightarrow$ kettes komplementer
 $\hline 10000000 \rightarrow$ a kettő összege

Ha a kettes komplementest hozzáadjuk egy tetszőleges nyolcbites számhoz az eredmény ugyanaz mintha az eredeti számot kivontuk volna belőle.

1110 1110 $\rightarrow 1010$ 1-es komplementens
 -0101 -0101 $+0001$ 1110
 $\hline 1001$ $\hline 1011$ $\rightarrow +1011$
 $\hline 1001$ $\hline 1001$

2.5.4. Bináris szorzás

*	0	1
0	0	0
1	0	1

A szorzás eredménye csak akkor 1-es, ha mindkét jegy egyes.

A szorzás nem más mint összeadások sorozata. Képezzük az egyes részletszorzatokat és összeadjuk.

pl. $011110 * 010011$ **ahogy a számítógép szorozza:**

$011110 * 010011 \rightarrow$ első szorzóbit
 000000 ❶ \rightarrow második szorzóbit
 $+011110$ \rightarrow harmadik szorzóbit
 $\hline 011110$ ❷
 $+011110$ ❸
 $\hline 1011010$ ❸
 $+011110$
 $\hline 1000111010 \rightarrow$ szorzás eredménye

2.5.5. Hexadecimális összeadás

Az oktális összeadásnál említetteknek megfelelően kell elvégezni, de a részösszeget itt 16- al kell osztani.

A hexadecimális számrendszerben lévő betűket a decimális értékükkel kell figyelembe venni

$1 \rightarrow 1$
 $A B C_{(16)} = 2748$
 $+ F D_{(16)} = 253$
 $\hline B B 9_{(16)} = 3001$

① a) $C_{(16)} + D_{(16)} = 25$ ($12+13=25$)
 b) $25/16 = 1$ (átvitel)
 maradék: 9

② a) $1 + B_{(16)} + F_{(16)} = 27$
 b) $27/16 = 1$ (átvitel)
 maradék: $11 = B_{(16)}$

③ $1 + A_{(16)} = B_{(16)}$

2.5.6. Hexadecimális kivonás

A kivonásnál is az egyes helyiértékeken decimálisan számolunk.

Ha a kivonás nem végezhető el egy adott helyiértéken, akkor alapszámnyi kölcsönt veszünk el a magasabb helyiértékről, ami itt most 16- ot ér.

$B 4 A 9_{(16)}$
 $- 8 A F 5_{(16)}$
 $\hline 2 9 B 4_{(16)}$

$B 4 A 9_{(16)}$
 $- 8 A F 5_{(16)}$
 $\hline 2 9 B 4_{(16)}$

3. Adatok ábrázolása a számítógépben

3.1. Kódrendszerek, ALFANUMERIKUS jelek ábrázolása

Számjegyek, betűk, írásjelek és egyéb jelek számítógépes ábrázolására bináris kódokat használunk.

Az ábrázolandó karakterek száma: 2^8 azaz 256 karakter.

A számítógépen egy karaktert egy bájtton ábrázolunk.

Nemzetközi kódszabványok biztosítják, hogy ugyanaz a bitkombináció ugyanazt a karaktert jelentse a világon.

Két fő számítógépes kódszabvány létezik:

- EBCDIC (Extended Binari Coded Decimal Interchange Code)
- ASCII (American Standard Code for Informacion Interchange)

Ezen kódszabványok tartalmazzák:

- angol ABC kis és nagybetűit,
- decimális számjegyeket, írásjelet,
- vezérlő karaktereket

A KGST országokban ezen szabványokat a cirill ABC kis és nagybetűivel egészítették ki, így:

- az EBCDIC- nek a DKOI, az ASCII- nek a KOI-8 szabvány felel meg.

Az ASCII kódrendszernek létezik a kibővített változata az ASCII-8. A személyi számítógépekben ezt használják.

- Jellemzője:
- 8 bites 256 különböző jel kódolására képes,
 - az első 128 jel rögzített, a második 128 helyen tárolt jelek változhatnak

43

- itt kapnak helyet az orszáfgüggő karakterek, pl. itt található az ékezetes betűk, hogy a programok képesek legyenek használni a magyar ABC- t.

- mivel országonként különbözők a betűk, az ASCII kódtábla több változatát használják:
 - pl. - az amerikai kódtábla 437- es jelű,
 - a kelet európai kódtábla 852- es jelű.

Ezen kódtáblák a második 128 jelben különböznek.

UNICODE: Sok nyelvnek több jele van, mint 256. Az IBM felállított egy olyan kódrendszert , amely

44

két bájtton ábrázolja a jeleket, azaz 65536 jelet különböztet meg.

Ez az UNICODE nevű rendszer, melynek első 256 jele megegyezik az ASCII- vel, a többiek kiosztása részben szabványos

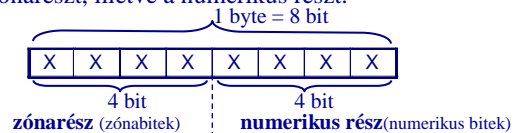
- EBCDIC:
- A központi egységben a karakterábrázoláshoz használatos,
 - A cirill betűkkel és ilyen formában írt vezérlőjelekkel kiegészített változata a DKOI,
 - Az EBCDIC kódban minden egyes karakter egy külön byte- ot foglal el.

45

Ennyi karakter ábrázolására nincs szükség, így lehetőség van a karakterkészlet csoportokba sorolására és a csoportok megkülönböztetésére.

Csoportot alkotnak a számjegyek, nagybetűk, kisbetűk.

Az EBCDIC a karakterek ábrázolására a byte- ot két részre, két 4 bites részre osztják, ezek alkotják a zónarészt, illetve a numerikus részt.



46

A zónabitek szolgálnak a karakterek egyes csoportjainak kijelölésére.

Az EBCDIC- kódban ábrázolt számokkal a gép számításokat nem tud végezni.

Alapvetően előjel nélkül, számok ábrázolására alkalmas. Az előjel, mint különleges karakter, külön kerül ábrázolásra.

47

3.2. Belső adatábrázolás

A számítógépben való adattárolást belső adatábrázolásnak nevezzük.

Gépi szó: Bináris jelek meghatározott véges sorozata amely, mint alapegység kerül a számítógépben feldolgozásra. (szg. Rendszerenként változó)

Egy változat:

1 byte = 8 bit

2 byte = 16 bit = félszó (FSZ)

4 byte = 32 bit = szó (SZ)

8 byte = 64 bit = dupla szó (DSZ)

48

Byte felépítése:



A tárolás alapegysége a byte, amely 8 bit tárolására alkalmas.

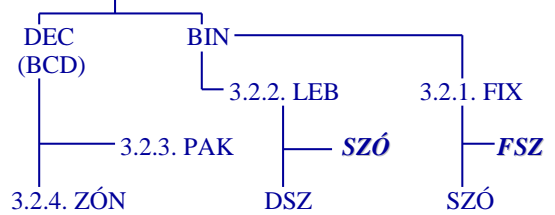
$$1 \text{ KB} = 2^{10} = 1024 \text{ byte}$$

$$1 \text{ MB} = 2^{20} = 1048576 \text{ byte}$$

$$1 \text{ GB} = 2^{30} = 1073741824 \text{ byte}$$

Egy 32 MB-os memória pontos mérete:
 $32 \cdot 1024 \cdot 1024 = 33\,554\,432 \text{ byte}$.

Belső számábrázolás



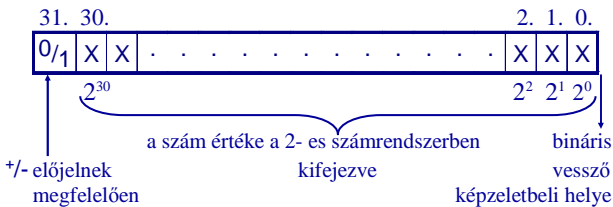
3.2.1. Fixpontos számábrázolás

Jellemzője: - fixpontos aritmetikai egysége majdnem minden számítógépnek van,

- minden fixpontos operandust egész számként kezel,
- a numerikus adatokat egész számoknak kell tekinteni, mert csak azok ábrázolhatók, törtek nem,
- tizedespon ábrázolására, és a számolás közbeni figyelésére nincs lehetőség,
- a tizedes vesszőt az operandus egy rögzített, fix helyére kell képzeletben tenni. Ez a fix pont a legutolsó ábrázolt szánjegy után következik,
- az ábrázoláshoz rendelkezésre álló hosszúság között
 - általában a gép által használt szó hosszával egyezik meg,

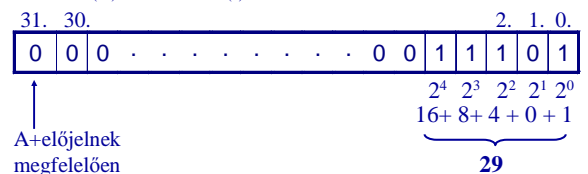
- egyes esetekben lehetőség van a szó hossz felének (*félszavas ábrázolás*) vagy
- a szó hossz kétszeresének (*duplaszavas ábrázolás*) használatára is,
- igen sok gépnél használatos a 32 bites (*szavas*) ábrázolási forma,
- ábrázolhatunk pozitív és negatív számokat,
 - ezért egy bit a 32-ből az előjel ábrázolására van fenntartva (*előjelbit*), ennek értéke pozitív szám esetén **0**, negatív szám esetén **1** lesz,
 - a többi pozícióhoz jobbról kezdve a kettes

számrendszer helyi értékei vannak hozzárendelve



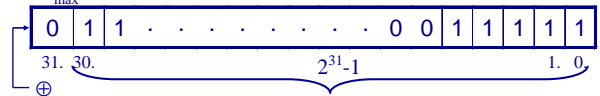
Az előjelábrázolás, vagyis a negatív számok ábrázolása a 2-es komplement (*komplement*) kódban történik.

pl. $+29_{(10)} = +11101_{(2)}$



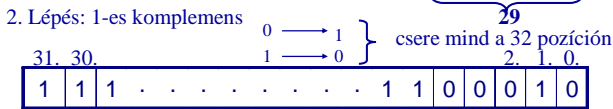
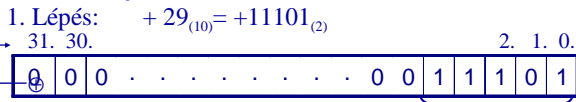
Az ábrázolható legnagyobb pozitív szám:

$$N_{\max} = R^n - 1 = 2^{31} - 1 = 2\,147\,483\,647$$

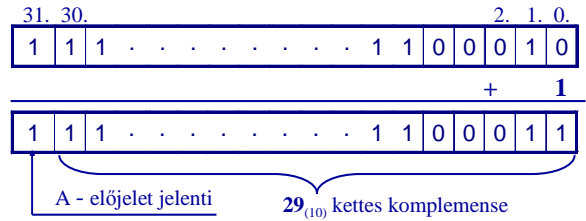


Negatív számok ábrázolása: pl. $-29_{(10)}$

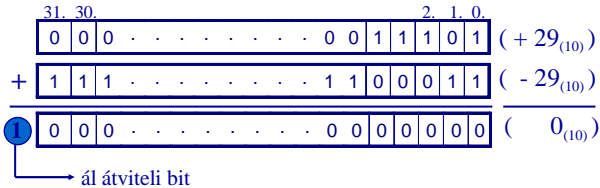
- 1. lépés:** a számnak megfelelő pozitív számot ábrázoljuk,
- 2. lépés:** invertáljuk az egészet, azaz minden pozícióban az ott lévő bitet ellenkezőjére változtatjuk.



3. lépés: a legalacsonyabb helyi értéken 1-et binárisan hozzáadunk, ez lesz az eredeti szám kettes komplementumbeli megfelelője,



Ell:



Az előjeles számok komplementum kódban történő ábrázolásának előnye, hogy minden művelet visszavezethető az előjel bittel együtt elvégzett összeadásra, ill. kivonásra.

$$18 - 5 = (+18) - (+5) \longrightarrow (+18) + (-5)$$

$$-23 - 2 = (-23) - (+2) \longrightarrow (-23) + (-2)$$

Ha két olyan nagy számot vonunk össze, hogy az eredmény már nem fér el az adott hosszban, pl. 31 biten, un. túlsordulás lép fel.

Ez az eset csak azonos előjelű (pozitív, vagy negatív) számok összevonásakor lép fel.

A túlsordulást az előjelbit indokolatlan megváltozása jelzi.

pl.

0010....0100101	(+.....)
+0110....0010010	+(+.....)
- előjel → 1000....0110111	

A kapott eredmény negatív holott két pozitív szám összeadásakor ez nem léphet fel, ugyanis az eredmény bináris jegyei nem fértek el 31 biten, „túlsordultak” a 32. Pozícióra, ami okozta az előjel indokolatlan megváltozását.

3.2.2. Lebegőpontos számábrázolás

Jellemzője:

- a fixpontos számábrázolás két hátrányos tulajdonságát küszöböli ki,
 - az ábrázolható számok tartománya kicsi,
 - az ábrázolt számok pontossága erősen korlátozott,

- bonyolultabb de pontosabb számábrázolás, amely több helyet igényel,
- a számtartomány lényegesen megnő és tört szám is ábrázolható,
- a számok hatványkitevős felírásán alapszik,

Bármely valós szám felírható:

$$a = + M \cdot p^k \text{ normál alakban,}$$

- ahol:
- a → az eredeti szám
 - M → mantissza
 - p → a hatvány alapja
 - k → a hatvány kitevője az un. karakterisztika

„M” mindig kisebb kell, hogy legyen mint 1, de a törtet nulla egészest tartalmazó számként fejezzük ki úgy, hogy a törtrész értékes jeggyel kezdődjön

$$1/p \leq M \leq 1$$

A hatvány alapja attól függően változik, hogy melyik számrendszerben - 10; 2; 16 - végezzük el a normalizálást,

A hatvány kitevőjének nagysága a (tizedes, bináris, hexadecimális) vessző eltolásának mértékét, az előjele pedig az eltolás irányát adja meg.

61

pl. $4,19_{(10)}$ normál alakja $0,419_{(10)} * 10^{+3}$
3 léptetés balra

$4,19_{(10)}$ normál alakja $0,419_{(10)} * 10^{+1}$
1 léptetés balra

$0,0419_{(10)}$ normál alakja $0,419_{(10)} * 10^{-2}$
2 léptetés jobbra

További példák a normalizálásra a kettes és 16- os számrendszerben

$$111001010_{(2)} = 0,111001010_{(2)} * 2^{+9}$$

$$10,00110_{(2)} = 0,1000110_{(2)} * 2^{+2}$$

62

$$-0,1000110_{(2)} = -0,1000110_{(2)} * 2^0$$

$$+0,000111011_{(2)} = +0,111011_{(2)} * 2^{-3}$$

$$E6AD_{(16)} = 0, E6AD_{(16)} * 16^{+4}$$

$$7D,82_{(16)} = 0, 7D82_{(16)} * 16^{+4}$$

$$-0,1BC3_{(16)} = -0,1BC3_{(16)} * 16^0$$

$$+0,000F03A_{(16)} = +0, F03A_{(16)} * 16^{-3}$$

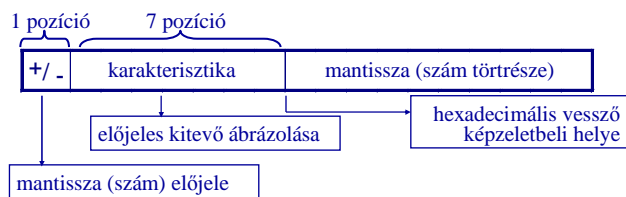
mantissza előjele mantissza karakterisztika

Lebegőpontos ábrázolásnál mindig a normál alakból indulunk ki.

A korszerű gépeknél általában 16- os alpra normalizálnak, azaz a hexadecimális vesszőt tolják el.

63

A szokásos ábrázolási forma:



A mantissza (szám) előjelének ábrázolása megegyezés szerint, pozitív mantissza esetén 0, negatív mantissza esetén 1- es bit kerül a legelső pozícióba.

64

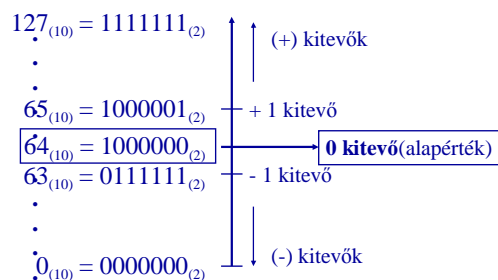
A karakterisztika (előjeles kitevő) ábrázolását alapértékes vagy feszített előjeles kitevő módján végzik, valamint még „64 többletes” kitevőábrázolásnak hívják.

Megegyezés szerint a rendelkezésre álló 7 pozíción a „0” kitevőt, ami az alapérték, az $1000000_{(2)} = 64_{(10)}$ segítségével ábrázolják. A 7 biten kifejezhető $2^7 = 128_{(10)}$ különböző állapot közepét, a 64-et önkényesen nullának, alapértéknek nevezték el.

A pozitív kitevőt ehhez az alapértékhez hozzáadják, a negatív kitevő értékét pedig ebből az alapértékből levonják.

65

7 pozíción ábrázolható állapotok száma:



66

pl. a +3 kitevő ábrázolása alapértékesen:

$$\begin{array}{r} \text{alapérték} \quad 1000000_{(2)} = 64_{(10)} \\ \text{kitevő 3 értékének hozzáadása} \quad + \quad 11_{(2)} = + 3_{(10)} \\ \text{a +3 kitevő ábrázolási formája} \quad 1000011_{(2)} = 67_{(10)} \end{array}$$

a -3 kitevő ábrázolása alapértékesen:

$$\begin{array}{r} \text{alapérték} \quad 1000000_{(2)} = 64_{(10)} \\ \text{kitevő 3 értékének levonása} \quad - \quad 11_{(2)} = - 3_{(10)} \\ \text{a -3 kitevő ábrázolási formája} \quad 0111101_{(2)} = 61_{(10)} \end{array}$$

A bináris vesszőt a 8. pozíció után kell elképzelni, így a mantissza jegyei a kilencedik pozíciótól kezdődik.

67

Az ábrázolás rögzített hosszon történik, rövid egyszeres pontosságú lebegő pontos alakban, ahol az ábrázolás hossza a gép által használt szó hossza 32 bit vagy hosszú lebegőpontos ill. duplapontos alakban, ahol az ábrázolás hossza a gép által használt szó kétszerese 64 bit.



68

Hosszú (duplapontos) lebegőpontos alak



pl. Ábrázoljuk rövid lebegőpontos alakban, 16- osra normalizálva a $-91_{(10)}$ - et.

$$-91_{(10)} \rightarrow -1011011_{(2)} \xrightarrow{5 \text{ B}} -5B_{(16)} \xrightarrow{\text{ábrázolandó}} -0,5B_{(16)} * 16^{+2}$$

A negatív előjel miatt a legelső pozícióba $1_{(2)}$ - es bit kerül

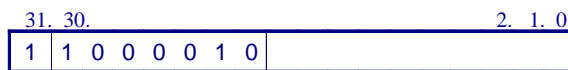
69



a „-” előjelenek megfelelően

A karakterisztika ábrázolása:

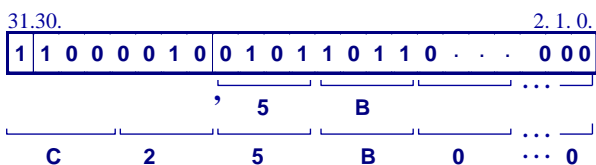
$$\begin{array}{r} \text{alapérték} \quad 1000000_{(2)} = 64_{(10)} \\ \text{kitevő 2 értékének hozzáadása} \quad + \quad 10_{(2)} = + 2_{(10)} \\ \text{a +2 kitevő ábrázolási formája} \quad 1000010_{(2)} = 66_{(10)} \end{array}$$



karakterisztika ábrázolása

70

A normalizált szám értékes jegyeit a mantissza tört részét bináris formában kell ábrázolni tetrádonként.



a $-91_{(10)}$, 16- osra normalizált rövid lebegőpontos alakja : **C25B0000**₍₁₆₎

71

pl. Ábrázoljuk rövid lebegőpontos alakban, kettes alpra normalizálva a $-91_{(10)}$ - et.

$$-91_{(10)} \rightarrow -1011011_{(2)} \xrightarrow{\text{ábrázolandó}} -0,1011011_{(2)} * 2^{+7}$$

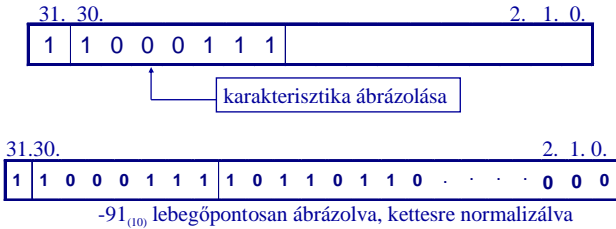


a „-” előjelenek megfelelően

72

A karakterisztika ábrázolása:

$$\begin{array}{r} \text{alapérték} \quad 1000000_{(2)} = 64_{(10)} \\ \text{kitevő 7 értékének hozzáadása} \quad + 111_{(2)} = +7_{(10)} \\ \text{a +7 kitevő ábrázolási formája} \quad 1000111_{(2)} = 71_{(10)} \end{array}$$

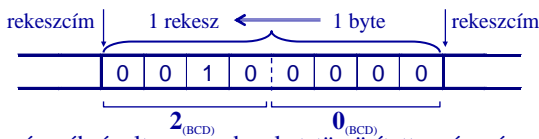


Decimális ábrázolás

- a numerikus operandusokat BCD számként ábrázoljuk,
- kevesebb a számrendszerek közötti konverzió, de a műveletvégzés ideje hosszabb,
- ez a számábrázolási mód adatfeldolgozási feladatoknál a legelterjedtebb ahol csak az alpműveleteket kell végezni,
- az ilyen operandusokat a decimális aritmetikai műveleteknél használják és feldolgozásukat a decimális aritmetika végzi,

3.2.3. Pakolt alak

Az operatív tárban a címezhető egységek, a rekeszek 8 bitet (1 bájt) képesek tárolni, egy ilyen rekeszbe „táregységben” egyidejűleg két binárisan kódolt decimális (BCD) számjegyet helyezhetünk el.

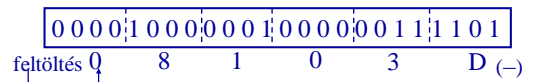


Az így ábrázolt operandusokat tömörített, más néven pakolt operandusoknak nevezzük. Az előjel ábrázolása a jobb félbájton történik.

Az előjelek kódja megegyezés szerint:

- + előjel $\rightarrow 1100_{(2)} = C_{(16)}$
- előjel $\rightarrow 1101_{(2)} = D_{(16)}$

pl. a - 8103₍₁₆₎ ábrázolása pakolt operandusként:
 - 5db félbájtot igényel (4db számjegy + egy előjel)
 - a gép páratlan félbájtot nem tud tárolni, az operandus 3 bájtot foglal el. A legmagasabb félbájtot nullákkal tölti fel.



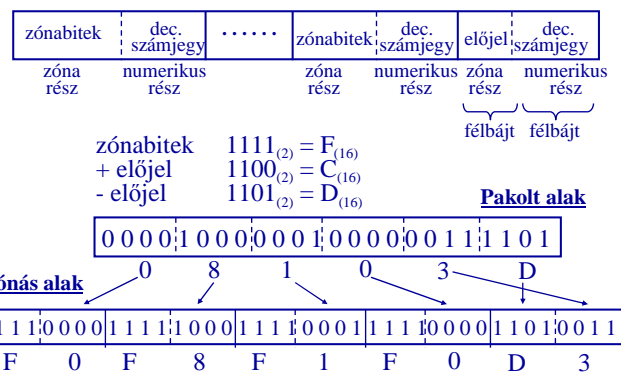
3.2.4. Zónás alak

A decimális számjegyeket 1 bájtton karakterenként ábrázoljuk.

Kivétel a legalacsonyabb helyi értékű bájt ahol a számjegy mellett az előjel van.

A bájtok jobb oldali felét numerikus résznek (itt vannak a decimális számjegyek binárisan kódolva) nevezzük, a baloldali félbájtot a zónarész.

A legalacsonyabb helyi értékű bájt zónarészében az előjel, a többi zónarészben a zónabitek találhatóak.



4. Logikai műveletek, logikai operátorok, igazságtáblák, De Morgan azonosságok

Minden logikai változónak (állításnak) két lehetséges állapota van:

- ha az állítás igaz: annak logikai értéke **IGAZ**, jelölése „**1**”
- ha az állítás hamis :annak logikai értéke **HAMIS**, jelölése „**0**”

Négy logikai alapművelettel foglalkozunk.

- **ÉS** (AND); a **VAGY** (OR); a kizáró **VAGY** (XOR); és a negálás, invertálás művelete a **NEM** (NOT).

79

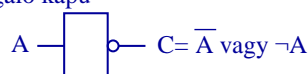
4.1. A logikai **NEM** művelet

Egyetlen bemenő eseményen végzett műveletvégzést jelent. Jelölése: \bar{A} ; vagy $\neg A$

A **NEM** művelet igazságtáblája:

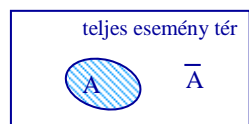
bemenő esemény A	kimenő esemény \bar{A}
0	1
1	0

Negáló kapu



80

Halmazelméleti értelmezés:



Nem kapcsolat áramköri értelmezése

4.2. A logikai **ÉS** kétoperandusú művelet

Jelölése: a logikai **ÉS** jelölésére a $(*)$ jele vagy a \wedge , valamint a \cap jel szolgál.

$$C = A * B, \text{ vagy } A \wedge B = A \cap B$$

81

Az **ÉS** műveletben szereplő logikai változók közül csak egynek az értéke **0** (HAMIS), a kimenő esemény logikai értéke is **0** (HAMIS) lesz.

A logikai **ÉS** kapu jele ill. **ÉS** művelet igazságtáblája:

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

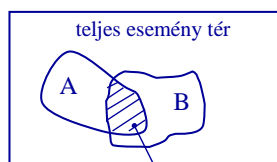
Az **ÉS** kapcsolat igazságtáblája



az **ÉS** kapu áramköri rajzjelei

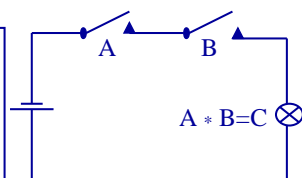
82

A logikai **ÉS** műveletet a halmazelméletben két halmaz közös részeként (metszeteként) értelmezik.



$$C = A \cap B$$

Az elektronikus eszközök területén kapcsolók sorbakötésével modellezhető az **ÉS** kapcsolat. Nyitott kapcsolóhoz logikai **0**-t, zárthoz logikai **1**-et rendelünk.



Az **ÉS** kapcsolat áramköri értelmezése

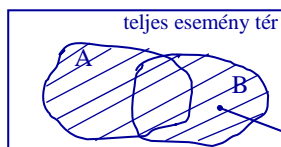
83

4.3. A logikai **VAGY** kétoperandusú művelet

Jelölése: az összeadás jele $(+)$; vagy a halmazelméletben a \vee ill. \cup jelekkel.

$$C = A + B = A \vee B = A \cup B$$

A két feltétel közül elég valamelyiknek bekövetkeznie ahhoz, hogy a **VAGY** kapcsolatban szereplő esemény logikai értéke **IGAZ** azaz **1** legyen.



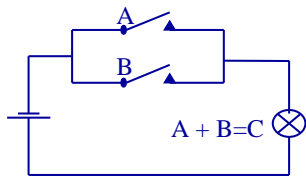
A halmazelméletben két halmaz egyesített területét (unióját) tekintik a **VAGY** kapcsolatnak.

$$C = A \cup B$$

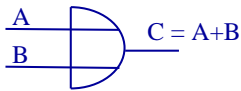
84

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

A **VAGY** kapcsolat igazságtáblája



A **VAGY** kapcsolat áramköri értelmezése



a **VAGY** kapu áramköri rajzjelei



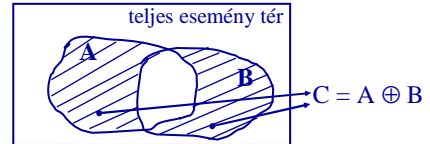
85

4.3. A **KIZÁRÓ VAGY** (XOR) kétoperandusú művelet

A **VAGY** művelettől csak annyiban különbözik, hogy a bemenő események egyidejű **IGAZ** (1) állapota esetén a kimeneten **HAMIS** (0) logikai eredményt ad. (kizárja a „bemenő események” egyidejű bekövetkezését)

Jelölése: \oplus ; szövegben: $C = A \oplus B$

A **KIZÁRÓ VAGY** halmazelméleti szemléltetése:



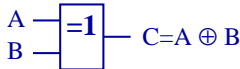
86

A számítógépek aritmetikai egységén belül a félösszeadó megvalósítására használják, mivel egy helyiértéken a bináris összeadást valósítja meg.

Bemenő események egyidejű megléte esetén **NEGÁLÁS** Összeállítható **ÉS**; **VAGY**; **NEM** kapukból

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

A **KIZÁRÓ VAGY** kapcsolat igazságtáblája



A **KIZÁRÓ VAGY** (XOR) kapu áramköri rajzjele

87

A fenti négy kapun kívül az integrált áramkörökben alkalmaznak **NAND**(nem és) valamint **NOR**(nem vagy) kapukat is.

4.4. De Morgan azonosságok

A programozási gyakorlatban ez fordul elő a leggyakrabban.

a) Bármely negált (tagadott) Boole- kifejezés egyszerűsíthető, ha az **ÉS** műveletet **VAGY**-ra változtatjuk vagy fordítva, és a változókat a negált értékükkel helyettesítjük.

$$\overline{A * B} = \overline{A} + \overline{B} \quad \text{A felülhúzás vonalát megszakítjuk,}$$

$$\overline{A + B} = \overline{A} * \overline{B} \quad \text{ha a műveleti jelet megváltoztatjuk.}$$

88

b) A nullás (hamis események) közötti kapcsolattól függetlenül az eredmény mindig **0**

$$0 * 0 = 0 \quad 0 + 0 = 0$$

c) A csupa „egyesből” álló bármely kombináció mindig „egyet” eredményez.

$$1 * 1 = 1 \quad 1 + 1 = 1$$

d) Az egymáshoz **ÉS** művelettel kapcsolt elemek sorozatában legalább egy elem **0**, a teljes kifejezés értéke is **0**.

$$0 * 1 = 0$$

e) Egy kifejezés logikai 1- gyel való szorzata egyenlő önmagával:

$$A * 1 = A$$

f) Egy kifejezés logikai 0- val való szorzata egyenlő nullával.:

$$A * 0 = 0$$

89

g) Az egymáshoz **VAGY** művelettel kapcsolt elemek sorozatában legalább egy elem **1**, a teljes kifejezés értéke is **1**.

h) Egy kifejezés logikai 1-gyel való **VAGY** kapcsolata egyenlő önmagával: $A + 1 = 1$

i) Egy kifejezés logikai 0- val való **VAGY** kapcsolata egyenlő önmagával: $A + 0 = A$

90

5. A hardver

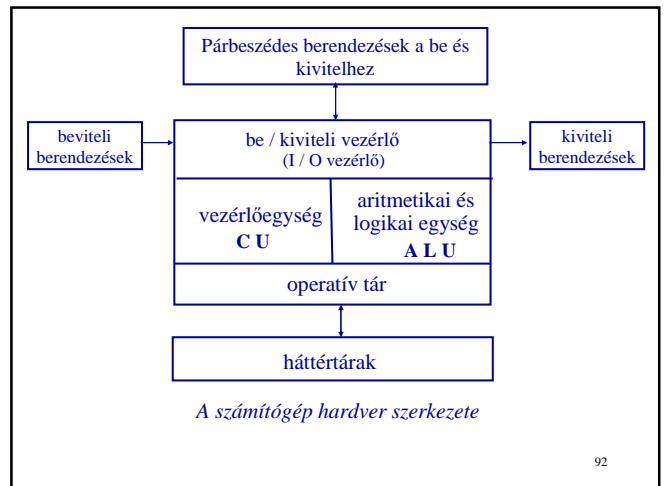
5.1. A számítógép felépítése, hardver blokk-vázlat, kapcsolódások

A számítógép feladata az adatok feldolgozása, melyhez az alábbi funkciókat valósítja meg:

- adatok bevitele
- adatok tárolása
- adatok feldolgozása
- adatok kivitele

A különböző funkciókhoz általában különböző berendezések tartoznak.

91



92

Központi egység : (CPU⇒ Central Processing Unit)

Feladata:

- a gép irányítása, illetve a feldolgozási folyamatvezérlése;
- az adatok feldolgozása;
- adatforgalom vezérlése a külvilág felé, ill. a külvilág felől ;

Részei:

a) Vezérlő egység (CU⇒Control Unit);

- irányítja a feldolgozás folyamatát,
- ütemezi a számítógép működését, belső órajel-generátora segítségével,
- program utasításainak végrehajtása,

93

b) Aritmetikai és logikai egység (ALU ⇒

Aritmetical and Logical Unit): aritmetikai és logikai műveletvégzés valamint a relációk kiértékelését végzi,

c) operatív tár vagy memória (OM ⇒ Operating Memory):

- az aktuálisan futó programokhoz szükséges adatokat és utasításokat tárolja,

d) be- és kiviteli vezérlőegység: (Input/Output unit: I/O), Funkciója: a perifériák és a központi egység közötti adatforgalom vezérlése

94

Háttértárolókon nagy adattömegeket, nyilvántartásokat tárolunk, a gyors működésű operatív tár viszonylag kevés adat tárolására alkalmas.

Beviteli- és kiviteli egységek az ember gép közötti kapcsolat, ill. adatok be- és kivitelt valósítják meg.

Párbeszédés berendezések: be- és kivitelt egyaránt végző párbeszédés üzemmódú terminálok.

A hardver fontos elemei az adatátviteli berendezések, melyet több gép összekapcsolása (hálózatok), vagy a központi egységtől nagy távolságra lévő perifériák használatánál alkalmazzuk.

95

Vezérlő egység (CU⇒Control Unit);

- irányítja a feldolgozás folyamatát,
- ütemezi a számítógép működését, belső órajel-generátora segítségével,
- program utasításainak végrehajtása:
 - minden utasításnak van egy műveleti kód része, mely alapján a vezérlő egység előállítja az utasítás végrehajtásához szükséges vezérlőjeleket,
 - a CU elektronikus vezérlőjeleket küld az ALU- nak, az operatív tárnak, I/O művelet esetén a be/kiviteli vezérlőegységnek, és utasítja azokat a kijelölt műveletek végrehajtására és ellenőrzi a művelet befejeződését,
 - Az utasítások végrehajtásához regiszternek nevezett átmeneti tárolókat is felhasznál,

96

A processzor mellett fontos működési egységeket ma már külön tokokba építik:

Operatív tár vagy memória (OM \Rightarrow Operating Memory):

- az aktuálisan futó programokhoz szükséges adatokat és utasításokat tárolja,
- a vezérlő egység által közvetlenül címezhető, elérhető tároló,
- rekeszekből (melyek önállóan címezhetők) és ezeket felépítő cellákból állnak,
- a legkisebb címezhető táregység a bájt,
- két alapvető része: a) állandó tár ROM
b) tényleges operatív tár RAM

97

- a tárat rekeszekre osztják fel és minden rekesznek van egy sorszáma,
- a rekesz sorszámát nevezzük a rekesz címének,
- a rekesz mérete géptípusonként változik 8 bitől - 64 bitig terjed, a 8 bites rekeszből felépített tárat bájtstruktúrájának nevezzük,
- a rekeszbe információt beírni vagy kiolvasni úgy lehet, ha előbb megcímezzük az illető rekeszt,
- a címzés azonos a rekesz megkeresésével,
- az operatív tár sem egységes: kis kapacitású, de rövidebb elérési idejű gyorsítótárból, és az eredeti operatív tárból épül fel,

98

Be- és kiviteli vezérlőegység: (Input/Output unit: I/O),
Funkciója: a perifériák és a központi egység közötti adatforgalom vezérlése

Az elektronikus vezérlő-jeleivel kiválasztja (*megcímszi*) a megfelelő perifériát, kiépíti a kapcsolatot és vezérli az átvitelt, miután a CU aktiválta az I/O vezérlőt és átadta a lebonyolításhoz szükséges információkat, a be/ kiviteli egység önállóan képes működni.

Programok párhuzamos végre hajtásának ez az alapja,

99

Regiszterek: A sok regiszterrel rendelkező központi egységekben a regiszterek halmaza gyors működésű tárként fogható fel, amely a központi egység és a tényleges operatív tár közötti adatcsere érdekében igyekszik gyorsítani.

- kis elérési idővel rendelkező átmeneti tárolók
- gyorsabban elérhetők mint a memória, a CPU-ba vannak beépítve,
- a processzor és a regiszterek működési sebessége azonos és szinkronban vannak,
- ha egy regiszter tartalmára szükség van, akkor az azonnal rendelkezésre áll,

100

Általános regiszterek:

- akkumulátor (ACC) a műveletek operandusait és a részeredményeket tárolja,
- bázisregiszter(BX)

Vezérlő regiszterek:

- veremmutató (SP) a részeredmények, hívási pontok, stb. tárolására szolgáló veremtár tetejére mutat,
- utasításmutató, (*programszámláló*) (IP) a soron következő utasítás címét tárolja,
- jelző (flag): egy művelet elvégzése során bekövetkezett változásokról tudósít, pl. az eredmény előjele, paritása, történt-e túlsordulás, kell-e átvitel,

101

Aritmetikai és logikai egység (ALU \Rightarrow Arithmetical and Logical Unit): aritmetikai és logikai művelet elvégzése valamint a relációk kiértékelését végzi,

102

6. A processzor

A **központi feldolgozó egységen** belül az egyes tevékenységeket önálló részek végzik. Az alábbi részeket együtt a *processzornak* is nevezzük. (CU+ALU+Regiszterek);



103

Órajelfrekvencia - órajel, a gép munkaiteme Megahertzben (MHz), másodpercenként hány műveletet hajt végre, melyet az órajelgenerátor határoz meg.

Órajel: egy rövid ideid tartó, de rendszeresen ismétlődő feszültségimpulzus.

Órajelfrekvencia: egy másodperc alatti órajelek száma amely meghatározza a processzor műveletvégző sebességét, amellyel adatokat dolgozhat fel.

Regiszterek mérete, a szó hossz amivel a processzor dolgozik ,

104

Napjainkban a 64 bites processzorok a legelterjedtebbek.

A belső szóhossz mellett, amellyel a processzor dolgozik fontos a **buszrendszer szóhossza**.

- adatbusz bitszélessége: a processzor hány bitet tud egyidejűleg a hozzá kapcsolt perifériákra küldeni,
- címbusz bitszélessége: meghatározza a közvetlenül megcímezhető címtartomány nagyságát,
- belső felépítés: pl. a Pentium nemcsak a nagyobb órajel- frekvencia miatt gyorsabb, hanem mert kialakítása lehetővé teszi több gépi utasítás egy ciklusban történő végrehajtását.

105

A processzor közvetlenül csak a gépi kódú program utasításainak végrehajtására képes.

- gépi kódú utasítás áll: elvégzendő műveletek kódjából és a művelet operandusaiból,
- a műveleti kód adja meg a processzor számára, hogy milyen műveletet, az operandusok pedig, hogy mivel kell a műveletet elvégezni,

6.1. Főbb jellemzői:

utasítás rendszer, utasításkészlet
címezési rendszer, módszer
megszakítás rendszer
védelmi rendszer

106

Utasítás rendszer

Az utasítás egy egyértelmű specifikál rendelkezés a számítógép felé amely operációhoz adatot/ adatokat rendel hozzá.

Az adatokat operandusoknak nevezzük.

Valamennyi utasítás, amelyet a számítógép végrehajthat egyértelmű formában íródik és ezeket az utasításlista foglalja össze.

Utasításkészlet: a processzor által értelmezhető műveletek összessége, azon utasítások, melyeket a CPU végre tud hajtani, 1,2,3 vagy esetleg 0 operandusú gépi utasítások,

107

Felépítése: műveleti kód + operandusok (adatok, regiszterek, de el is maradhat)

- az utasításkészlet processzor függvénye.
- az utasításszerkezet gépenként igen változatos
- legtöbb utasítás egy utasításkódból, és egy vagy több címből áll, valamint valamilyen járulékos információ is szerepel az utasításban, (címezési mód, indexelés,...)

Művelet kód	Címezési mód	Címek
-------------	--------------	-------

108

A processzor csak azt tudja megcsinálni, amit megmondunk neki. Két számot akkor és csak akkor tud tud összeadni, ha megmondjuk neki, hogy hol találja az egyik és hol a másik számot, mit kell vele csinálni és az eredményt hova rakja.

Valamennyi utasítás un. utasításszóban kerül ábrázolásra, melynek klasszikus formája a következő:

A 4- címes utasításszerkezet

Műveleti kód	1. Oper. címe	2. Oper. címe	Eredmény címe	Következő utas. címe
MIT ?	MIVEL ?	HOVA ?	HONNAN ?	

109

CPU működési állapotok:

- normál üzemmód: az utasításkészlet csak egy rész érhető el
- privilegizált üzemmód: minden utasítás elérhető és végrehajtható

Operációs rendszer: mindig privilegizált módban működik - ezt nevezzük Supervisor vagy Master módnak

Felhasználói programok: mindig normál módban használja a CPU-t

I/O műveletek csak privilegizált állapotban hajthatók végre

110

RISC és CISC típusú processzorok

A processzor teljesítményét alapvetően az határozza meg, hogy az utasításokat milyen sebességgel lehet végrehajtani és ezek mekkora változást eredményeznek a rendszer működésében.

A sebesség növelésnek két módja van:

- az utasítások végrehajtási idejét rövidítjük le, (egy utasítás végrehajtási idejét lecsökkentjük, mintha a processzor órajelét megnöveltük volna)
- az utasítások komplexitását növeljük meg,

111

A komplexitás azt jelenti, hogy egy utasítás végrehajtásának eredménye egyezzen meg akár több utasítás egymás utáni végrehajtásának eredményével.

Ezek alapján a processzorok két irányba fejlődtek:

RISC (Reduced Instruction Set Computer), csökkentett utasításkészletű számítógép.

CISC (Complex Instruction Set Computer), csökkentett utasításkészletű számítógép.

112

CISC processzorok jellemzője:

- utasításkészlete bőséges,
- utasítások többsége összetett,
- egy utasítás lehetőség szerint több funkciót is magában foglal,

RISC processzorok jellemzője:

- jelentősen szegényebb utasításkészlet,
- ezen utasításokat rendkívül gyorsan és nagy hatékonysággal képesek végrehajtani,

Az egyszerűbb utasításkészlet gyorsabb dekódolást tesz lehetővé, ami nagysebességű utasítás-végrehajtást eredményez .

113

A csökkentett utasítás készletet úgy határozzák meg, hogy minden utasítás egyforma hosszúságú legyen, és egyetlen gépi utasításciklus alatt végrehajtható.

A PC- k esetében a CISC processzorok terjedtek el,

A nagy teljesítményű szerverekben és grafikus alkalmazásokat futtató gépekben elsősorban RISC processzorok végzik a műveleteket.

A RISC processzorok azonos órajel mellett általában nagyobb sebességgel képesek futtatni az alkalmazásokat, mint a CISC processzor.

114

RISC: nagyságrenddel kevesebb utasítást tartalmaznak mint a CISC processzorok, viszont azokat sokkal gyorsabban sőt sok esetben párhuzamosan hajtják végre (superskalár processzorok). Speciális igényeknek jobban megfelel, hatékonyságuk nem minden területen megfelelő.

Bővített utasításrendszer: meglévő utasítások felhasználásával mikroprogramokat lehet írni, s ezeket mint új utasításokat használhatjuk.

115

Többprocesszoros rendszerek

Olyan rendszerek, melyben több processzor dolgozik, főleg RISC processzorokkal alkalmazzák ezt

- egy közös tárban több processzor dolgozik
- számítógép-hálózatok

a) van egy kitüntetett processzor, amely vezérli a többi processzort, és elvégzi az alapfunkciókat (tárkezelés, I/O szervezés, taszk-ütemezés); adott esetben egy - a fő - processzort csak az operációs rendszer használja, a többi osztják szét a felhasználói folyamatok között

116

b) back-end processzor kapcsolat: egy főprocesszor bírja munkára a társprocesszort vagy processzorokat, ez alárendeltségi viszonyt mutat (pl. grafikus vagy matematikai co-processzori megvalósítások)

c) a sok processzor teljesen egyenrangú, és az operációs rendszer osztja szét közöttük a feladatokat a megfelelő ütemező algoritmus segítségével (maga az operációs rendszer is az épp szabad processzort használja)

117

Bármelyik processzor indíthat I/O műveletet, amely megszakítást okoz azon a processzoron, amelyik fogadni tudja a megszakítást (ütemezés, szinkronizálás)

Nem lehet tudni, hogy egy adott felhasználó programja melyik processzoron fut

Az időszelvény végén a program az új aktivizálódáskor bármelyik processzoron futtat tovább, és folytathatja a munkát

118

Egy processzorhiba esetén egyik program sem szakad meg, hanem a hibát okozó processzor leáll, és valamelyik másik átveszi a rajta futó program végrehajtását (mintha csak az időszelvény járt volna le)

Sajnos ezek a megoldások ma még túl drágák, jobbak a katonai rendszerekben, az űrutatásban, s a folyamatvezérlésben használják (vegyszer-üzemek, atomreaktorok)

Bár helyenként megfigyelhetők a közlekedéstechnikában is (autók, repüléstechnika, automata metrő-irányítás)

119

7. A gépi kódú utasítások

A gépi kódú utasítás binárisan kódolt vezérlő információ, amely a processzort valamilyen művelet végrehajtására utasítja.

A gépi kódú programok is elemi részekből állnak ezek a gépi kódú utasítások. A gépi utasítás meghatározott terjedelmű bitsoportosítás melyek különböző műveleteket jelentenek, vagy pedig a számítógép regisztereire, operatív tárára, ... stb. vonatkoznak.

Mivel minden gépi utasítás bitekből épül fel, a gépi kódú program tisztán bináris formájú, 1- esekből, és 0- ákból áll.

120

Egy - egy számítógépre vonatkozólag a gépi utasítások összességét a számítógép gépi nyelvének nevezzük. A gépi nyelv általában kötött.

Egy gépi kódú program csak akkor futtatható egy adott számítógépen, ha a gép „érti” ezt a programot, vagyis az illető gép „nyelvén” írták.

A gépi utasítások válfajai:

Annyiban hasonlítanak a mikroutasításokhoz, hogy valamilyen formában tartalmazzák az elvégzendő műveletet kódolt formában, valamint az operandusok és az eredmény címét és utalnak a következőként végrehajtandó gépi utasítás címére.

121

A legteljesebb utasításszerkezet a műveleti kódon kívül négy címet tartalmaz.

A 4 - címes utasításszerkezet

Műveleti kód	1. Oper. címe	2. Oper. címe	Eredmény címe	Következő utas. címe
MIT ?		MIVEL ?		HOVA ? HONNAN ?

Az ilyen utasításokból álló program nagy helyet foglal a tárban. A mai gépeken az utasítások nincsenek „láncra fűzve” ezeket a korai számítógépeken használták.

122

A 3 címes utasítás annyiban egyszerűsödött az előzőhöz képest, hogy nincs utalás arra, hogy hol kell a programot folytatni. Itt feltételezzük, hogy a vezérlő egység olyan sorrendben foglalkozik az utasításokkal, ahogyan azok az operatív tárban egymás után következnek. A mai számítógépeken nem használják, elavultnak tekinthető.

A 3 - címes utasításszerkezet

Műveleti kód	1. Oper. címe	2. Oper. címe	Eredmény címe
--------------	---------------	---------------	---------------

123

A 4 és 3 címes utasítások hibája a nagy helyfoglalás ésszerűbb műveletszervezéssel ugyanaz az eredmény rövidebb utasításokkal is elérhető.

A korszerű számítógépek utasításai kétcímesek, azaz a műveleti kódon kívül csak a két operandusra történő hivatkozást tartalmaznak.

A 2 - címes utasításszerkezet

Műveleti kód	1. Oper. címe	2. Oper. címe
--------------	---------------	---------------

124

Az első két cím helyet közvetlen adatok is szerepelhetnek, de közvetlen adatok használata nem célszerű a programokban mert nem tudjuk azokat változtatni, míg a tárban lévő adatok tetszés szerint változtathatók.

Az operatív táron kívül, speciális rendeltetésű tárolórekesz (regiszter) is van a vezérlőegységben.

A regisztereknek saját címük van melyek szerepelhetnek az utasítás operandusrészében. A regiszterekkel végzett utasítások igen gyorsak.

Az eredmény címe lehet az operatív tár egy rekeszének címe, ha továbbra is szükség van az eredményre, az ALU egy regiszterében is maradhat.

125

Az 1+1 címes utasítás: már csak egyetlen operandust címezhetünk a másik már eleve a vezérlőegység valamelyik regiszterében áll. Külön utasítással kell oda betölteni a másik operandust műveletvégzés előtt, az eredmény a vezérlőegység regiszterében képződik.

Az 1+1 - címes utasításszerkezet

Műveleti kód	Operandus címe	Következő utasítás címe
--------------	----------------	-------------------------

126

Az 1- címes utasítás: némely egyszerű (8 bites) processzoroknál egyetlen operandust címezhetünk, az eredmény a vezérlő egység regisztereiben képződik. Az egyik operandus az ALU egy speciális akkumulátor regiszterében, a másik az operandusrészben megadott tárcímen, az eredmény az akkumulátor regiszterbe íródik, és feltételezzük, hogy az utasítások feldolgozása előfordulásuk sorrendjében történik.

Az 1 - címes utasítás szerkezet

Műveleti kód	Operandus címe
--------------	----------------

127

A 2 - címes utasítás ma a leginkább elterjedt utasításforma, ahol a rendszerben az utasítások nem jelölik ki a végrehajtandó utasítást, helyette egy speciális rekeszben (utasításszámláló, utasításregiszter, program counter) tárolja a következő utasítás címét, és az aktuális művelet elvégzése után, az utasításszámlálóban lévő cím alapján választja ki, a következő utasítást.

Ehhez szükséges, hogy:

- a program utasításai a memóriában egymás után,
- a végrehajtás sorrendjében helyezkedjenek el,
- a következő utasítás tárcímének kiszámításához a processzornak ismerni kell az aktuális utasítás hosszát.

128

Programvégrehajtás menete:

1. A betöltő (program loader) betölti a végrehajtandó gépi kódú programot a tár egy rekeszétől kezdve,
2. A betöltési címet elhelyezi az utasításszámlálóban
3. A processzor lehívja a tárból az utasításszámláló által kijelölt utasítást. A műveleti kód megadja, hogy hány bajtot foglal el az utasítás.
4. A lehívott utasítást végrehajtja, ha nem az utolsó volt, betölti a következő utasítás címét az utasításszámlálóba. Ha az utolsó utasítás volt a program leáll, ha nem,
5. Folytatás a 3. ponttól

129

A vezérlés átadó vagy ugró utasítások célja: megváltoztatják a programutasítások tárolási sorrendjét.

pl. ha egy logikai kifejezés igaz vagy hamis értékétől függően más- más helyen kell a végrehajtást folytatnunk, vagy ha egy utasítássorozatot át akarunk lépni.

A 2- címes utasításokban szokás, hogy az eredmény címe az 1. operandus helyén van, ha az 1. operandus értékére még szükség van, át kell írni végrehajtás előtt egy másik tárrekeszbe.

130

Input / Output műveletek

A mai korszerű számítógépek nagy műveleti sebessége és a perifériás egységek viszonylagos lassúsága közötti feszültség nélkülözhetetlenné tette, hogy a rendszerben folyó I/O műveletek önálló vezérlést kapjanak. Így tehermentesítjük a processzort az I/O műveletek felügyeletétől, és áthidaltuk a sokféle periféria különböző műveleti sebességéből fakadó problémákat.

Az adatok és programok átvitele a perifériás egység és a központi egység között adatátviteli berendezések segítségével valósul meg.

131

Két legjellemzőbb eszköz: - csatornarendszer
- sínrendszer

Csatornarendszerű adatátvitel

Információátvitel a csatornarendszer mellett, a központi egység operatív tára és a periféria között zajlik. Az átviteli folyamat vezérlése sok művelettel jár ezért célszerű volt különválasztani a központi processzortól.

Az átviteli folyamat vezérlésére külön processzort építettek be, ez a csatorna.

Csatorna feladata: - az átviteli folyamat vezérlése,
- a nagysebességű processzor és a lassú periféria közötti összhang megteremtése,

132

A perifériális egység munkáját a perifériavezérlő szervezi és irányítja, mely közvetlen kapcsolatban áll a csatornával, és értelmezi annak utasításait és végrehajtja azokat a rákötött perifériákkal, ill. küldi a csatornának az adatátvitelre vonatkozó információt.

Egy perifériavezérlő több periféria munkáját is irányítja és egy csatornához csatlakoznak. Lehetőség van arra is, hogy egy vezérlő több csatornán keresztül is elérhető legyen akár több számítógépről is.

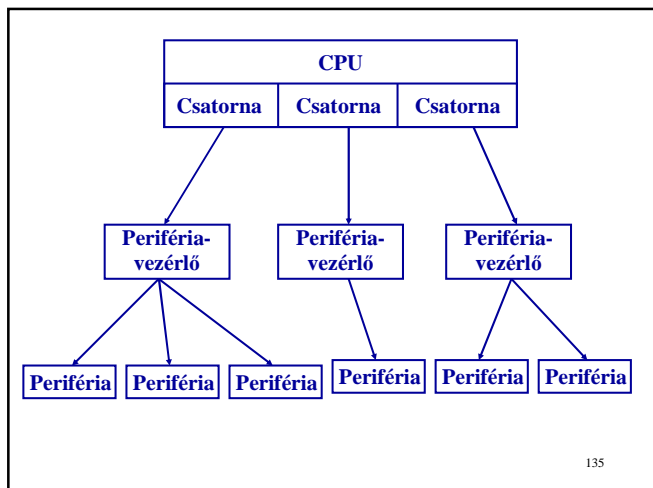
133

A csatorna ugyan úgy programozható mint a központi egység de a felhasználónak a csatorna programozására nincs lehetősége, mert a csatornaprogram a gép része.

A konkrét adatforgalom az operatív tár és a perifériák között megy végbe. A CPU az átvitelt miután kezdeményezte felszabadul és más műveleteket végezhet.

Csatornák üzemmódjai : - monopol üzemmód
- multiplex üzemmód

134



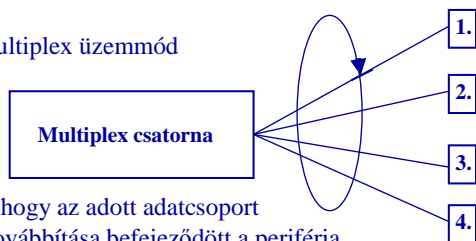
135

monopol üzemmód: a periféria a teljes adatátvitel tartalmára leköti a csatornát, a csatornára más periféria nem kapcsolódhat az átvitel befejező-déséig. Ha csak leköti, de tényleges adatátvitel nincs a gép hibát jelez.

multiplex üzemmód: a csatorna egyidejűleg több perifériát szolgál ki, ahol egy periféria csak egy rövid időintervallumra kapja meg a csatornát. Egy időpillanatban csak egy periféria van kapcsolatban a csatornával, ezután a csatorna „tovább” megy, ellátja a többi perifériát, majd ciklikusan visszatér.

136

A multiplex üzemmód



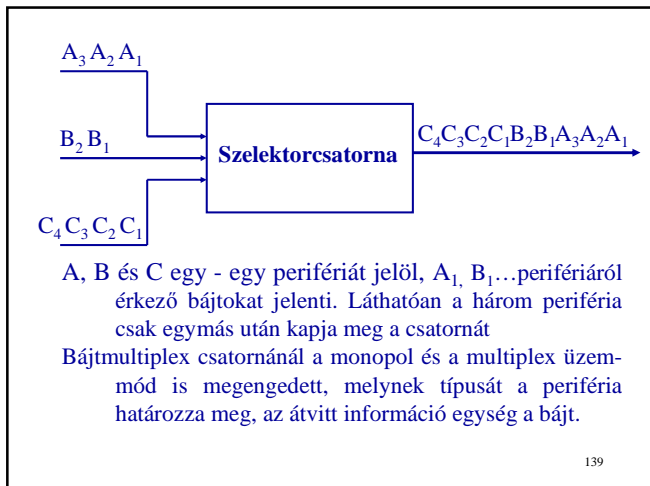
Azt, hogy az adott adatszoport továbbítása befejeződött a periféria megszakításkérésrel jelzi a csatornának
A csatornákat - működésük szerint - osztályokba sorolhatjuk: - szelektor csatorna
- bájtmultiplex csatorna
- blokkmultiplex csatorna

137

Szelektorcsatorna:

- csak monopol üzemmódban tud dolgozni ,
(ha valamelyik perifériával adatátvitelt kezdeményezett, akkor az a periféria „kisajátítja” monopolizálja magának a csatornát)
- egyszerre egész adatblokk, vagy ezek sorozatának átvitelére szolgál,
- Ez az üzemmód alkalmas a mágneses perifériák működtetésére, mert ezeknél a nagy adatsűrűség és átviteli sebesség miatt a megkezdett I/O műveletet nem lehet adatvesztés nélkül megszakítani
- jóval nagyobb átviteli sebességet nyújt, mint a multiplex csatorna.

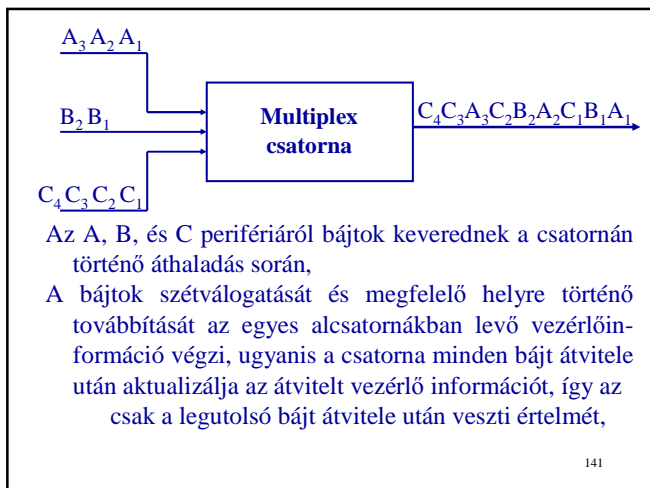
138



Blokkmultiplex csatorna:

- mind monopol mind multiplex üzemmódban képes működni,
- a legkisebb átvitt adatcsoport az adatblokk, mely akár néhány ezer bájt is lehet,
- az adatblokk átvitelére a periféria monopolizálja a csatornát, majd ezután a periféria dönti el hogy folytatódik-e a multiplexelés vagy nem,
- egy - egy periféria átviteli művelete időben szét van darabolva, emiatt tűnik úgy, mintha a multiplex csatorna egyidőben több perifériával dolgozna,
- szorosabb értelemben mindig csak egy perifériával foglalkozik a csatorna,

140



A felügyelő program:

- az egész adatátvitelt irányító felügyelő program (supervisor)
- a supervisor az operációs rendszer központi magja tartalmazza a megszakításkezelő alprogramokat is,

Feladata:

- a különböző programokban kezdeményezett adatátvitel megindítása,
- beérkező megszakítások kezelése, ezen keresztül a műveletek koordinálása,
- az esetleges hibák javítása, ha ez nem lehetséges a gépközvetítő értesítése a hibáról

142

Sínrendszerű adatátvitel:

- a sínrendszer esetén bármely , a rendszerre kapcsolódó funkcionális egység vezérelheti a rendszert, míg a csatornarendszerben központi, hierarchikus irányítás vezérli a műveleteket,
- a sínrendszer egy közös adatátviteli berendezés :
címvonalak, adatvonalak, vezérlők összessége,
- a sínrendszerre kapcsolódnak:
 - a központi vezérlő egység
 - a periféria vezérlők
 - egyéb csatlakozó egységek

143

Funkcionálisan a sínrendszerre csatolt összes egység egyenrangú, bármelyik irányíthatja a teljes folyamatot.

- a csatornarendszernél meglévő processzor intelligenciáját (vezérlését) az egyes egységekben elosztva kell kialakítani

Az alábbi előnyök miatt a sínrendszer korszerűbb a csatornarendszernél:

1. Csatornarendszernél az adatok az alábbi útvonalon közlekednek:
processzor - operatív tár - csatorna - periféria kezelő mindkét irányba,

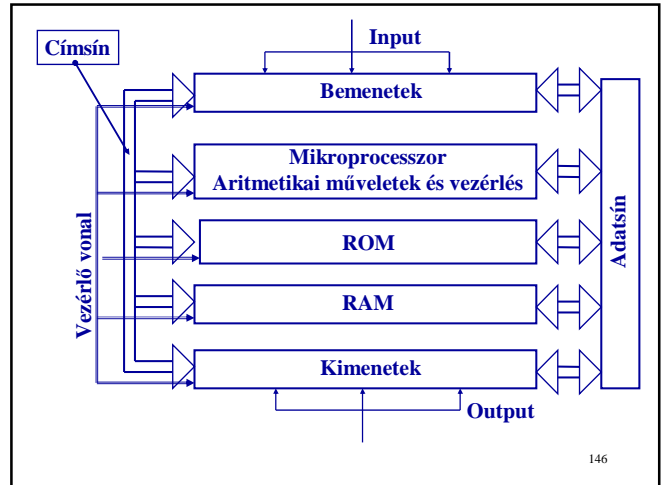
144

2. Sínrendszer esetén lehetőség van:

- központi processzor - perifériák közötti közvetlen átvitelre, operatív tár igénye nélkül,
- tár - perifériák közötti közvetlen kapcsolatra a processzor igénybevétele nélkül,
- egyes perifériavezérlők közvetlenül cserélhetnek adatokat a CPU igénybevétele nélkül,
- az I/O utasítások megszűnnek a sínrendszerű adatátvitelnél, mert a perifériavezérlők regiszterei ugyanúgy kezelhetők programból mint az operatív tár rekeszei,

A sínrendszerben az átvitel, az egész gép működésének koordinálása, a megszakítási rendszerrel történik.

145



146

Az input/output műveletet vezérlő információ 3 kötött formátumú szót használ:

- **csatorna címszó:** az operatív tár rekeszében található, az elindított csatorna mindig ennek a rekesznek a tartalmát tekinti csatornacímszónak,
- **csatorna parancsszavak:** duplaszó hosszúságúak, az operatív tárban bárhol lehetnek az I/O utasítást használó program területén. A csatornaprogramot alkotják egy gépi kódú programhoz hasonlóan, amely speciálisan az I/O műveleteket irányítja.

147

-csatorna állapot szó: az operatív tár megfelelő rekeszében található, hossza 8 bit. Azokat a körülményeket tárolja itt, melyek az I/O műveletek lefolyására jellemzőek, és a feldolgozó programnak tudomására akar hozni,

A csatorna címszó felépítése:

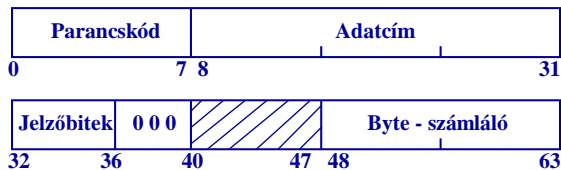


- a 0 - 3. biteken található a csatorna program tárvédelmi kulcsának numerikus része: a vezérlő egység akkor használja amikor a csatorna az operatív tárhoz fordul és a tárhivatkozás jogosultságát ellenőrzi.

148

- a 8 - 31. Biteken az a tárcím található, amelyen a csatornaprogram képződik. Ez az általunk megírt első csatorna parancsszó címe.

A csatorna parancsszó felépítése:



149

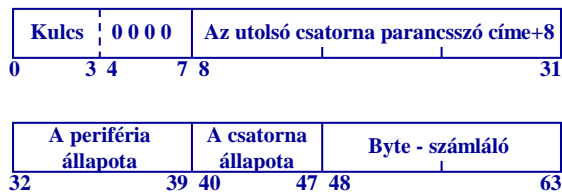
- a 0 - 7 bitek tartalmazzák a parancskódot. Hasonló a funkciója, mint a gépi utasítások műveleti kódjának, megadja milyen periféria műveletet kell végrehajtani,

- adatcím: annak az operatív tárbeli területnek a kezdőcíme, amelyet az adatátvitelben szerepeltetni akarunk,
- az öt jelzőbit jelentése: 32- adatláncolás; 33- parancsláncolás; 34- hosszhibaelnyomás; 35- átvitelelnyomás; 36- programvezérelt megszakítás

E jelzőbitek segítségével szervezhető és befolyásolható a csatorna program.

150

- a 48 -63. Bitek azt a bináris számot tárolják amely megadja, hány bájtot kell átvinni a perifériáról;-ra
A csatorna állapot szó:



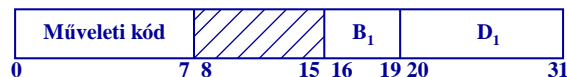
151

- a 0 - 3. bitek a csatornaprogram tárvédelmi kulcsának numerikus részét tárolják,
- a 8 - 31 bitek olyan tárcím amely 8- cal nagyobb mint az utoljára végrehajtott csatorna parancs szó, normál esetben a csatorna program végét jelzi,
- a következő 2 bájtban tárolja a csatorna az I/O rendszer állapotát.

152

I/O utasítások: A processzor utasításkészletében a beviteli utasítások 32 bit hosszúságúak, kétcímesek, A cím két összetevőből áll: a csatorna és a periféria címéből, mindkettő nyolcbites
 Az I/O utasítás felépítése: az I/O művelet céljára négy gépi utasítás szolgál:
 SIO - Start I/O; a művelet indítására,
 TIO - Test I/O; állapot lekérdezésre,
 HIO - Halt I/O; a művelet leállítására,
 TCH - Test Channel, a csatornaállapot lekérdezésére,

153



A gépi utasításokhoz hasonlóan, az I/O utasítások is megadják, hogy milyen jellegű műveletet végezzen és melyik perifériával,

- A műveleti kód határozza meg az utasítás típusát,
- A B₁ jelű mező határozza meg az utasítás által használt bázisregiszter értékét,
- A D₁ jelű mező a címeltolás értéke,
- A B₁ és D₁ mezők együttesen határozzák meg a periféria címét,

154

A (B₁) + D₁ számot a gép nem operandusként hanem I/O rendszer címeként értelmezi.
 A cím a következő alakú lesz:



Az I/O utasítás végrehajtása véget ért a megfelelő periféria kiválasztásával, a kiválasztás kimenetelét (szabad, foglalt,...stb. periféria), a feltételkód tartalmazza.

155

- Az I/O művelet lebonyolítása ezután a csatornarendszer és a csatolt perifériák feladata
- a csatorna megkeresi a csatornaprogramot, azaz kiolvassa az első csatorna címszót,
 - majd kiolvassa az első csatorna állapotszót és hozzáfog annak végrehajtásához,
 - ezt követi a második...stb. csatorna parancsszó kiolvasása és végrehajtása, míg a csatorna program véget nem ér,
 - végül a csatorna kitölti a csatorna állapotszót és I/O megszakítást generál, hogy az I/O művelet lefolyásának körülményeit, a feldolgozó program tudomására hozza.

156

Az I/O utasítások privilegizáltak, ezeket a gép csak Supervisor állapotban hajtja végre.
A feldolgozó program program az I/O művelet indítása előtt át kerül Supervisor állapotba.
Ha a számítógép operációs rendszer vezérlete alatt dolgozik, akkor az I/O műveleteket a Supervisor indítja, egyébként tetszőleges, általunk megírt programból is indíthatók.

157

8. A megszakítások, a megszakítási rendszer
Megszakítási rendszer

A processzor valamilyen okból az éppen futó processz következő utasítása helyett egy teljesen más címen levő másik program utasításait hajtja végre
Az operációs rendszer az előbb futó processzt valamilyen okból felfüggeszti egy megszakítással, majd később visszaadja a vezérlést, ha még az lehetséges
Megszakítás: hardver vagy szoftver; külső vagy belső
Adott esetben részleges vagy teljes tiltás is lehetséges

158

A megszakítás szükséges mert:
Bizonyos tevékenységeket csak az operációs rendszer végezhet el, alkalmazói program nem
I/O művelet lebonyolítása
hardver eszköz állapotváltozása
hibák kiszűrése
rendszeradminisztráció lebonyolítása
ütemezése, szinkronizálások, rendszerhívás

159

Megszakítások okai:

- a) programhiba miatt
- érvénytelen utasítás
 - címzési hiba
 - nullával való osztás
 - túlsordulás, stb.
- b) géphiba miatt
- hardver eszközök hibája

160

- c) szoftverből generált megszakítás (SVC, INT) miatt
d) külső megszakítás miatt
- timer tick (időzítő)
 - interrupt billentyű
 - másik processzortól érkező kérelem
- e) I/O művelet miatt
- I/O műveleti igény, kezdés vagy vég
 - esetleg I/O hiba

161

A megszakítás menete:

- megszakítási kérelem érkezik a központi egységhez
- az operációs rendszer átveszi a vezérlést
- ha a megszakítás tiltott, akkor várakozni kell (ciklikusan vagy csak meghatározott ideig - time out)
- amikor elfogadja, akkor kezdi a megszakítás-vezérlő a processz állapotvektorát (regiszterek) lementeni (operációs rendszer)

162

- elmentés után a megfelelő megszakítási vektorból előveszi a CPU a megszakítási rutin induló állapotvektorát, és átadja a vezérlést az rutin első utasítására
- fut a megszakításvezérlő, felderíti a megszakításkérelem okát, és annak megfelelően elvégzi a dolgokat. Normál esetben eközben már újra engedi a megszakításkérelmeket, de lehet hogy az adott megszakítás vagy akár mindegyik tiltva (!) van

163

- a komplett lefutás után visszaadja a vezérlést az operációs rendszerek, aki dönt a további programfuttatásról (ütemezés)
- futtatás: állapotvektor visszatöltése, CPU-ra kapcsolás (az operációs rendszer visszaadja a vezérlést valamely programnak)

Fontos: egyáltalán nem biztos, hogy az operációs rendszer ütemezője annak a processznek adja vissza a vezérlést, aki az adott megszakítást kérte; ekkor a megszakítás eredménye rendszerterületen tárolódik

164

Fontos továbbá mg az is, hogy a rendszer minél rövidebb ideig legyen megszakítás fogadására alkalmatlan (védtelen) állapotban

CPU működési állapotok:

- normál üzemmód: az utasításkészlet csak egy rész érhető el
- privilegizált üzemmód: minden utasítás elérhető és végrehajtható

165

Operációs rendszer: mindig privilegizált módban működik - ezt nevezzük Supervisor vagy Master módnak

Felhasználói programok: mindig normál módban használja a CPU-t

I/O műveletek csak privilegizált állapotban hajthatók végre

166

Megszakításkezelést elősegítő eszközök:

- CPU működési állapotok
- több regiszterkészlet bevezetése
- megszakítási maszk bevezetése
- megszakítási osztályok - párhuzamosítás
- megszakítási precedenciák, prioritások, sorrendek
- gyors utasítások az állapotvektor mentésre/töltésre

167

9. A memória

A számítógépekben a legkisebb táregység a 8 bitből álló bájt, mely elegendő bármely szimbólum tárolására, a matematikai műveleteknél több bájtból álló szavakkal dolgozik a gép.

A tár felépítését tekintve alapvetően két részre oszlik:

- állandó tár ROM
- operatív tár RAM

A memóriák közvetlen hozzáférések, azaz bármely rekesz megtalálásához és kiolvasásához ugyanannyi időre van szükség.

Elérési idő: a memóriához fordulás pillanatától az adatnak a CPU- ba való megérkezéséig eltelt idő.

168

ROM (Read Only Memory)

- csak olvasható memória,
- benne található az operációs rendszer magja,(firmware)
- tartalmát programozási eszközökkel a felhasználó nem módosíthatja,
 - a felhasználó géppel együtt, programmal feltöltve kapja, a gép része
- gép alapműködéséhez szükséges programok számára alkalmazott memória,
- gyárilag programozott „nem felejtő”, tápfeszültség nélkül is megőrzi a beírt adatokat pl. a PC-ék ROM lapkájában tárolt ROM BIOS amely tartalmazza a SETUP programot,

169

A ROM négy alaptípusát különböztetjük meg:

- szabványos ROM : a gyártó alakítja ki,
- PROM : a felhasználó által speciális berendezéssel (beégető készülék) egyszer programozható ROM, a felhasználó saját programot és adatot helyezhet el benne,
- EPROM: felhasználó által programozható, olyan mint a PROM, de tartalma többször is módosítható, először ultraibolya (UV) fényvel törlik a régi tartalmat, majd EPROM égetővel viszik be az újat,
- EEPROM: elektromosan törölhető, a beírás és a törlés eltérő elektromos impulzusokkal történik,

170

RAM (Random Access Memory: közvetlen hozzáférési memória - RWM),

Jellemzője: rekesz hozzáférési sebesség(elérési idő) kapacitás (a tárolt bájtok száma)

- írható olvasható memória,
- általános feladatok megoldásához szükséges programokat és adatokat tárolja
- feszültség kimaradás esetén elveszítik tartalmukat,ezért a gép kikapcsolása előtt a tartalmuk mentéséről gondoskodni kell, kivétel a CMOS RAM

171

- CMOS RAM: a merev és hajlékonylemez jellemzőit tárolja, valamint a PC belső óráját működteti, kis energia fogyasztású, ezért erre a célra szolgáló elemmel táplálható, kikapcsolás után is megőrzi információtartalmát, tartalma bármikor átírható, egy a ROM- ból futtatható segédprogrammal, a BIOS Setup segítségével.

A RAM kivitele lehet: - Dinamikus RAM, (DRAM)
- tartalmát rövid időközönként fel kell frissíteni
- Statikus RAM, (SRAM)
- frissítést nem igényel

172

Dinamikus RAM, (DRAM)

- könnyű gyártani, viszonylag olcsó
- ma már a DRAM fejlettebb változatait használják leginkább, melyek a DRAM- ra épülnek, annak tovább- bővítései,
- egy tipikus DRAM áramkör egy bitet olyan memóriacellában tárol, mely két elemből, egy speciális tranzisztorból FET-ből (Field Effect Transistor- Térvezérlésű tranzisztor), és egy kondenzátorból áll, a bemenetére adott feszültségtől függően, kinyit vagy lezár, mint egy miniatűr kapcsoló,

173

- minél kisebb a kondenzátor annál hamarabb kisül elveszíti a benne tárolt töltéseket, ennek kivédésére frissíteni kell.
- a memória cellákat mátrix alakban szervezik
- a memória vezérlőnek kell szolgáltatnia a sor és oszlopcímeket, hogy egy memória cellába írjunk vagy onnan olvassunk,
- a PC-ék DRAM- jai egy vagy kétoldalasak lehetnek
 - az egyoldalasak a SIMM- ek a memóriamodul nyomtatott áramköri kártyájának csak egyik oldalán
 - a DIMM- ek mindkét oldalán tartalmazznak RAM lapkákat,

174

FPRAM (Fast Page RAM, gyors lapozású memória)

- mivel a DRAM- nál a sor- és oszlopcímeket csak párosával adhatjuk ki, ezért a címzés hosszadalmas, és még folyamatosan frissíteni is kell,
- a mátrixszervezés egyszerű egyszerű sebességnövelő módszer, itt a lapszervezés hatékony kihasználásával a RAM sebessége felgyorsítható,
- egy lapon lévő cellák sorcíme azonos, csak az oszlopcímek változnak, ha azonos lapon lévő cellákat szeretnénk elérni, célszerű a sorcímet megtartani és csak az oszlopcímeket változtatni,

175

- az oszlopcímek kiadása között egy kis időnek el kell telnie, amely alatt a frissítés elvégezhető,
- ezzel a módszerrel mintegy 40%- os sebességnövekedés érhető el,

EDORAM (Extended Data Out RAM, kiterjesztett adatkimenetű RAM)

- az FPRAM- hoz képest mintegy 20%- kal a hagyományos DRAM- hoz képest mintegy 60%- kal gyorsítja a memóriaműveleteket,
- az elődjeikhez képest csökkentett kiolvasási idejük,

176

- a DRAM- nak két blokk kiolvasása között egy várakozási állapotot kell beiktatni a memória frissítése céljából, ezzel szemben az EDORAM esetében ez a várakozási idő kimarad, tehát a két blokk közvetlenül egymás után kiolvasható,
- ezt úgy érik el, hogy az EDORAM- ban reteszeket vagy másodlagos memóriákat adnak a meglévő memóriacellákhoz, amelyek addig tartják stabilan a DRAM- ból kiolvasott információt, amíg azok megbízhatóan elérhetik a processzort,
- az áramköri kiépítése miatt fogyasztása is kevesebb,

177

SDRAM (Synchronous DRAM, szinkron DRAM)

Az alapgondolat, hogy a processzor órajelével működjön a memória, így kiküszöbölhetőek a szinkronizálás hiányából adódó várakozási ciklusok.

- az SDRAM felfogható úgy is mint két EDORAM párhuzamos kapcsolása,
- a memóriát két területre (bankra) osztják, a memóriavezérlő miközben az egyik bankból olvas vagy ír, addig a másiknak már átadja a rá vonatkozó memóriacímet, mikor az első bank adatai feldolgozásra kerültek, azonnal rendelkezésre áll a kiolvasásra szánt adat,

178

vagy készen áll az adatok beírására. Mindkét memóriabank lapszervezésű.

Statikus RAM (SRAM)

Az SRAM- bankét állapotú kapcsolóelemek tárolják az információt, áramköri kialakításuk szerint flip- flopok, melyeknek két stabil állapotuk van

- azáltal gyorsabbak a korábbi RAM- oknál, hogy összehangolják a processzorsín és a memória sebességét,
- nagyon gyorsak és drágább az előállításuk, kevéssé integrálható, kevesebb energiát fogyasztanak
- elsősorban gyorsító (cache) memóriaként használják, nem igényel frissítést,

179

- Fajtái:
- Aszinkron SRAM- a processzorral nem szinkronban működik, a processzornak az olvasás és írás esetén néha várakoznia kell (a modern PC-ékben ma már nem használják)
 - Szinkron SRAM ez a típus a processzorral szinkron üzemben működik együtt
 - a processzor és a memória működési sebessége azonos, mindkettő ugyanarról a rendszer órajelről működik,
- Gyorsító tár (cache): a processzor és a memória közötti sebességkülönbség áthidalása,

180

- Fajtái:
- belső (internal cache) gyorsítár: a processzorba épített,
 - külső (externál cache) gyorsítár: az alaplapon található,
 - lemezgyorsító tár: a lemezes tárok feldolgozási sebességének a növeléséhez, mely csökkenti az adatok átlagos elérési idejét,
 - memória gyorsító tár (RAM- cache)

Flash-RAM:

- a félvezető memóriának újabb fajtája,
- úgy programozhatók mint a RAM, de elektromosan is törölhetők, leginkább az EEPROM- okhoz hasonlíthatók,

181

- ez a szuperlapos memória már 16, 32 Mbit-es chippekkel kaphatók,
- nagyobb sűrűséggel integrálhatók mint a DRAM chippek, nem szükséges hozzájuk akkumulátor, mert nem törlődnek,
- képesek az eddigi RAM és ROM chippek feladatát átvenni,
- a kis energiatakarékos memóriákon kívül, helyettesítheti az EPROM- ot
- memóriakártya formájában használják mely megfelel a PCMCIA szabványnak,
- a Flash- kártyák nem tartalmaznak mozgó részeket, kevés áramra van szükségük, laposak,

182

Címzési módszerek:

- közvetlen (direkt vagy egyenes) címzés: közvetlen az operatív tár bájttjára mutat
- indirekt címzés: index regiszteres (indexregiszter tartalmazza a megfelelő címet), bázisregiszteres címzés (bázisregiszter tartalma egy alapcím, ehhez adódik hozzá az ún. eltolás = így kapjuk meg a a végleges címet)
- virtuális címzés: amikor a tényleges cím nem az operatív tárban van, hanem egy közvetlen elérési gyors háttértárolón (pl. mágneslemez).

183

Virtuális címzéshez szükség, hogy a processzor nagyobb területet tudjon címezni, mint a tényleges operatív tár mérete

Társzervezés: akkor van rá szükség, ha processzor nem tudja címezni a teljes operatív tárat. Ilyenkor szegmensekre osztják a fizikai memóriát, s a processzor egyszerre egy szegmenst tud belátni

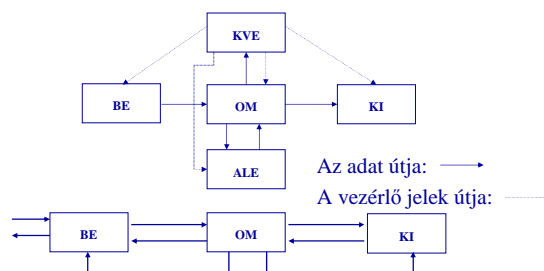
184

Adatforgalom a CPU és a memória között.

A processzorok sebességének növekedésével a CPU és a perifériák sebességkülönbsége egyre nagyobbá vált. Hogy a perifériák lassú működése ne akadályozza a KVE nagy elvi működésének kihasználását, a CPU-t mentesíteni kellett a korábban takarékosági okokból rábízott I/O operáció közvetlen vezérlésétől.

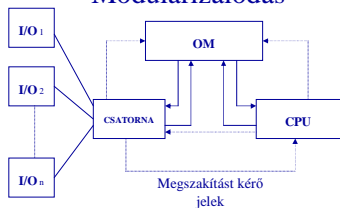
185

Memória központú architektúra



186

Modularizálódás



Az OM több irányból is hozzáférhetővé vált. Ezt használják ki az I/O operációk közvetlen vezérlését a CPU-tól átvállaló új funkcionális blokkok az ún. **autonóm csatorna-egységek (busz)**.

Az OM és a perifériák közé bekeleődve végzik az adatátvitel bufferelését.

187

A DMA jelentése, használata

Az alaplapon a mikroprocesszor mellett DMA vezérlőt is találunk. A PC és XT típusokba egy 4 csatornás, az AT típusokba két 4 csatornás egységet építettek be.

A DMA vezérlő típus egy buszciklus felhasználásával viszi át az adatot a memória és a periféria között. A címvonalakra a memória címet helyezi, a perifériát a csatornához tartozó elfogadás jellel választja ki.

188

- A DMA vezérlők kettős szerepet játszanak, de időben szétválnak a két szerep. Az adatátvitel fogadása alatt SLAVE az adatátvitel végrehajtása alatt MASTER.
- Egy buszon lehet több MASTER vagy SLAVE egység, de egyszerre csak egy aktív lehet. A buszt felhasználni kívánó MASTER -eknek el kell dönteniük, hogy melyik legyen az aktív. Ezt a folyamatot nevezik **arbitációnak**. Az adatátvitelben szerepet játszó SLAVE egységeket a MASTER jelöli ki.

189

10. Háttértárak

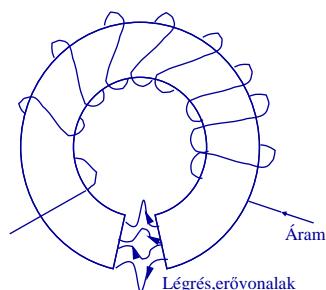
Mágneses jelrögzítés

A keménymágneses anyag jellemző tulajdonsága: hogy hosszú ideig megőrzik mágneses tulajdonságukat.

Cél : kis területre koncentrált olyan erősségű mágneses tér gyors létrehozása és megszüntetése amely képes a egy keménymágneses tulajdonságú anyagot mágnessé változtatni és viszont,

Elektromágneses teret használunk erre a célra, a mágneses tér fölerősítéséhez gyűrű alakú vasmagos tekercset.

190



A mágneses erővonalak csak a légrésben lépnek ki a vasmagból, így kis területre koncentráltuk, A lemez felülete speciális anyag melynek elemi részecskéi a mágneses mezőtől függő irányban állnak

191

A kigyűrődő mágneses erővonalak érintkeznek a lemezzel. Ha a vezetékbe ellenkező irányú áramot vezetünk a mágneses tér iránya is megváltozik. Az erővonalak iránya a lemezbevonat részecskéit a megfelelő irányba állítja. Mivel a számítógép digitális működésű és bináris kódolást használ, ezért a jeleknek csak két állapota, 0 és 1 lehet, az ennek megfelelő részecskéirány:



192

Mágnesszalag:

- a fájloknak a soros egymás utáni tárolását végzi,
- a fájlokat alkotó logikai rekordok blokkokba, adattömbökbe kerülnek
- a blokkoknak saját címük nincs,
- az ilyen soros elérésű tárnál valamely adat megkeresése csak úgy lehetséges, ha az összes többi előtte levő adaton már átléptünk,
- a rögzítendő karaktereket a szalag mozgására merőlegesen, keresztirányú oszlopokban, ún. Frame- okban helyezik el,
- a szalag kezdeténél van az úgynevezett,

193

Szalagkezdő jel(BOT), a szalagvégén a szalagvég jel (EOT),

- a szalagon vannak olyan blokkok, mely feladata az adattekeresén található adatok, fájlok azonosítása, ellenőrzése, ez az ún. belső címke vagy azonosítók,
- ha a fájl tartalmaz címkét az lehet:
 - rendszercímkek: a rendszer(software) állít elő és dolgoz fel,
 - felhasználói címkek: a fájl azonosítására szolgál,
- szalagon alkalmaznak még szalagjelzőt, egy szalagjelző a fájl végét jelenti

194

A mágnesszalag egység működése a magnetofonhoz hasonlítható.

A szalag az írási olvasási folyamat alatt egy leadó orsóról, egy üres felvevő orsóra kerül.

A szalag mozgatását a továbbító görgők végzik a szalagszállítás lehet : nyomógörgős; vákuumgörgős;

Az orsó és a fej között bizonyos szalagrészt átmenetileg tárolnak, pufferolnak.

195

Mágneslemez:

A hajlékonylemezes táruk az adathordozó cserélhetősége, és hordozhatósága miatt előnyös,

A hajlékony lemezt egy speciális műanyag tok védi,

A mágneses réteg hordozóanyaga hajlékony rugalmas műanyag mely a floppy meghajtóba (FDD, Flexible Disk Drive), a centrifugális erő hatására merevvé válik, amely 300 vagy 3600ford/perc állandó fordulatszámmal forgatja,

Mérete: 5,25 és 3,5 inch

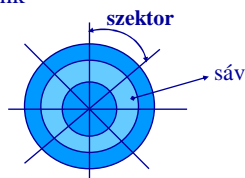
Tulajdonságai: - a rajta tárolható adatmennyiség,

- adat- hozzáférési idő

- adatsűrűség

196

- kétoldalas dupla sűrűségű DS DD,
- kétoldalas nagysűrűségű DS HD, lemezeket használunk



- több szektorból álló logikailag összetartozó szektorcsoport: **cluster**
- a lemezen lévő koncentrikus körök a **sávok**, melyek a fejek sugár irányú elmozdulásával érhetők el,

197

- minden sávot megadott számú egyenlő hosszúságú részekre osztunk ez a **szektor**,
- a **FAT tábla** az állományok rekordjainak lemezen történő elhelyezkedését tárolja,
- a **formázás** során a hajlékony lemezt sávokra és szektorokra osztjuk fel, ugyanis a sávokon egy meghatározott mintázatot kell létrehozni az adatfelírás előtt,
- a szektorokban az adaton kívül feljegyzésre kerül a sáv és szektor száma, a szektor üres vagy foglalt állapota, valamint annak a sávnak és szektornak a száma ahol az adatok folytatódnak.

198

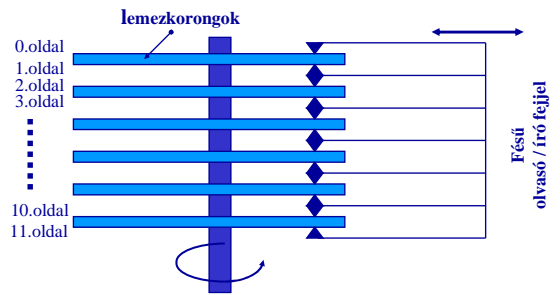
Merevlemez táruk

Az adathordozó merev speciális mágnesezhető felülettel bevont lemezkorong,

Többféle merevlemez tár létezik, de a szem. szg.-nél, a winchester HDD(Hard Disk Drive) terjedt el.

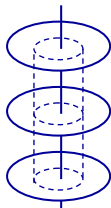
- az egymás fölött elhelyezkedő sávokat nevezzük **cilindernek**,
- a fizikai meghajtót önálló logikai részekre lehet osztani, ez a **partíció**

199



Lemezköteg mágneses bevonatú alumínium lemezekkel

200



Egy cylinder a lemezsomagon

A lemezek a meghajtóegység belsejében vannak, hermetikusan lezárt tokban, nem cserélhetők

A lemezek mindkét oldalára írnak, a lemezkorongok egyirányba és egyszerre forognak, közös tengelyen

201

- a lemezek közötti résekbe nyúlnak be az író- olvasó karok, lemezoldalanként egy- egy és együtt mozognak,
- a hajlékonylemezknél jóval nagyobb kapacitás és gyorsabb az adatok átvitele,
- a fej olvasásnál és írásnál nem fekszik a lemezre, fizikailag nem érintkezik vele, hanem néhány ezredmilliméteres légpárnán lebeg felette, tehát a zárt meghajtóban lévő lemeznek nagyon tisztának kell lennie,
- az adatok elérési ideje mintegy tízezerszer lassabb, mint a memóriáké,

202

- a fenti tulajdonságok lehetővé teszik, hogy sokkal nagyobb jelsűrűség érhető el,
- az állandóan 3600 ford/perc- el forgó lemezkötegekbe benyúló író- olvasó fejek lényegesen finomabban pozícionálhatók
- a hajlékonylemezhez hasonlóan a merevlemez is formattálni kell, mielőtt írásra használnánk, de még különböző tartományokat (partíciókat) is készíthetünk a különböző oprendszerek számára,
- először partícionáljuk a lemezt, majd az egyes partíciókat, formattálni kell,

203

CD-ROM

Optikai jelrögzítés:

A lézérfénynek egyetlen színe van, ami koherens így lehet fókuszálni és erősíteni is tehát nagyon jól irányítható.

A CD-ROM- ok olyan fényt használnak, ami a színspektrum kisebb frekvenciájához tartozik pl. a piros és a sárga, kísérlet folyik a kék és zöld lézerekkel melynél ötször akkora adatsűrűség érhető el mint a piros lézer esetén.

A szabványos CD lemezek átmérője 4,75" (120mm) átmérőjű és 1,2mm vastag

204

Anyaga: átlátszó polikarbonát műanyag hordozóra alumínium (vagy arany) réteget, majd egy lakkréteget visznek fel,

A hordozó lemezt öntési eljárással készítik, melynek során az egyik oldalán apró bemélyedéseket hoznak létre. Erre az oldalra viszik fel a jól tükröző alumínium réteget majd a lakkréteget ami véd az oxidációtól és a mechanikus sérülésektől.

A tükröző réteg felületét **land**- nek a bemélyedéseket pedig **pit**- nek nevezzük A pitek spirál alakban sorakoznak a lemezen, belülről kifelé.

205

Az információkat egy spirál alakú, belülről kifelé vezető barázdákkal ellátott sáv tárolja, amelyeket a gyártásnál sajtolnak bele.

Az adatok melyek csigavonalban vannak felírva, völgyek és dombok egyenletes sűrűségű sorozatából áll,

Olvasáskor az olvasó a vele szemben forgó lemezsávra fókuszált lézersugarat irányít, melyet a sáv megvilágított völgyei szétszórják, dombjai viszont, melyek a fémfelület síkjában vannak visszatükrözik.

A visszavert fény intenzitása lesz más a kiemelkedéseknél és a mélyedéseknél.

206

A visszatükrözött lézernyaláb a lézeres olvasó fényérzékelőjébe jutnak.

Egy áramkör aszerint, hogy az érzékelő a lemez egyes helyeiről visszatükröződő sugarat észlelt-e vagy nem, nullákat vagy egyeseket állít elő.

- a mágneslemezek állandó szögsebességgel forognak, ezért a lemez fordulatszáma állandó, ezáltal a belső sávok sebessége kisebb, mint a külsőké, ezáltal a belső sávokban az adatsűrűség is nagyobb mint a külső sávokban.

- a CD- knél ilyen gond nincs mert minden szektor azonos hosszúságú, a lemezt állandó kerületi sebességgel forgatják, így a fogási sebesség, attól

207

függően változik, hogy éppen a lemez mely részét olvassa, így a lemez és a lézerfej egymáshoz viszonyított helyzete állandó,

- mivel az adatsűrűség állandó, a tárolható információ mennyisége sokkal nagyobb mint a mágneslemezeknél

- a nagy sűrűségű lemezek kisebb völgyeket (bemélyedéseket) tartalmaznak mint a szabványosak, így több adat fér el rajtuk, de a sűrűbben felírt adatok kiolvasásához keskenyebb lézernyalábot és fejlet-
tebb olvasófejet használnak

208

- a többréteges lemezek az információt egymásra rétegelten tartalmazzák, úgy hogy a felső információs réteget felül normál fényvisszaverő felülettel, alul pedig félig áteresztő felülettel zárják le, és ez alá teszik a következő információs réteget. Az olvasandó réteg kiválasztása a lézernyaláb fókuszának ráállításával történik. A nagysűrűségű lemezeket **DVD** lemezeknek nevezik.

A CD- ROM- nál az elnevezés utal arra (ROM), hogy csak olvasható tárról van szó.

CD- R: az egyszer írható CD. A tömegben gyártott CD- ken a pit- eket a gyártás során hozzák létre,

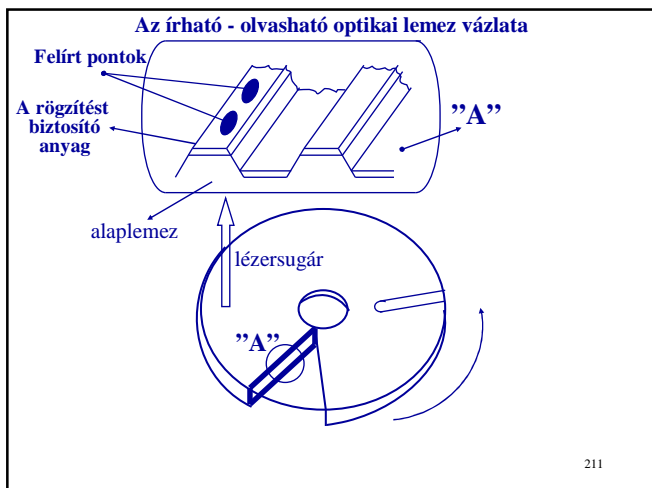
209

A CD-R lemezek üresek.

Ezen lemezek szénszál as hordozóanyaga hasonló a CD- ékhez, de a gyártás során felvisznek egy spirális sávot, ami íráskor a lézersugár pozicionálását segíti, az alumínium réteg helyett egy szerves festékreteget használnak, majd erre vékony aranyréteg kerül. Ezután védőlakkréteget visznek fel.

CD írása: íráskor a CD író egy a energiájú lézersugarat a festékretegre fókuszálja és az írás során beégeti azt, lyukakat éget bele, így a pit- hez hasonló elváltozás lép fel, melyet a szabványos CD meghajtó képes olvasni.

210



A normál CD- ken 650 MB a tárolókapacitás , ami 74 percnyi hanganyaghoz, 25-30 percnyi digitalizált videofilmhez elég,
 A helyigényes multimédia- alkalmazásokhoz igen nagy tárolóhelyekre van szükség. Az ennek megfelelő új tárolómédium a **DVD** .
 Az előállításához szükséges technika alig változott, mivel a külső méretek nem változtak, ezért a gyártáshoz szükséges gépeket könnyen át lehet alakítani, így az új lejátszó eszközök lefelé kompatibilisek és velük az eddigi CD- ket is lehet olvasni.

Az első információtároló rétegen 4,7 GB fér el, a másodikon 3,7 GB azaz maximum 8,4 GB tárolókapacitással.
 Ez 2 x 135perc film tárolására elegendő MPEG-2 kódolást alkalmazva.
 Újszerű lehetőség, hogy a műsorban el lehet helyezni rögzített mutatókat, és nemcsak egyszerre hanem a mutatókkal tagolt fejezetekben is le lehet játszani a filmet, multimédia dokumentumot.
 CD-WORM egyszer írható többször olvasható optikai lemez.
 CD-RW újraírható lemez.

10. Egyéb perifériák
Billentyűzet
 A billentyűzetek a számítógép kiemelt fontosságú beviteli egysége.
 A billentyűzet bonyolult integrált áramköröket, így saját processzort is tartalmaznak.
 - ezek az áramkörök észlelik a billentyű leütését,
 - átmenetileg tárolják, majd közlik a számítógéppel az egyes lenyomott billentyűket egyértelműen azonosító bináris számokat, az ún. *Scan kódokat*, amely bekerül a gép memóriájába,

A billentyűzet gombjai ún. Mátrixba vannak kapcsolva, ezzel elérhető, hogy kevesebb vezetékkel kapcsolódjon a számítógéphez.
 Amikor egy gombot lenyomunk, a gombhoz tartozó sor és oszlop kijelölésre kerül. Minden sor oszlop kombinációhoz tartozik egy kód.
 Mechanikai kialakításuk:
 - kapacitív érintkezős: ahol a billentyű lenyomása töltésváltozást eredményez,
 - nyomóérintkezős: ahol a billentyű lenyomása nyomóérintkezőket zár,

A billentyűk lenyomásakor az érintkezőkhöz kapcsolt áramkör érzékeli a változást, majd a változás helyének és jellegének megfelelően a billentyűzet saját processzora egy *scan kódnak* nevezett bináris számot generál amely bekerül a saját memóriájába a billentyűzet pufferbe, és közli a számítógéppel hogy a billentyűzet állapota megváltozott,
 A változás bejelentése *megszakításkéréssel* történik. A megszakításkérés hatására a számítógép processzora félbeszakítja az éppen folyó műveletet, és rátér egy olyan program végrehajtására, amely kideríti, hogy mi történt a billentyűzeten, majd beolvassa a billentyűzet által generált scan kódot.

Fajtái: 84, 101, 102, 105 gombos

Felépítése: Alfanumerikus (karakteres) billentyűzet,
Numerikus billentyűzet,
Funkcióbillentyűk: F1 - F12 ,
Kurzormozgató billentyűk: ←↑→↓ ,
Segédbillentyűk(jelentésmódosítók):
SHIFT; CTRL; ALT; Num Lock;...stb.,
Vezérlő billentyűzet: ENTER, BREAK,

A 101- től felfelé a billentyűzet parancsokat is képesek elfogadni melyekkel a kijelzők állapotát, a gépelés sebességét, az üzemmódot lehet állítani, A kétirányú forgalom az esetleges hibákat jelzi, és az átvitelt ismételni képes,

A billentyűzet lenyomási és felengedési kódokat használ

217

Monitor(display)

A monitorok a számítógép alapvető kiviteli eszközei.

A képernyőn jelennek meg a számítógép felhasználónak szóló üzenetei.

A képernyők is pontokból (pixelekből) állítják össze a karaktereket és rajzokat. A nyomtatókhoz hasonlóan a monitor minősége a megjelenített képpontok sűrűségétől és méretétől függ.

Minél **több a képpont**, annál **jobb a képernyő felbontóképessége**, annál **élesebb a kép**.

A monitorok általában egy kiegészítő kártyával vannak összekötve az alaplap buszával, ezt szokás grafikus kártyának, monitor vezérlőnek, nevezni.

218

Monitorvezérlő kártya teszi lehetővé, hogy a monitoron megjelenhessenek azok az információk amit a számítógép közöl a felhasználóval.

A kártyákon található egy grafikus memória, hogy a képalkotás zökkenőmentes legyen. A memóriát a PC felől lehet írni és olvasni egyaránt , de az újabb memóriákat egyidőben lehet írni és olvasni. A kártyán lévő céláramkör a monitor vezérlését látja el.

219