Számítógépek architektúrája

1. előadás

Istenes Zoltán

Eötvös Loránd Tudományegyetem
Informatikai Kar
Programozáselmélet és Szoftvertechnológiai Tanszék

Programtervező Informatikus BSc - B szakirány
2007 / Budapest

Az előadó (Istenes Zoltán) elérhetősége

- Tanszék: ELTE - Informatikai Kar
  Programozáselmélet és Szoftvertechnológiai Tanszék
- Iroda: ELTE Déli tömb 2.604
- E-mail: `istenes@inf.elte.hu`
- Honlap: `http://quasar.inf.elte.hu`
- Telefon: 2090555 / 8484
- Egyéb: tárgyfórum, ETR-kurzusfórum, postaláda, titkárság (2.614)...

Tudnivalók a tárgyról

- jegyzetek, segédanyagok:
  `http://quasar.inf.elte.hu/oktatas/szamarch`
- irodalom:
  Számítógép-architektúrák,
  Andrew S.Tanenbaum,
  Panem kiadó, ISBN: 9635452829
- kollokvium, követelményrendszer, kiselőadások

A tárgy "alapgondolatai"

- "Nem PC-ből áll a világ..."
- Általános alapelvek, fogalmak, lehetőségek, nagyságrendek, összehasonlítások, szintek, fizikai megvalósítások, ...
- mit, miért, hogyan?

A félév témakörei

bevezetés, történet, fogalmak, adatábrázolás, globális bemutatás, Neumann architektúra

Központi feldolgozó egység (CPU)

vezérlő egység, aritmetikai logikai egység, utasítás: készlet, típusok, felépítése, végrehajtás

Memória

hierarchia, cache, virtuális tárkezelés

Be- és Kimeneti rendszer (IO)

megszakítás-rendszer, DMA, csatorna, perifériák

többprocesszoros, párhuzamos gépek, hálózatok, egyebek...

Magasszintű programozási nyelv - gépi nyelv (kód)

Magasszintű nyelv (C)

```
swap (int v[], int k)
{ int temp;
temp = v[k];
v[k] = v[k+1];
v[k+1] = temp;
}
```

Assembler nyelvű program

```
swap:
muli $2, $5,4
add $2, $4,$2
lw $15, 0($2)
lw $16, 4($2)
sw $16, 0($2)
sw $15, 0($2)
jr $31
```

Bináris, gépi kódú program

```
00000010110110010001110100011010
11101110010100101000101110000010
11111001000000010100001010000001
11111001000000011000001100000001
00011001000000010100001100000001
00101011100000000000000001111111
```


Többszintű számítógépek

1. problémaorientált nyelvi szint
2. assembly nyelvi szint
3. operációs rendszer gépi szintje
4. utasításrendszer-architektúra szintje
5. mikroarchitektúra szintje
6. digitális logikai szint

Fizikai - logikai szintek

"Fizikai"

- "A számítógép"
- alaplap, processzor, memória, ...
- integrált áramkörök
- tranzisztorok, félvezetők (0-5Volt)
- elektronok, félvezető rétegek

"Logikai"

- processzor, memória, sínrendszer, IO
- logikai áramkörök (összeadó, számláló, ...)
- logikai kapuk (ÉS, VAGY, NOT, logikai 0-1)

Sok-sok NOP... Mennyi a sebesség különbség?

1 utasítás

```
NOP
```

2 utasítás

```
NOP
NOP
```

3 utasítás

```
NOP
NOP
NOP
```

4 utasítás

```
NOP
NOP
NOP
NOP
```


Tömbösszeadás... Lehet sebességkülönbség?

soronként

```
for i=1 to n
for j=1 to m
sum=sum+t(i,j)
```

oszloponként

```
for j=1 to n
for i=1 to m
sum=sum+t(i,j)
```


Módszeres programozás... Meghökkentő ciklusok

1/16-ok...

```
Program ciklus;
var s:real;
begin
s:=0;
while s<>1 do
s:=s+1/16;
End.
```

1/10-ek...

```
Program ciklus;
var s:real;
begin
s:=0;
while s<>1 do
s:=s+1/10;
End.
```


Számítógépek sebessége

- Milyen gyors a leggyorsabb számítógép?
- Hány szorzást végez másodpercenként?
- Mennyivel gyorsabb egy otthoni gépnél?
- Miért gyorsabb?
- Miért nem gyorsabb?
- Hogyan lehetne gyorsabb?

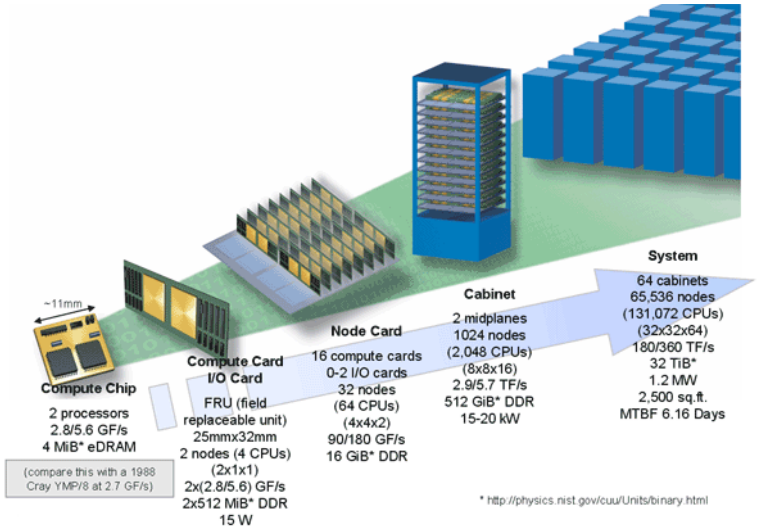
BlueGene/L - eServer Blue Gene Solution (IBM)

Processor type:	PowerPC 440
Processor:	700 MHz (2.8 GFlops)
Topology/Interconnect:	3D torus
Operating System:	CNK/Linux
Processors:	131072
Rpeak (GFlops):	280600
Rmax (GFlops):	367000

`http://top500.org`

Flops (Floating Point Operation per Seconds) =
Lebegőpontos művelet másodpercenként

BlueGene/L - eServer Blue Gene Solution (IBM)



Két gép összehasonlítása

	EDVAC 1	CRAY-1	
évszám	1952	1976	24 év
órajelciklus	2000ns	12,5ns	160x
mártixszorzás	100/s	130000000/s	1300000x

- az órajel növekedése: "technológiai"
- azonos órajel mellett is 8000x növekedés!
- a 8000x növekedés: "felépítésbeli" ("architekturális")

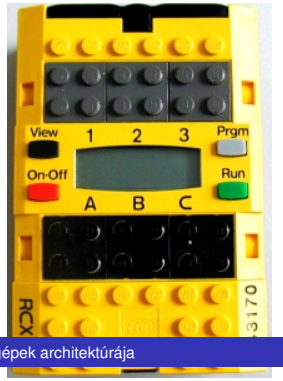
Számológép / számítógép

számológép

Főleg számtani műveletek végzésére alkalmas, gyakori, közvetlen emberi beavatkozást igénylő eszköz.

számítógép

Utasításlista (program) alapján adatokat manipuláló gép



Számítógép-generációk

- http://en.wikipedia.org/wiki/Timeline_of_computing
- http://www.computerhistory.org/
- http://oprendszer.elte.hu/architekturak/segedanyag.html

Összefoglalás, fogalmak

- alapgondolatok
- témakörök
- nyelvi szintek
- fizikai-logikai szintek
- sebesség
- számológép / számítógép
- számítógép-generációk

Számítógépek architektúrája

2. előadás

Istenes Zoltán

Eötvös Loránd Tudományegyetem
Informatikai Kar
Programozáselmélet és Szoftvertechnológiai Tanszék

Programtervező Informatikus BSc - B szakirány
2007 / Budapest

Informatika

Az informatika (kb. information technology) az információk:

- rögzítésével,
- kezelésével,
- rendszerezésével,
- továbbításával foglalkozik.

Az informatika tevékenységei

Az informatika a tevékenységét főként számítógépeken végzi:

- elméleti úton azáltal, hogy módszereket, modelleket, formalizmusokat dolgoz ki a számítógépek készítéséhez és működtetéséhez;
- mérnöki tevékenységgel úgy, hogy számítógépeket készít, illetve azokhoz elektronikai eszközöket alkot (hardver);
- rendszertervezéssel és készítéssel azáltal, hogy a számítógépek működtető eszközeit hozza létre, illetve azokat működteti (szoftver);
- alkalmazza a számítógépet azáltal, hogy különböző feladatok elvégzésére alkalmassá teszi, pl.: orvosi alkalmazások, kereskedelmi rendszerek, CAD, nyilvántartások, stb).

Az informatika részei, részterületei

- matematikai alapok
- algoritmusok és adatszerkezetek
- programozási nyelvek és fordítóprogramok
- adatbázisok
- konkurrens, párhuzamos és elosztott rendszerek
- számítás-elmélet
- számítógépek felépítése
- szoftvertechnológia
- mesterséges intelligencia
- lágy számítástechnika
- számítógépes grafika
- tudományos számítás

Az információ

- Információnak nevezünk mindent, amit a rendelkezésünkre álló adatokból nyerünk.
- Az információ olyan tény, amelynek megismerésekor olyan tudásra teszünk szert, ami addig nem volt a birtokunkban.
- Az információ minden olyan tény, állítás, hír, amelyeket a beszélő közöl. Ez egy csatornán keresztül eljut a hallgatóhoz. A hallgató számára a hír új jelentéssel bír és ezáltal tájékozottabbá válik.

Információ-tartalom (self-information)

Az információ-tartalom annak az információnak a mennyisége, amelynek a tudása egy bizonyos eseményről hozzáadódik valaki teljes tudásához.
Az információ-tartalom mértékegysége a bit.
Az információ-tartalom $I(A_n)$ ha A_n esemény bekövetkeztének a valószínűsége p
$I(A_n) = log_2\left(\frac{1}{p(A_n)}\right) = -log_2(p(A_n))$

Példa

Az információ tartalma, hogy egy kockával négyest dobunk:
$I(\text{'négyes'}) = log_2\left(\frac{1}{1/6}\right) = log_2(6) = 2.585 bit$

A bit fogalma

A bit az információ alapegysége.
A bit a kettes számrendszer egy bináris számjegye (**bi**nary digi**t**).
Például a 1001011_2 szám 7 bit hosszú.
A bit egy mérőegység, egy bináris számjegy információ tartalma.
Jele: b
A bájt (byte) bitek csoportja. 8 bit = 1 bájt
A szó (word) bájtok csoportja, a számítógép kialakításától függ.

SI (tizes számrendszer) prefixumok

Elnevezés	Jele	Értéke	Gyakori de hibás értelmezés
kilo	k	10^3	2^{10}
mega	M	10^6	2^{20}
giga	G	10^9	2^{30}
tera	T	10^{12}	2^{40}
peta	P	10^{15}	2^{50}
exa	E	10^{18}	2^{60}

$10^3 = 1000 \neq 2^{10} = 1024$

Példa

1 kB (kilobájt) = 1000 bájt (SI szabvány szerint), de nagyon gyakran 1024 bájtot értenek alatta

Kettes számrendszerben használatos prefixumok

Elnevezés	Jele	Értéke
kibi	Ki	2^{10} = 1 024
mebi	Mi	2^{20} = 1 048 576
gibi	Gi	2^{30} = 1 073 741 824
tebi	Ti	2^{40} = 1 099 511 627 776
pebi	Pi	2^{50} = 1 125 899 906 842 624
exbi	Ei	2^{60} = 1 152 921 504 606 846 976

IEC 60027-2 Nemzetközi Elektrotechnikai Bizottság (International Electrotechnical Commission; IEC) 1998.

Példa

1 kb (kilobit) = 1000 bit viszont 1 Kib ("kibibit") = 1024 bit
1 MB = 1000000 bájt viszont 1 MiB ("mebibájt") = 1048576 bájt

Az adat fogalma

Az adat egy objektum egy meghatározott változójának az értéke.
Egy konkrét adat tehát akkor tekinthető definiáltnak, ha meghatározzuk, hogy milyen objektum, melyik változója, milyen értéket vesz fel.

Példa

Az Eiffel torony magassága 300 méter

Az adat fogalma

Az adatoknak önmagukban nincs jelentésük.

Az adat fogalma jól elkülöníthető két másik rokonértelmű fogalomtól az ismerettől és az információtól:

- Az ismeret valamilyen objektummal kapcsolatos tapasztalataink, általánosításaink és fogalmaink összessége.
- Az információ olyan adat vagy ismeret, amely viselkedésünket befolyásolni képes.

A jel fogalma

Az információ-elméletben egy jel a kommunikációs csatorna állapotainak egy sorozata, amelyet üzenetté lehet dekódolni.
Egy kommunikációs rendszerben, egy küldő kódol egy üzenetet jelekké, amelyet a kommunikációs csatorna segítségével a vevőnek el lehet juttatni.

Példa

A küldő szavakat mond a telefon mikrofonjába.
A telefon a hangokat elektromos feszültségű jelekké alakítja.
A telefon az elektromos jeleket telefonhálózat segítségével a vevő telefonjába továbbítja, ahol az visszaalakul hanggá.
A vevő hallja a szavakat.

Analóg és digitális jelek

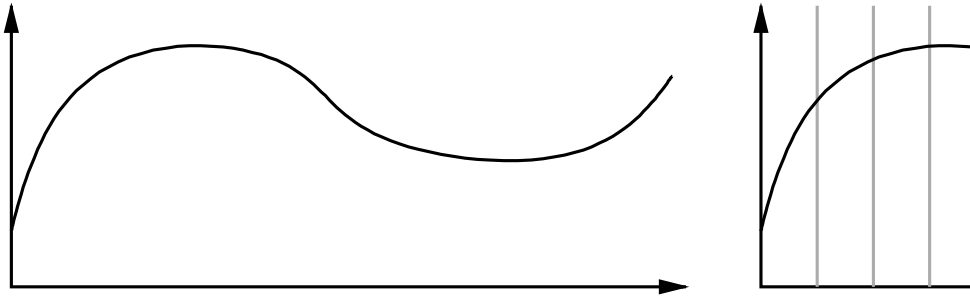
Két fő jeltípus:

- analóg jel: jelparamétere folytonos
  (valós életbeli folyamatok, pld.: hőmérséklet, nyomás, ...)
- digitális jel: diszkrét és kvantált jellemzőkkel bír
  (digitális számítógéppel feldolgozható)

Analóg-digitális átalakítás / Digitális-analóg átalakítás

Analóg jelből digitális jel

kvantálás, diszkretizálás, mintavételezés



Az algoritmus fogalma

Az algoritmus vagy eljárás olyan módszer, utasítás(sorozat), részletes útmutatás vagy recept, amely valamely felmerült probléma megoldására alkalmas.

- "végesség" (az algoritmus véges számú lépés után leáll)
- "meghatározottság" (mindegyik lépés pontosan, egyértelműen van meghatározva)
- bemenet (megadott halmaz elemeiből)
- kimenet (meghatározott kapcsolatban áll a bemenettel)
- hatékonyság (egy lépést, véges időn belül, papírral ceruzával is végre lehet hajtani)

Elemi, megengedett lépések

- az elemi lépések az adott szinttől függnek
- az elemi lépések egy nyelv utasításai
- egy nyelv meghatározott elemi (megengedett) utasításokkal rendelkezik
- egy magasabb szintű nyelv egy utasítását, egy alacsonyabb szintű nyelv több utasítására lehet lefordítani
- egy magasabb szintű nyelv egy utasítását, egy alacsonyabb szintű program értelmezheti

Számítógép program

A legtöbb számítógép program procedurális vagy deklaratív program.

procedurális program(ozás) (imperativ program)

utasítások listája, amelyek egyértelműen (explicit) megvalósítanak (implementálnak) egy algoritmust

deklaratív program(ozás)

jellemzők listája, amelyek meghatározzák a kimenetet, melyet egy algortimus határoz meg

Hardver - szoftver - firmware

hardver (hardware)

- a számítógép fizikai része
- például: digitális áramkörök
- ritkán változik

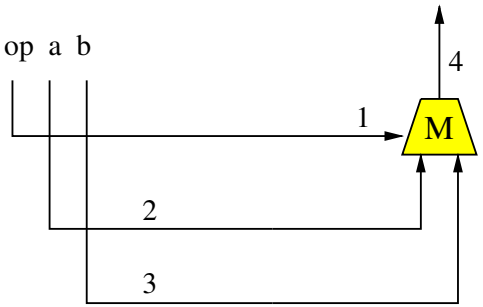
szoftver (software)

- program ami segítségével a számítógép megadott feladatokat hajt végre
- például: operációs rendszer
- gyakran változik

firmware

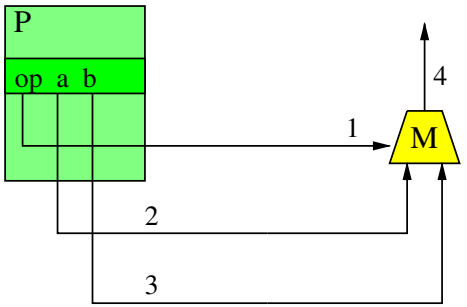
- hardvereszközbe épített szoftvertípus
- biztosítja a hardver működését és alapvető funkcióit

egyszerű gép - műveletvégző



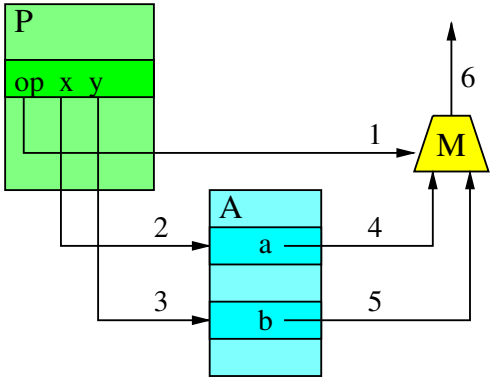
M - műveletvégző
1 - művelet (op)
2,3 - adat értéke (a,b)
4 - eredmény értéke

egyszerű gép - programmemória



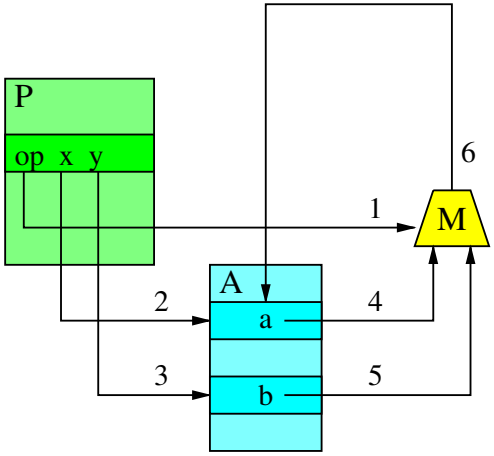
P - programmemória
M - műveletvégző
1 - művelet
2,3 - adat értéke
4 - eredmény értéke

egyszerű gép - adatmemória



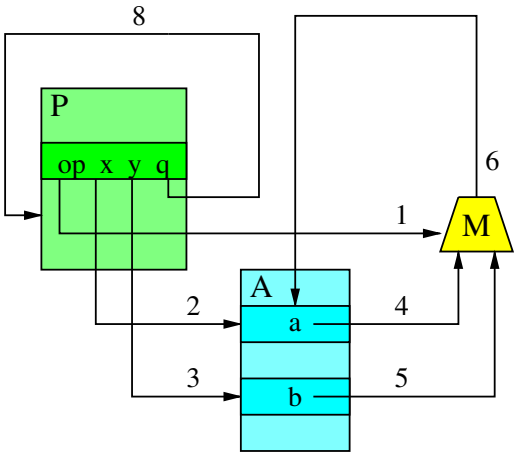
P - programmemória
A - adatmemória
M - műveletvégző
1 - művelet
2,3 - adat címe
4,5 - adat értéke
6 - eredmény értéke

egyszerű gép - eredmény visszaírása



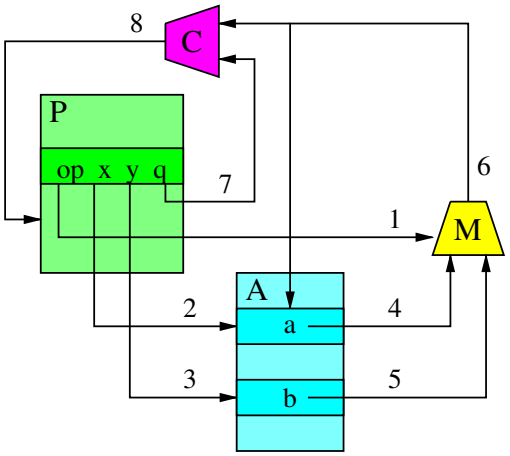
P - programmemória
A - adatmemória
M - műveletvégző
1 - művelet
2,3 - adat címe
4,5 - adat értéke
6 - eredmény értéke

egyszerű gép - következő utasítás címe



P - programmemória
A - adatmemória
M - műveletvégző
1 - művelet
2,3 - adat címe
4,5 - adat értéke
6 - eredmény értéke
8 - következő cím

egyszerű gép



P - programmemória
A - adatmemória
M - műveletvégző
C - címkiszámító
1 - művelet
2,3 - adat címe
4,5 - adat értéke
6 - eredmény értéke
7 - következő
8 - következő cím

Adatábrázolás, kettes számrendszer

Adatábrázolás kritériumai:

- hatékony tárolás
- egyértelműség (könnyen értelmezhető)
- egyszerű, gyors műveletvégzés

A kettes számrendszer előnyei:

- technikai okok: legjobban megkülönböztethető állapotok
- elvi, matematikai okok: "tömörség"
  hány darab számjegy (d), hány féle számjegy (f) ; $x = f^d$

számolás véges pontosságú számokkal

- véges és fix pontosságú számok
- véges pontosságú számok (finite precision numbers)
- zárt (closure): "halmazon belül marad"
  - túlcsordulás (overflow) , alulcsordulás (underflow)
  - "másféle" szám

Példa

Az egész számok halmazára az összeadás, szorzás, kivonás művelete zárt, az osztás nem.
A természetes számok halmazára az összeadás, szorzás művelete zárt, a kivonás és az osztás nem.
A véges pontosságú számok halmazára az összeadás, szorzás, kivonás, osztás művelete nem zárt.

számrendszer alapszáma (radix)
legfontosabb alapszámok: 10 (tizes), 2 (bináris), 16 (hexadecimális)
$d_n...d_2 d_1 d_0 . d_{-1} d_{-2}...d_{-k}$
szám = $\sum_{i=-k}^{n} d_i \times 10^i$
A k alapú számrendszerhez k számjegy (szimbólum):

- tizes: 0 1 2 3 4 5 6 7 8 9
- bináris: 0 1
- hexadecimális: 0 1 2 3 4 5 6 7 8 9 A B C D E F

Példa

$2006 = 2 \times 10^3 + 0 \times 10^2 + 0 \times 10^1 + 6 \times 10^0 = 2000 + 0 + 0 + 6$
$2006 = 1 \times 2^{10} + 1 \times 2^9 + 1 \times 2^8 + 1 \times 2^7 + 1 \times 2^6 + 0 \times 2^5 + 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 = 1024 + 512 + 256 + 128 + 64 + 0 + 16 + 0 + 4 + 2 + 0 = 11111010110_2$
$2006 = 7 \times 16^2 + 13 \times 16^1 + 6 \times 16^0 = 1792 + 208 + 6 = 7D6_{16}$

Negatív bináris számok

- előjeles nagyság (signed magnitude)
- egyes komplemens (ons's complement)
- kettes komplemens (two's complement)
- 2^{m-1} többletes (excess 2^{m-1})

Példa

dec.	*előjeles*	*1-es komp.*	*2-es komp.*	*128 többlet.*
5	*00000101*	*00000101*	*00000101*	*10000101*
-5	*10000101*	*11111010*	*11111011*	*01111011*
100	*01100100*	*01100100*	*01100100*	*11100100*
-100	*11100100*	*10011011*	*10011100*	*00011100*

- fixpontos számábrázolás...

lebegőpontos számok

Tudományos jelölés

n = $f \times 10^e$

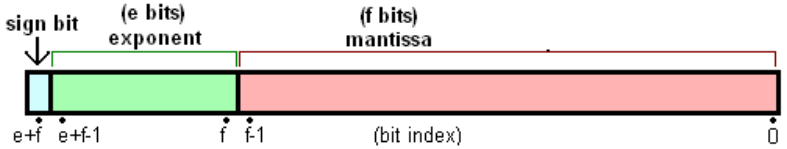
- n lebegőpontos szám (floating point)
- f törtrész (fraction) vagy mantissza (mantissa)
- e egész szám, kitevő vagy exponens (exponent)

Példa

háromjegyű törtszám 0,1 ≤ | f | < 1 vagy 0, exponens kétjegyű

$..-0,999 \times 10^{99}..-0,100 \times 10^{-99}..0..0,100 \times 10^{-99}..0,999 \times 10^{99}..$

IEEE 754 szabvány



A szám értéke v:

v = $s \times m \times 2^e$

ahol

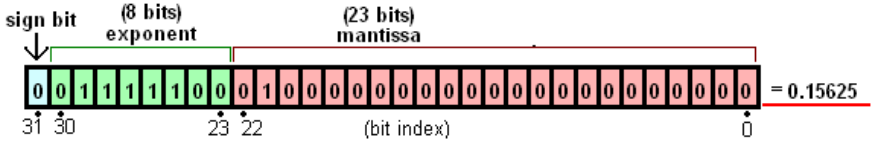
s = +1 (pozitív szám) alol az előjel bit 0

s = -1 (negatív szám) ahol az előjel bit 1

e = exponent - 127 (a 127 többletes ábrázolás miatt)

m = 1.mantissa binárisan ($1 \leq m < 2$), egyesekre normalizált forma

IEEE 754 szabvány szám példa (1)



előjel bit: 0 ; s = +1

exponens: 01111100_2 = 124 ; e = 124-127 = -3

mantissza: 01 ; m = $1,01_2$

v = $s \times m \times 2^e$

v = $+ 1,01_2 \times 2^{-3}$ = +1,25 / 8 = +0,15625

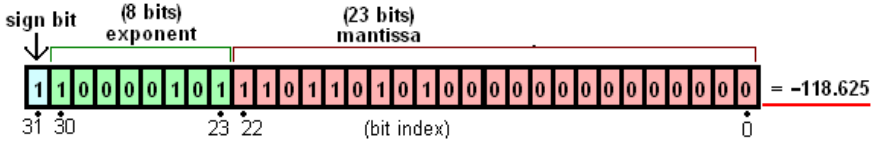
IEEE 754 szabvány szám példa (2)

v = -118,625 = $-1 \times 1110110.101_2 = 1.110110101 \times 2^6$

-1 negatív szám: s = 1

m = 1.110110101 = 1.mantissza ; mantissza = 110110101

e = 6 ; exponens = 6 + 127 = 133 = 10000101_2



IEEE 754 számok

Jellemző	Egyszeres pontosság	Dupla pontosság
Szám hosszúság	32 bit	64 bit
Előjel	1 bit	1 bit
Kitevő (exponens)	8 bit	11 bit
Törtrész (mantissza)	23 bit	52 bit
Kitevő ábrázolása	127 többletes	1023 többletes
Kiterjedés	kb. $10^{-38} - 10^{38}$	kb. $10^{-308} - 10^{308}$

Típus	Exponens	Mantissza
Nulla	0	0
Denormalizált szám	0	nem nulla
Normalizált szám	1 to $2^e - 2$	bármi
Végtelen	$2^e - 1$	0
Nem szám (NaNs)	$2^e - 1$	nem nulla

BCD, Grey, hexadecimális

BCD (Binary Coded Decimal)

$1\ 9\ 7\ 3_{10} : 0001\ 1001\ 0111\ 0011_{BDC}$

Grey kód - csak egy bit változik

0000, 0001, 0011, 0010, 0110, 0111, 0101, 0100, ...

Hexadecimális

$107_{10} = 0110\ 1011_2 = 6\ B_{16}$

- ASCII (American Standard Code for Information Interchange):
  7 bites, 32-126 kódú nyomtatható karakterek,
  például 32 = ' ' (szóköz), 48 = '0', 65 = 'A', 97='a'
- UTF-8 (8-bit Unicode Transformation Format):
  változó hossz (1-4 bájt), 1 bájt ASCII, 2 bájt ékezetes betűk
- ISO 8859:
  különféle nemzeti kódtáblák (ISO 8859-2 a magyar)
- EBCDIC: IBM nagygépeken

Egyéb "ábrázolandók"

- képek, filmek, zenék ...
- gondolatok, érzések, illatok ...
- ...

Összefoglalás, fogalmak

- informatikai alapfogalmak, informatika, információ, bit, adat, jel
- algoritmus, elemi, megengedett lépések
- hardver, szoftver
- egyszerű gép
- adatábrázolás

$1/3 = 0.3333333333...$
$1/16_{10} = 0,0001_2$
$1/10_{10} = 0,000\ 1100\ 1100\ 1100\ 1100...{}_2$

Számítógépek architektúrája

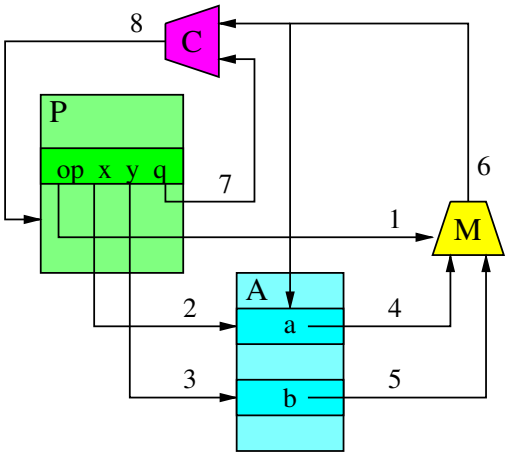
3. előadás

Istenes Zoltán

Eötvös Loránd Tudományegyetem
Informatikai Kar
Programozáselmélet és Szoftvertechnológiai Tanszék

Programtervező Informatikus BSc - B szakirány
2007 / Budapest

Egyszerű gép



P - programmemória
A - adatmemória
M - műveletvégző
C - címkiszámító
1 - művelet
2,3 - adat címe
4,5 - adat értéke
6 - eredmény értéke
7 - következő
8 - következő cím

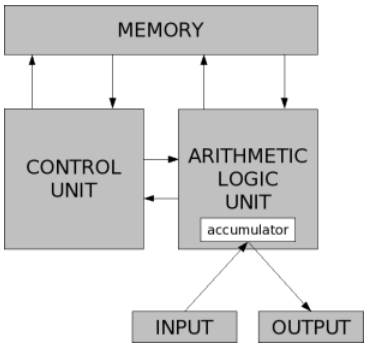
Egyszerű gép - módosítások

- programmemória és adatmemória összevonása
- memória és a központi feldolgozó egység (Central Processing Unit - CPU) különválasztása:
  - műveletvégző egység, aritmetikai logikai egység (Arithmetical Logical Unit - ALU)
  - vezérlő egység (Control Unit - CU)
  - címkiszámító egység
- a memóriát és a központi feldolgozó egységet sínek kötik össze
  - adatsín
  - címsín
  - vezérlősín

Regiszterek

- memória - központi feldolgozó egység sebességkülönbség
- utasítástároló regiszter (Instruction Register - IR)
- utasításszámláló regiszter (Program Counter - PC)
- adatregiszterek

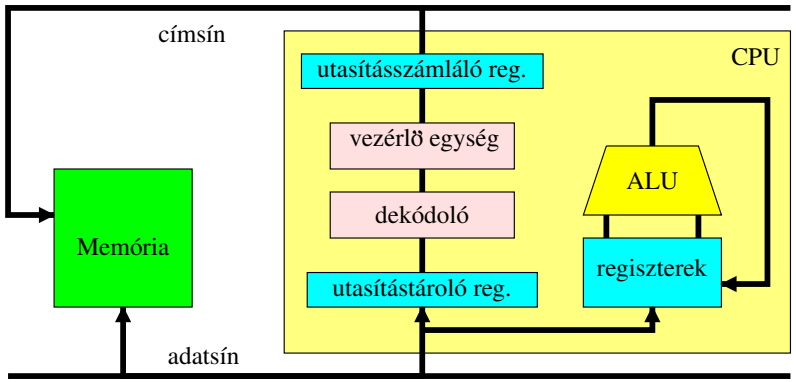
Neumann architektúra



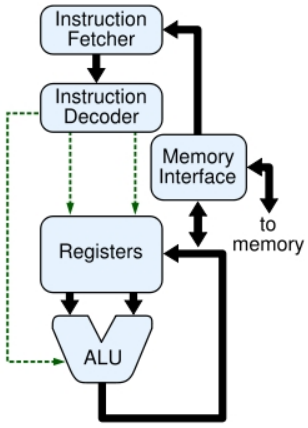
Neumann elv:

- soros utasításvégrehajtás (az utasítások végrehajtása időben egymás után történik)
- kettes (bináris) számrendszer használata
- belső memória (operatív tár) használata a program és az adatok tárolására
- teljesen elektronikus működés
- széles körű felhasználhatóság, alkalmasság bármilyen adatfeldolgozási feladatra

Memória - CPU - sinek blokk diagramm



CPU blokk diagramm



Utasításciklus (instruction cycle, fetch-execute cycle)

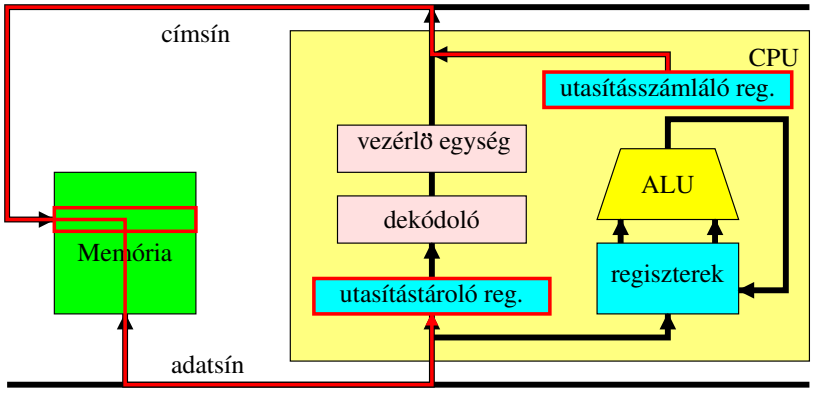
elérés (fetch): minden utasításra azonos

1. utasítás elérése a memóriából
   - programszámláló (program counter, PC)
   - utasítástároló regiszter (instruction register, IR)
2. utasítás dekódolása

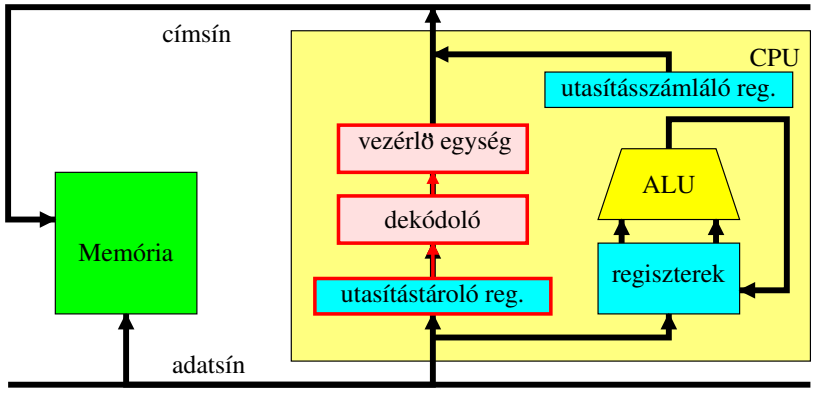
végrehajtás (execute): utasításonként változó

3. operandus(ok) beolvasása a memóriából
4. utasítás végrehajtása
   - vezérlő egység, adatút, regiszterek
   - aritmetikai-logikai egység (arithmetic logic unit, ALU)
5. a művelet eredményének az eltárolása
   - a következő utasítás címének a kiszámítása

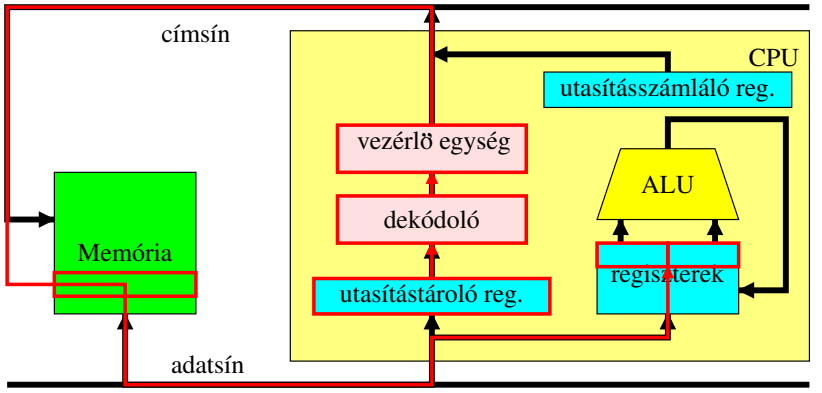
Utasításciklus - utasításelérés



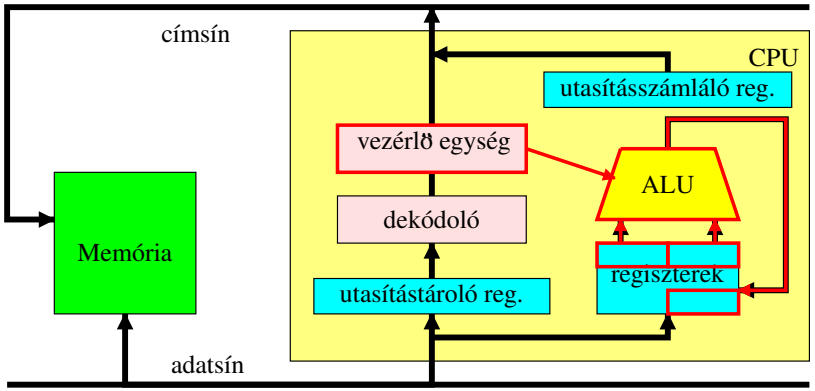
Utasításciklus - utasítás-dekódolás



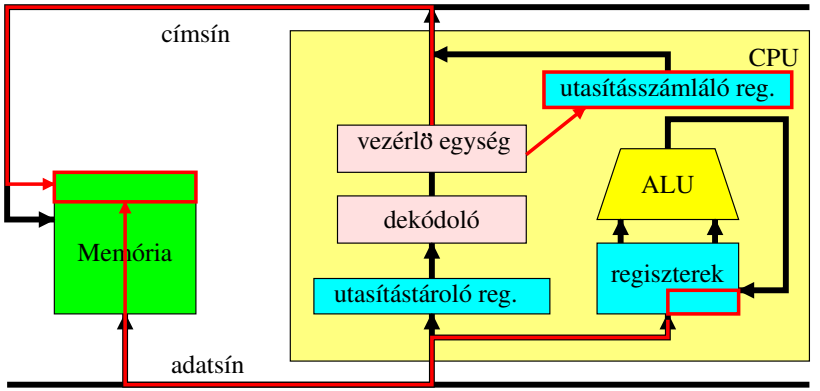
Utasításciklus - operandusok elérése



Utasításciklus - műveletvégzés



Utasításciklus - erdemény eltárolása



utasítástípusok

- adat-töltő, adatátvitel a számítógép funkcionális részei között
- aritmetikai, logikai műveletek végzése adatokkal
- programvezérlő, vezérlésátadás a program egyik részéről egy másik részére
- speciális, rendszervezérlő, hatékony működést segítő
- I/O, adatátvitel a számítógép és a környezete között

Egy utasítás felépítése

műveleti kód	módosításra utaló rész	címrész

4 címes utasításforma

műveleti kód	1. adat címe	2. adat címe	eredmény címe	következő utasítás címe

Példa

- *256 különféle utasítás - 8 bites műveleti kód*
- *16MiB memória - 24 bites címek*
- *utasításhossz - 1 + 4×3bájt = 13 bájt*

3 címes utasításforma

műveleti kód	1. adat címe	2. adat címe	eredmény címe

utasításszámláló regiszter használata

2 címes utasításforma

műveleti kód	1. adat címe	2. adat címe (eredmény címe)

"eredmény címe = 2. adat címe" használata

1 címes utasításforma

műveleti kód	adat címe

- művelet csak 1 adaton (például: növelés 1-el)
- akkumulátor regiszter (Accumulator register):
  - a 2. adat az akkumulátor regiszterből
  - eredmény az akkumulátor regiszterbe

0 címes utasításforma

műveleti kód

művelet az akkumulátor regiszteren (például: növelés 1-el)

jelölések

- S: tárcím
- R: regisztercím
- A: akkumulátor regiszter
- I: közvetlen adat az utasításban
- MK: műveleti kód
- @: tetszőleges művelet
- (S): adat a tárban
- (R): adat a regiszterben
- (A): adat az akkumulátor regiszterben
- LI(n): az n. utasítás hossza
- SI(n): az n. utasítás címe

4 cím	MK S1 S2 S3 S4	(S1) @ (S2) -> (S3) SI(n+1) = S4
3 cím	MK S1 S2 S3	(S1) @ (S2) -> (S3) SI(n+1) = SI(n) + LI(n)
2 cím	MK S1 S2	(S1) @ (S2) -> (S2)
1 cím	MK S1	(S1) @ (A) -> (A)
0 cím	MK	@ (A) -> (A)
1 R cím	MK R	@ (R) -> (R)
1+R cím	MK R S	(R) @ (S) -> (R) vagy (R) @ (S) -> (S)
2 R cím	MK R1 R2	(R1) @ (R2) -> (R1)
3 R cím	MK R1 R2 R3	(R1) @ (R2) -> (R3)
közvetlen operandus	MK I S vagy MK I	I @ (S) -> (S) vagy @ I -> (A)

Címértelmezés és címmegadás

címértelmezés

közvetlen (immediat) adatmegadás	MK I	@ I
közvetett (inherent), implicit címzés	MK	@ (A)
közvetlen (direkt) címzés	MK S	@ (S)
közvetett (indirekt) címzés	MK S	@ ((S))

címmegadás

abszolút címmegadás	MK S	@ (S)
rövidített címmegadás	MK s	@ (s+bázis cím)

Utasítások kódolása - példa

Példa

8 bites utasításhossz, 3 bites címhossz:

- *3 címes címzés: 3×3 > 8 : lehetetlen*
- *2 címes címzés: 8 - 2×3 = 2 : 4 féle utasítás*
- *1 címes címzés: 8 - 3 = 5 : 32 féle utasítás*
- *0 címes címzés: 8 bit : 256 féle utasítás*

Utasítások kódolása

műv. kód 2 bit	1. cím 3 bit	2. cím 3 bit
00	xxx	yyy
01	xxx	yyy
10	xxx	yyy
11	xxx	yyy

műv. kód 2 bit	1. cím 3 bit	2. cím 3 bit
00	xxx	yyy
01	xxx	yyy
10	xxx	yyy
11	000	yyy
11	001	yyy
11	010	yyy
11	011	yyy
11	100	yyy
11	101	yyy
11	110	yyy
11	111	yyy

Utasításkészlet - MIPS 3000 (1988)

Regiszterek (32 bites): $0 .. $31

Formátumok:

Típus	Formátum(bitek)					
R	opcode(6)	rs(5)	rt(5)	rd(5)	shamt(5)	funct(6)
I	opcode(6)	rs(5)	rt(5)	immediate(16)		
J	opcode(6)	address(26)				

Példák:

Típus	Szintaxis	Szemantika (magyarázat)
R	add $1,$2,$3	$1=$2+$3 (regiszter összeadás)
I	addi $1,$2,CONST	$1=$2+CONST (regiszterhez konstans)
I	lw $1,CONST($2)	$1= Mem[$2+CONST] (szó betöltés)
I	sll $1,$2,CONST	$1=$2 < < CONST (bit eltolás $\times 2^{CONST}$)
I	beq $1,$2,CONST	if ($1=$2) goto PC+4+CONST (ugrás ha)
J	j CONST	goto CONST (feltétlen ugrás)

Eltérő címzésű módú gépek programozása példa

Adott 4 különféle típusú gép:

- 0 címes gép (verem), pld.: ADD, PUSH (M), POP (M)
- 1 címes gép, pld.: ADD M = ACC <- (ACC) + (M)
- 2 címes gép, pld.: ADD X,Y = X <- (X) + (Y)
- 3 címes gép, pld.: ADD X,Y,Z = Z <- (X) + (Y)

Műveletek: MOV, ADD, SUB, DIV, MUL, (LDA,STA)
Kiszámolandó: Z:= ((A+B)*C)/((D-E)*F)
Adatok:

- M - 20 bites memóriacím
- X,Y,Z - 20 bites memória, vagy 3 bites regisztercím
- a műveleti kód rész hossza 8 bit

Kérdés mindegyik géptípusra (programra): utasítások száma, program mérete (bit-ben), felhasznált regiszterek száma

Példa eltérő címzésű módú gépek programozására

0 címes gép: `Z=A+B`

`PUSH A`	8+20
`PUSH B`	8+20
`ADD`	8
`POP Z`	8+20
3 utasítás	92 bit

2 címes gép: `Z=A+B`

`MOV R0 A`	8+3+20
`MOV R1 B`	8+3+20
`ADD R1 R2`	8+3+3
`MOV Z R1`	8+20+3
4 utasítás	107 bit

1 címes gép: `Z=A+B`

`LDA A`	8+20
`ADD B`	8+20
`STA Z`	8+20
3 utasítás	84 bit

3 címes gép: `Z=A+B`

`ADD Z A B`	8+20+20+20
1 utasítás	68 bit

Összefoglalás, fogalmak

- központi feldolgozó egység
- egyszerű gép
- Neumann architektúra
- utasításciklus
- utasítástípusok
- utasításkészlet
- címzési módok

Számítógépek architektúrája

4. előadás

Istenes Zoltán

Eötvös Loránd Tudományegyetem
Informatikai Kar
Programozáselmélet és Szoftvertechnológiai Tanszék

Programtervező Informatikus BSc - B szakirány
2007 / Budapest

ÉS, VAGY, NEM logikai kapuk

ÉS (AND)

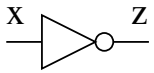
x	y	z
0	0	0
0	1	0
1	0	0
1	1	1

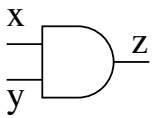
VAGY (OR)

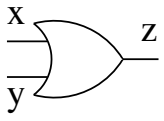
x	y	z
0	0	0
0	1	1
1	0	1
1	1	1

NEM (NOT)

x	z
0	1
1	0







NEM-ÉS, NEM-VAGY, KIZÁRÓ-VAGY logikai kapuk

NEM-ÉS (NAND)

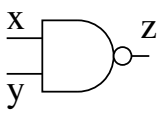
x	y	z
0	0	1
0	1	1
1	0	1
1	1	0

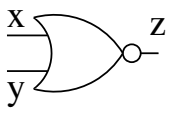
NEM-VAGY (NOR)

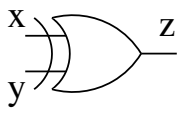
x	y	z
0	0	1
0	1	0
1	0	0
1	1	0

kizáró vagy (XOR)

x	y	z
0	0	0
0	1	1
1	0	1
1	1	0



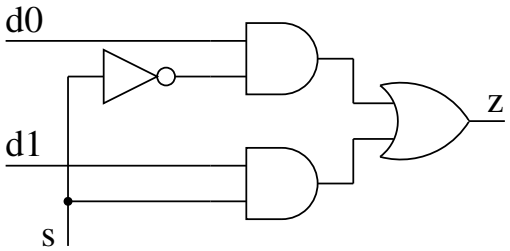


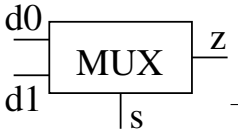


Multiplexer

multiplexer

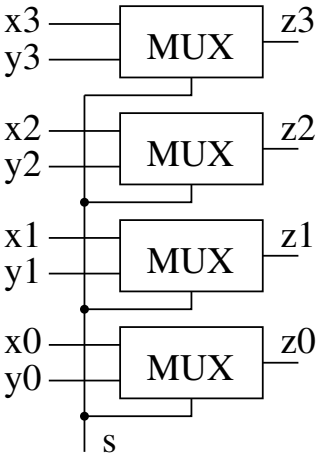
s	d0	d1	z
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

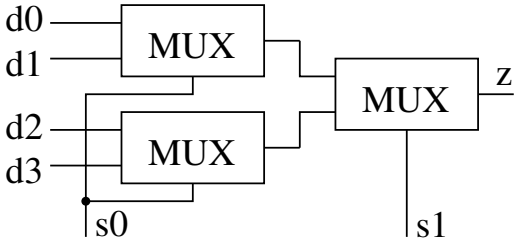




s	z
0	d0
1	d1

4x2 bemenetű és 4 bemenetű multiplexer

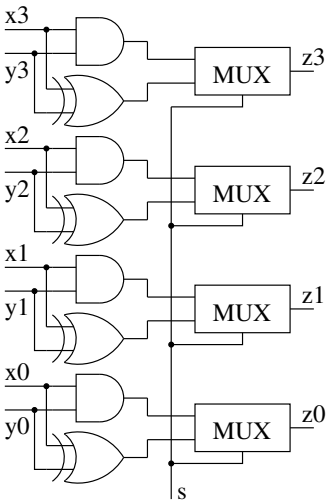




s	z
0	x
1	y

s0	s1	z
0	0	d0
0	1	d2
1	0	d1
1	1	d3

4 bites AND/XOR művelet (logikai egység)



s	z
0	x AND y
1	x XOR y

Összeadás

Reprezentáció?!

0+0=0

0+1=1

1+0=1

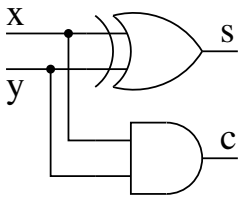
1+1=...

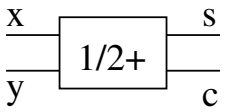
Félösszeadó

félösszeadó

x	y	c	s
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

s = sum = összeg
c = carry = átvitel





Többbites számok összeadása

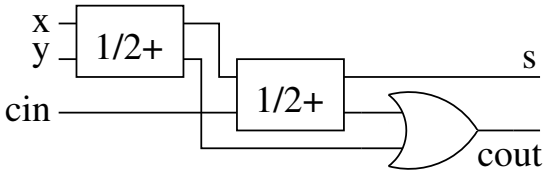
```
 1000
+0110
 1110
```

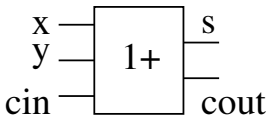
```
 1100   átvitel (carry)
 1011
+0110
10001   fix pontosság...
```


1 bites teljes összeadó

teljes összeadó

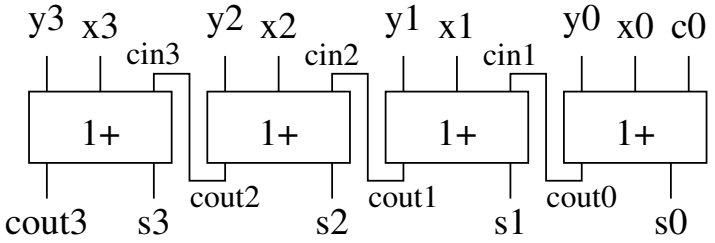
x	y	cin	cout	s
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1





cin = carry in = bemeneti átvitel
cout = carry out = kimeneti átvitel

4 bites összeadó



terjedő átvitel (ripple carry)

s = x + y

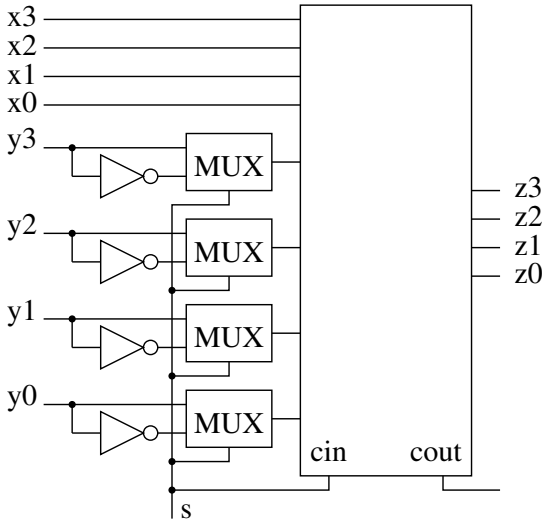
Kivonás kettes komplemensel

5	0101
+2	+0010
7	0111

2	0010	bináris
-2	1101	egyes komplemens
-2	1110	kettes komplemens

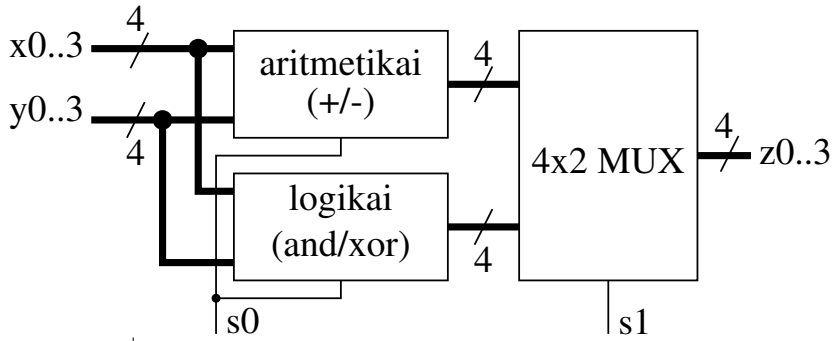
5	0101	
+-2	+1110	kivonás helyett, negált összeadás
3	10011	túlcsordulás (overflow)

4 bites összeadó-kivonó kapcsolás

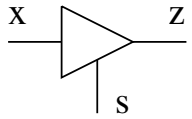


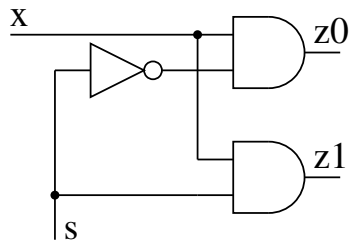
s	z
0	x+y
1	x-y

4 bites aritmetikai-logikai kapcsolás



s0	s1	z
0	0	x+y
0	1	x AND y
1	0	x-y
1	1	x XOR y





kapuzó áramkör

x	s	z
0	0	hi-Z
0	1	0
1	0	hi-Z
1	1	1

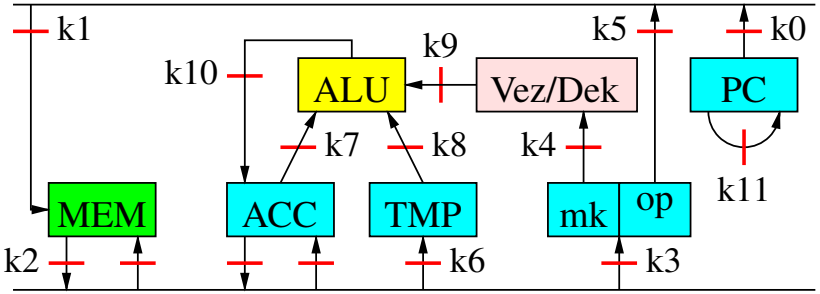
hi-Z = high impedance

demultiplexer

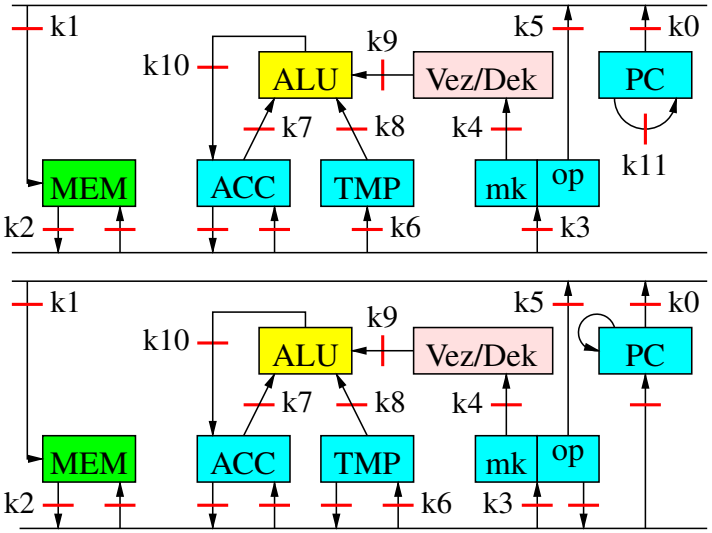
x	s	z0	z1
0	0	0	0
0	1	0	0
1	0	1	0
1	1	0	1

Kapuzás

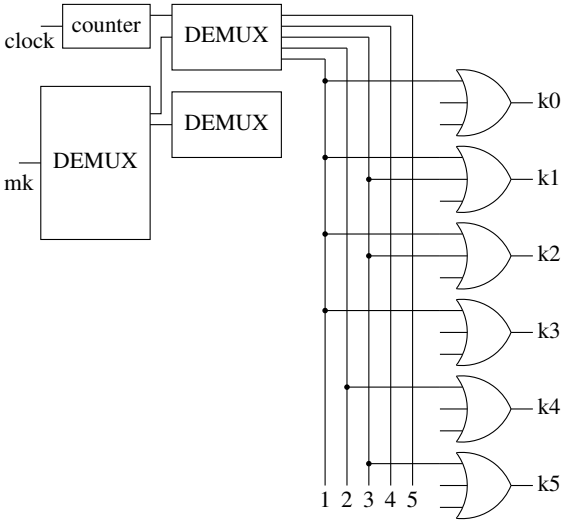
1. utasításelérés: 0,1,2,3
2. dekódolás: 4
3. operandusok beolvasása: 5,1,2,6
4. műveletvégzés: 7,8,9
5. eredmény tárolása, következő: 10,11



Kapuzás összehasonlítás



Huzalozott vezérlő egység



1.
   utasításelérés: 0,1,2,3
2. dekódolás: 4
3. operandusok beolvasása: 5,1,2,6
4. ...

Mikroprogramozott vezérlő egység

mikroprogramtár

.....
111100000000
000010000000
011001100000
000000011100
000000000011
.....

mikroutasítás vezérlő jelek

1. utasításelérés: 0,1,2,3
2. dekódolás: 4
3. operandusok beolvasása: 5,1,2,6
4. műveletvégzés: 7,8,9
5. eredmény tárolása, következő: 10,11

Összefoglalás, fogalmak

- elemi logikai kapuk
- egyszerű logikai kapcsolások, multiplexer, demultiplexer
- összeadás, félösszeadó, teljes összeadó, többbites összeadó
- "kapuzás"
- vezérlő egység
- huzalozott vezérlő
- mikroprogramozott vezérlő

Számítógépek architektúrája

5. előadás

Istenes Zoltán

Eötvös Loránd Tudományegyetem
Informatikai Kar
Programozáselmélet és Szoftvertechnológiai Tanszék

Programtervező Informatikus BSc - B szakirány
2007 / Budapest

Gyorsabb számítógép?

- Órajel-frekvencia
- Adatsín . . . (sávszélesség, átvitel)
- Memória gyorsítása (memória hierarchia)
- Párhuzamosítás utasítás
- Több processzor
- Több regiszter (gyors memória hozzáférés)
- Kevesebb utasításból
- 1 utasításhoz kevesebb órajelciklus
- Műveletvégzés (ALU) gyorsítása
- ...

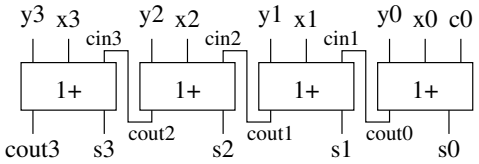
Órajelfrekvencia, idő, távolság

1 „normál” kapu kapcsolási idő $\approx 10ns$

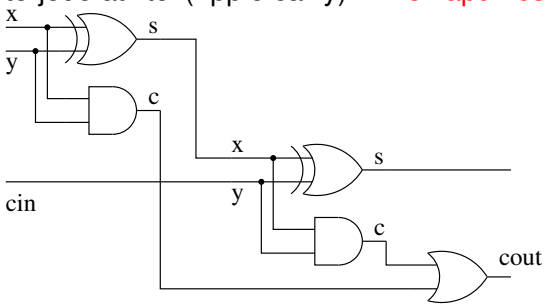
Példa

2400 MHz-es órajel
két órajel közötti idő: $1/(2400 \times 10^6)s \approx 0.4ns$
"fénysebesség": $\approx 300000km/s$
mekkora utat tesz meg a fény két órajel között:
$(300000 \times 10^3 m/s) \times (1/(2400 \times 10^6 s)) \approx 0.125m$

4 bites összeadó



terjedő átvitel (ripple carry) 4 × 3 kapu késleltetés



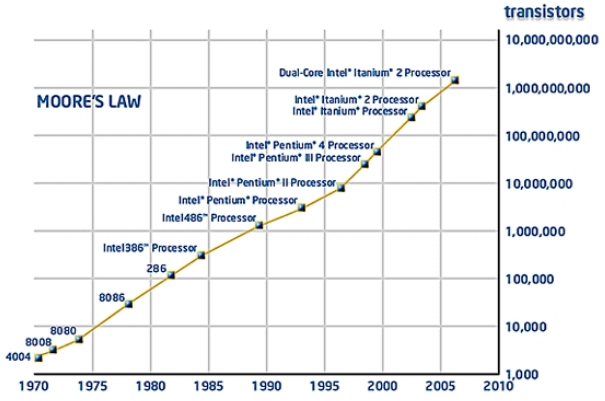
4bites átvitel-előrelátás ("carry lookahead")

Átvitel kiszámítás

Ci+1= (Xi AND Yi) OR (Xi AND Ci) OR (Yi AND Ci)
Ci+1=(Xi * Yi) + (Xi * Ci) + (Yi * Ci)
Ci+1=(Xi * Yi) + ((Xi + Yi) * Ci)
Gi = generate = Xi *Yi
Pi = propagate =Xi + Yi
Ci+1=Gi+Pi*Ci

C1=G0+P0*C0
C2=G1+P1*C1=G1+P1*G0+P1*P0*C0
C3=G2+P2*C2=G2+P2*G1+P2*P1*G0+P2*P1*P0*C0
C4=G3+P3*C3=
=G3+P3*G2+P3*P2*G1+P3*P2*P1*G0+P3*P2*P1*P0*C0
C4 független C1..C3-tól, csak X1..4 , Y1..4 és C0-tól függ

Processzor - memória sebesség különbség



Az integrált áramkörök összetettsége ("sebessége")
18 hónaponként megduplázódik (Moore törvénye).
1986-2000 CPU sebesség: +55%/év , memória: +10%/év

Memória olvasása gyorsítótár nélkül

memória olvasás, gyorsítótár nélkül

1. memória megcímzése
2. memóriából adat kiolvasása

Gyorsítótár sikeres olvasása

gyorsítótár olvasáskor az adat megtalálható (cache hit)

1. cache megcímzése, adat a cache-ban
2. cache-ból adat kiolvasása

Gyorsítótár sikertelen olvasása

gyorsítótár olvasásakor az adat nem található meg (cache miss)

1. cache megcímzése, adat nincs a cache-ban
2. memória megcímzése
3. memóriából adat kiolvasása, adat beírása a cache-ba
4. adat elküldése a CPU-nak

Gyorsítótár írása

írásáteresztés (write-through)

- cache írásakor a memóriába is bekerül az új érték
- "lassú de biztos"

cache - memória koherencia (megegyezőség)

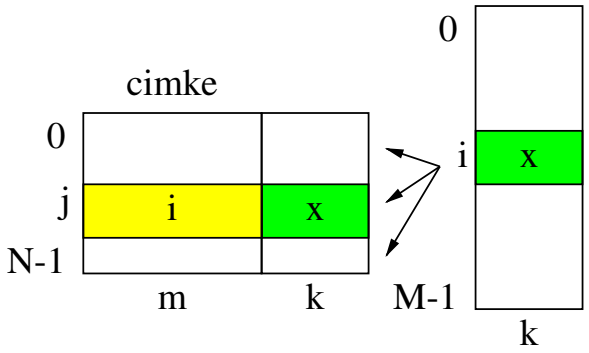
késleltetett írású (write-back)

- cache írásakor bejegyzés a módosult blokkban (piszkos bit - dirty bit),
  a memória nem változik
- cache törlésekor
  - ha korábban módosult a blokk akkor a memóriában is módosítani kell
  - ha nem, akkor a memóriát nem kell módosítani
- "felesleges" memóriaírásoktól mentes

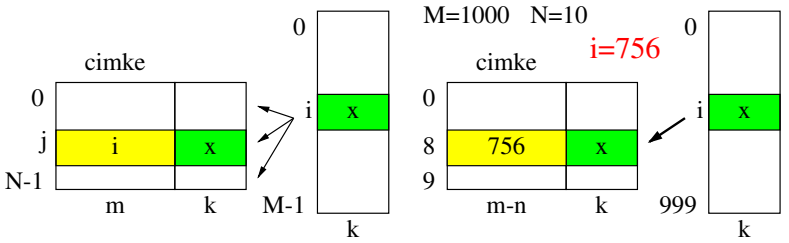
Kérdések, fogalmak a cache-sel kapcsolatban

- találati arány (h)
  - cacheelérés (t_c), memóriaelérés (t_m), átlagos idő (t_a)
  - $t_a = h \times t_c + (1 - h) \times (t_c + t_m)$
  - $t_a = h \times t_c + (1 - h) \times t_m$
- miért működik
  $h = 1/1000$, $t_c = 1ns$, $t_m = 20ns$
  $t_a = 0.001 \times 1 + .999 \times 21 = 1 + 20.979 = 21.979ns$
  - programok helyi lokalitása
  - programok időbeli lokalitása
  - blokkos adatátvitel
- memória leképzése a cache-re
  - teljesen asszociatív
  - közvetlen leképzés
  - halmazasszociatív
- teli cache esetén?

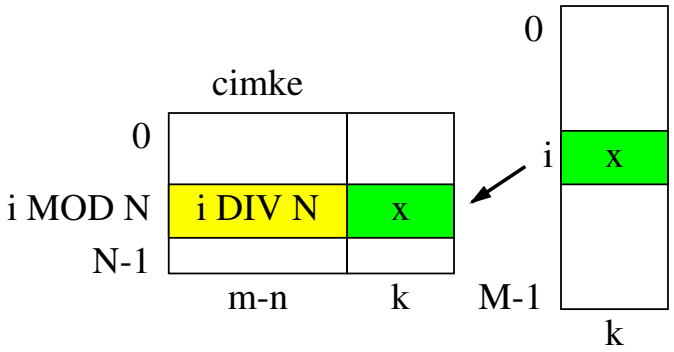
Memória teljesen asszociatív leképzése cache-re



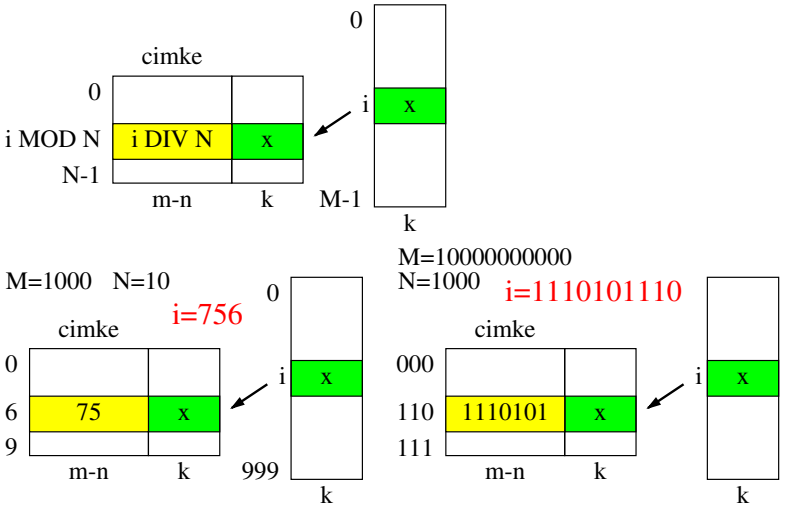
Memória teljesen asszociatív leképzése cache-re



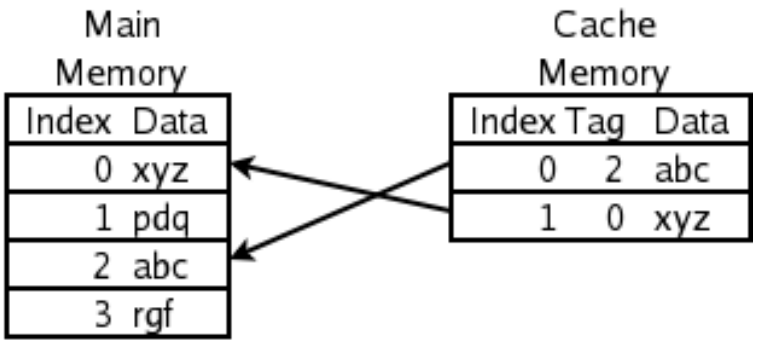
Memória közvetlen leképzése cache-re



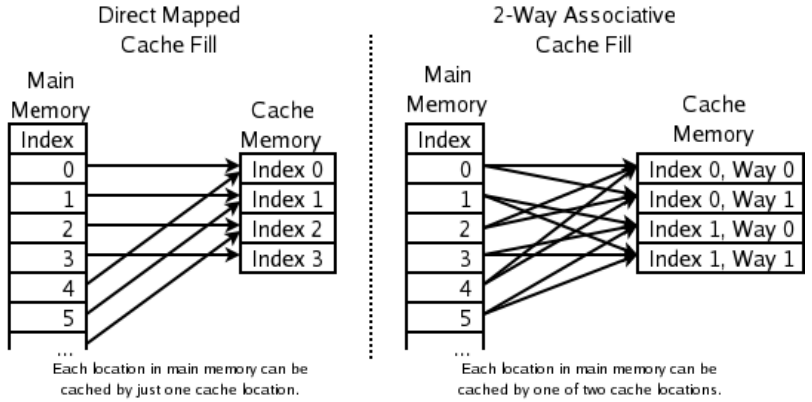
Memória közvetlen leképzése cache-re



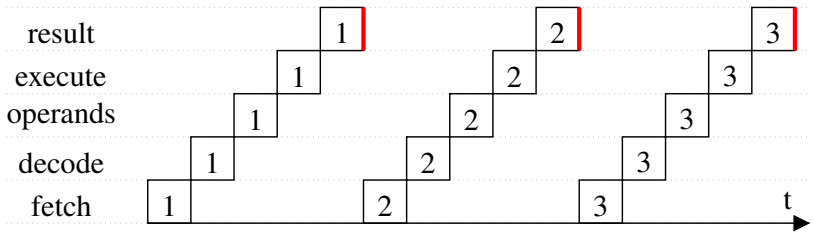
Teljesen asszociatív leképzés



Közvetlen leképzés, halmazasszociatív leképzés



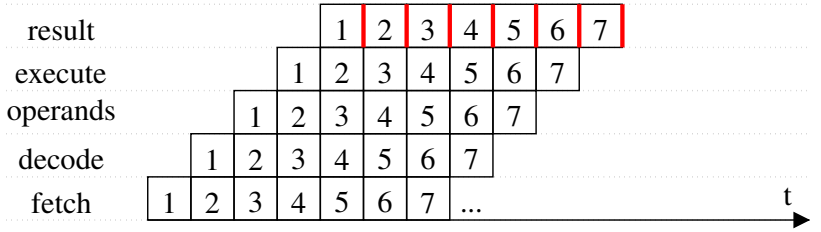
Soros utasítás-feldolgozás



csővezeték követelmények

- részfázisokra bontás
- független részfázisok, önálló erőforrások
- egyik fázis eredménye a következő induló adata

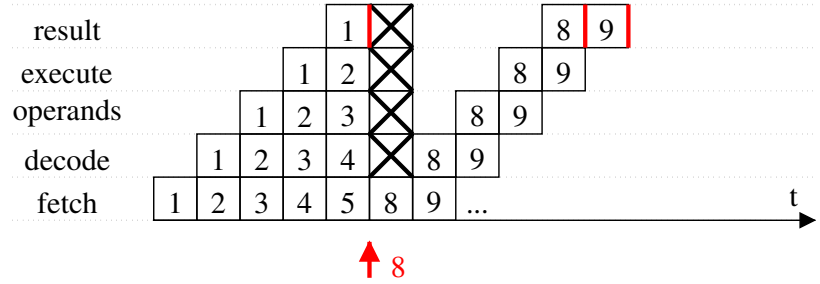
Csővezeték-feldolgozás



csővezeték működése

- ha egy utasítás (1) egy feldolgozási fázisa (fetch) befejeződik,
  - az utasítás (1) egy következő fázisa elkezdődik (decode)
  - a következő utasítás (2) feldolgozása (fetch) elkezdődik
- egy időben több részfázis működik
- egy időben több utasítás dolgozódik fel

Csővezeték-feldolgozás, ugróutasítás esetén



ugró utasítás esetén

- csak az ugró (elágazó) utasítás feldolgozása végén ismert az ugrási cím (nem a következő utasítás)
- a csővezeték már elkezdte feldolgozni a (soron) következő utasításokat
- a csővezetéket ki kell "üríteni" és "újraindítani"

Csővezeték-feldolgozó problémák és megoldások

problémák...

- lépcsők (fázisok) számának a növelése
- nem egyenlő hosszú fázisok
- ugró, vezérlésátadó utasítások, megszakítások
- regiszter (adat) függőségek

megoldások...

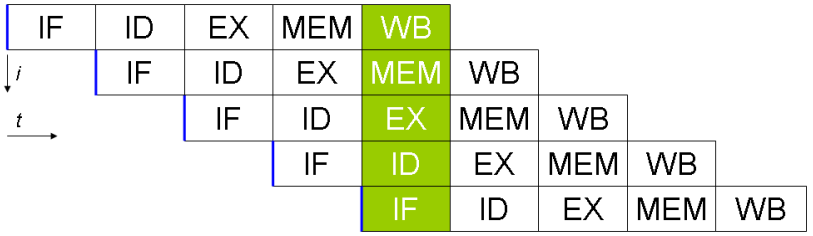
- üres utasítások beiktatása
- elágazásjövendölés (branch prediction)
- utasítások átrendezés

Soros utasítás-feldolgozás

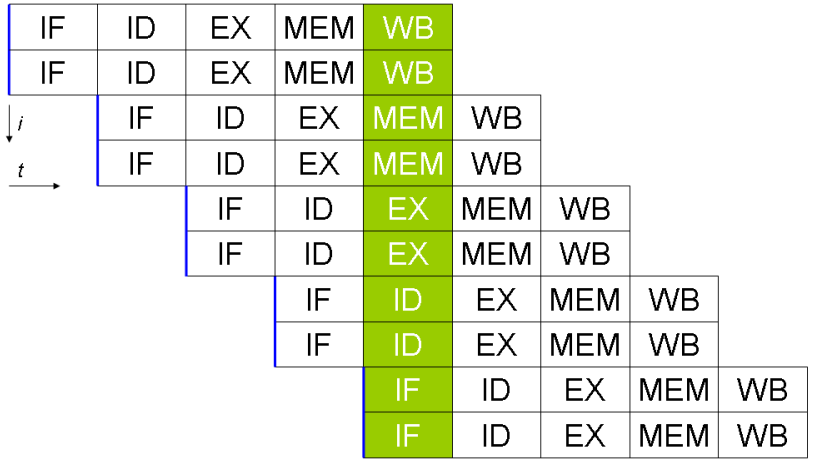
Példa

- *5 "fázisból" ("lépésből") áll egy utasítás feldolgozása (IF,ID,EX,MEM,WB)*
- *az utasítások egymás után következnek*

Csővezeték-feldolgozó



"Szuperskalár" csővezeték-feldolgozó



- több párhuzamos csővezeték

CISC / RISC

CISC

Complete **I**nstruction **S**et **C**omputer
teljes utasításkészletű számítógép

- sok, bonyolult utasítás
- változó utasításhossz
- mikroprogramozott vezérlő
- több órajel/utasítás
- rövid program
- példa: VAX, Intel x86

RISC

Reduced **I**nstruction **S**et **C**omputer
csökkentett utasításkészletű számítógép

- kevés, egyszerű utasítás
- fix utasításhossz
- huzalozott vezérlő
- sok regiszter
- 1 órajel/utasítás
- csővezeték-feldolgozó
- példa: PowerPC, Alpha

Összefoglalás, fogalmak

- gyorsítás, órajelfrekvencia...
- műveletvégzés gyorsítása
- Moore törvénye
- gyorsítótár, találati arány, leképzés memóriára
- csővezeték feldolgozó
- CISC / RISC

Számítógépek architektúrája

6. előadás

Istenes Zoltán

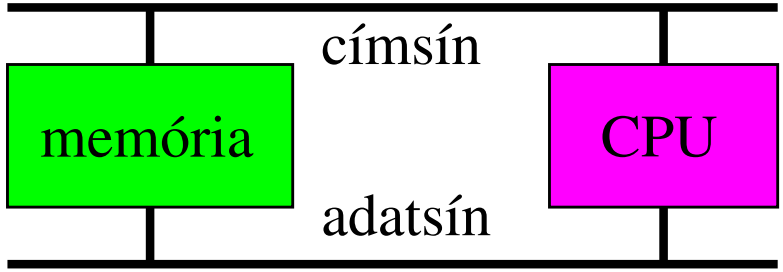
Eötvös Loránd Tudományegyetem
Informatikai Kar
Programozáselmélet és Szoftvertechnológiai Tanszék

Programtervező Informatikus BSc - B szakirány
2007 / Budapest

Memóriák...

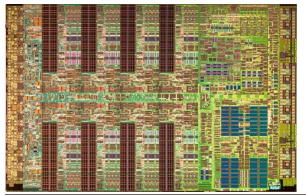
Memória feladata

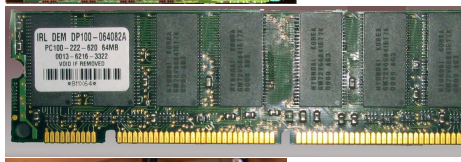
Adatok (több-kevesebb) megőrzése(rövidebb-hosszabb) ideig.



Memória-hierarchia

- CPU, regiszterek,
  1 CPU ciklus, ∼100B
- gyorsítótárak (cache) (L1-3),
  ∼ CPU ciklus, ∼10kB
- központi memória,
  ∼100 CPU ciklus, ∼GB
- lemez háttértár, szalagos,
  optikai, ∼ms, ∼100GB
- "harmadlagos" háttértár,
  lemezegységek ∼10s, ∼TB-PB
- "offline" tár, emberi beavatkozás
  szükséges - $$$



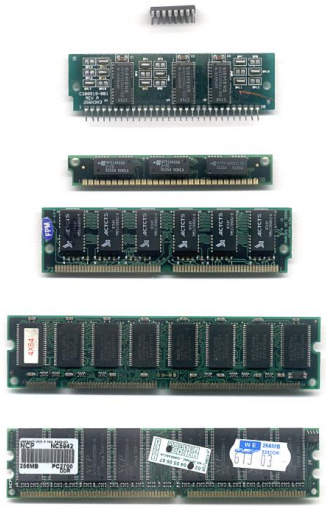


Tárolókezelés feladatai

- a CPU által megcímzett címen lévő érték elérése
- tárhierarchia hatékony működtetése
- virtuális címek kezelése
- lapozás, szegmentálás
- memória "szétosztása"
  - több program,
  - több felhasználó
- "védelem"
  - rendszerprogramok a felhasználótól
  - felhasználók adatai
  - felhasználók programjai (de közös eljárások)

Memória régen...

- IBM 650 (1950-ben a legjobb tudományos számítógép)
  2000 szavas memória
- ALGOL fordítóprogram
  1024 szavas gépre
- PDP-1 időosztásos rendszer (!)
  4096 db 18 bites szó
- 1961 virtuális memória...



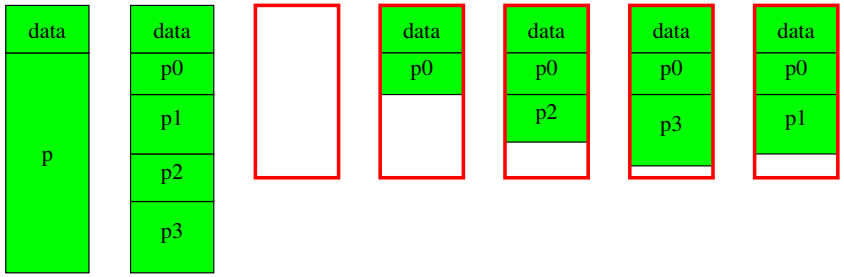
Átlapolásos (overlay) technika

Probléma

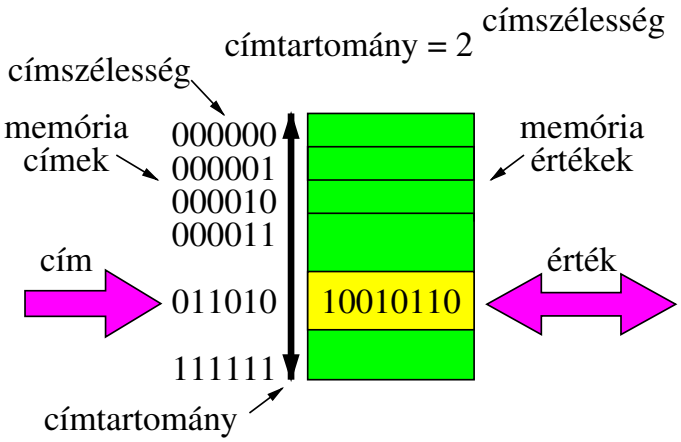
Nagy program - kicsi memória

Megoldás

Program feldarabolása, csak a szükséges darab(ok) a memóriában



Cím, címartomány, címszélesség

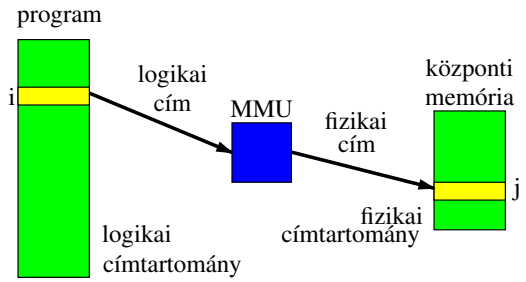


Logikai - fizikai címek

Logkai cím(tartomány)
- CPU által címezhető
- programban használt

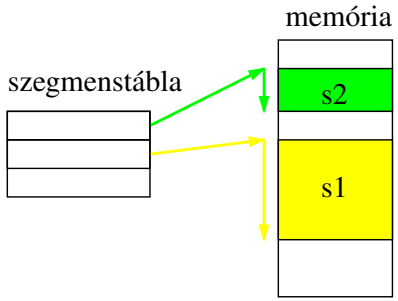
Fizikai cím(tartomány)
- központi memória mérete
- adott géptől függ

Logikai - fizikai címek megfeleltetése



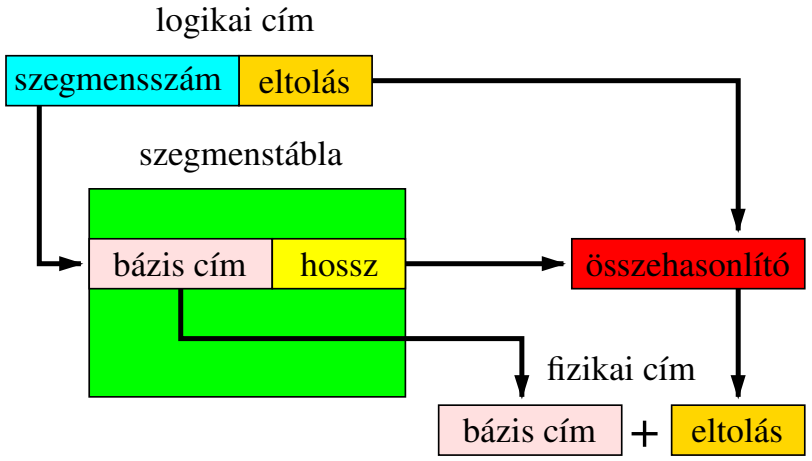
Memóriakezelő egység (Memory Management Unit - MMU)

Szegmentált memória kezelése

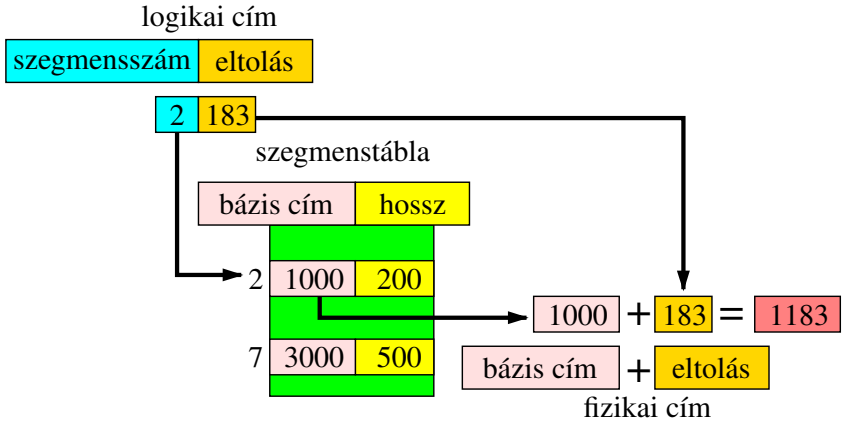


- szegmens: tetszőleges méretű memória blokk
- cím meghatározása: szegmens meghatározása + szegmensen belüli "eltolás" (offset) meghatározása
- szegmenstáblában: szegmens kezdőcíme + szegmens hossza
- üres memóriatartományok kezelése

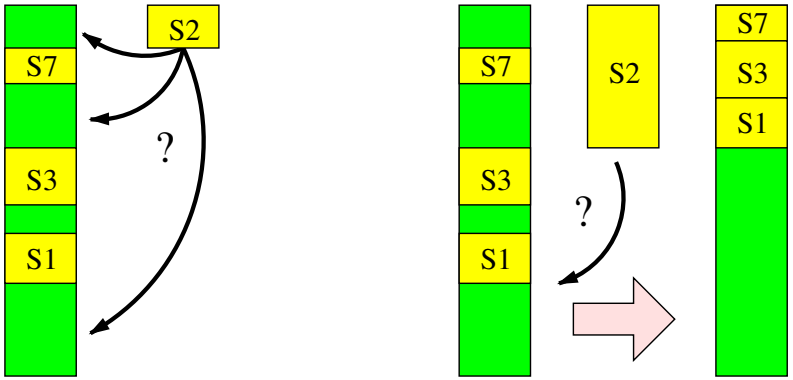
Logikai cím - szegmenstábla - fizikai cím



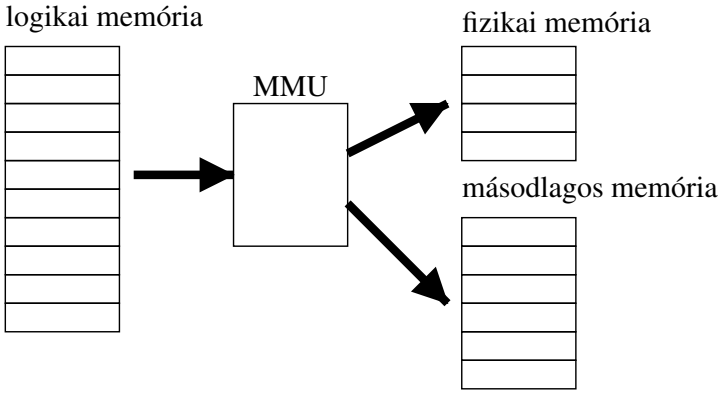
szegmentált címkiszámítás - példa



Szegmens elhelyzése, "memória-kompaktálás"

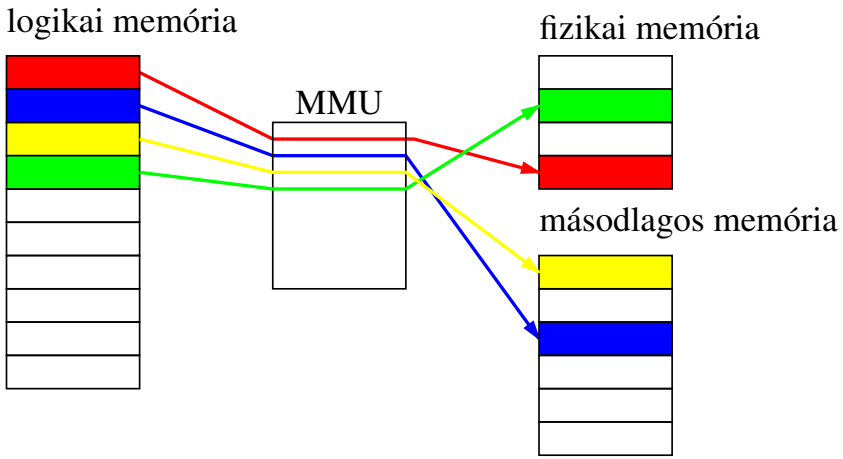


Lapozásos virtuális memóriakezelés



lapok: azonos (logikai, fizikai) rögzített méretű adatblokkok

Lapozásos virtuális memóriakezelés

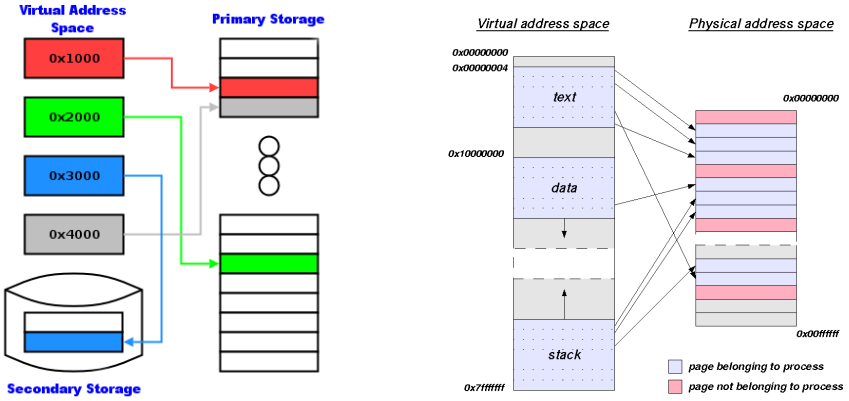


logikai lapok - fizikai lapkeretek

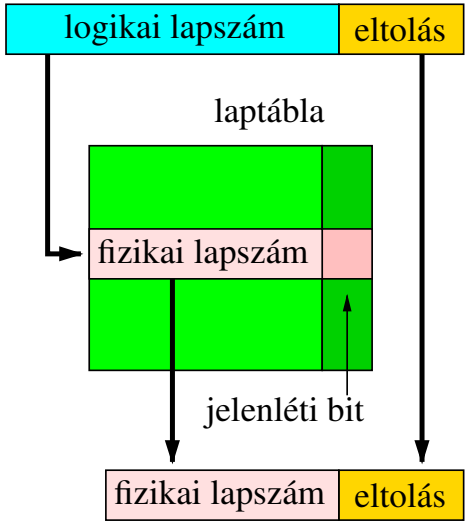
Kapcsolódó fogalmak

- logikai - fizikai címtartomány
- memóriakezelő egység (Memory Management Unit - MMU)
- lapozás (paging), lapcsere
- laptábla "
- cserehely" (swap)
- jelenléti bit (present bit)
- laphelyettesítési eljárás

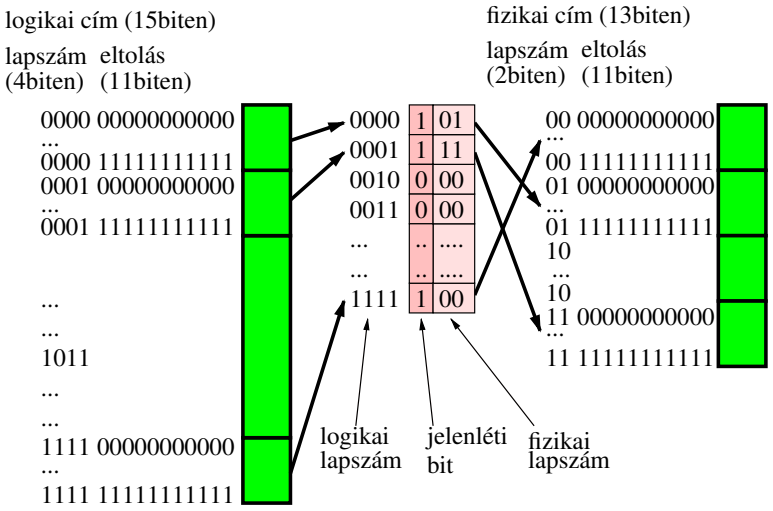
Lapozásos virtuális memóriakezelés



Logikai cím - laptábla - fizikai cím



Lapozásos címkiszámítás - példa



Lapozási mechanizmus

Ha egy lap nincs a memóriában:

1. laphiba (page fault)
2. megszakítás
3. a kért lap betöltése a háttértárolóról

Ha nincs hely a lapnak a memóriában:

- laphelyettesítési eljárás (egy lap eltávolítása...)
  - optimális ("ha tudnánk előre a lapkérelmeket")
  - legrégebben bentlévő (FIFO)
  - legrégebben használt (LRU)
  - legritkábban használt
  - ...

megvalósítás bonyolultsága, tárigénye

Optimális laphelyettesítés - példa

Optimális laphelyettesítés 3 lapkerettel

lapkérelmek	2	3	2	4	6	2	5	6	1	4	6
lapkeret 1	>2	2	2	2	2	2>	>5	5>	>1	1	1
lapkeret 2		>3	3	3>	>6	6	6	6	6	6	6
lapkeret 3				>4	4	4	4	4	4	4	4
laphibák	1	2		3	4		5		6		

találati arány (hit ratio) = 5/11

FIFO laphelyettesítés - példa

FIFO laphelyettesítés 3 lapkerettel

lapkérelmek	2	3	2	4	6	2	5	6	1	4	6
lapkeret 1	>2	2	2	2>	>6	6	6	6>	>1	1	1
lapkeret 2		>3	3	3	3>	>2	2	2	2>	>4	4
lapkeret 3				>4	4	4>	>5	5	5	5>	>6
laphibák	1	2		3	4	5	6		7	8	9

találati arány (hit ratio) = 2/11

LRU laphelyettesítés - példa

LRU laphelyettesítés 3 lapkerettel

lapkérelmek	2	3	2	4	6	2	5	6	1	4	6
lapkeret 1	>2	2	2	2	2	2	2	2>	>1	1	1
lapkeret 2		>3	3	3>	>6	6	6	6	6	6	6
lapkeret 3				>4	4	4>	>5	5	5>	>4	4
laphibák	1	2		3	4		5		6	7	

találati arány (hit ratio) = 4/11

Belady anomália - példa

FIFO laphelyettesítés 3 lapkerettel

lapkérelmek	3	2	1	0	3	2	4	3	2	1	0	4
lapkeret 1	3	3	3	0	0	0	4	4	4	4	4	4
lapkeret 2		2	2	2	3	3	3	3	3	1	1	1
lapkeret 3			1	1	1	2	2	2	2	2	0	0
laphibák	1	2	3	4	5	6	7			8	9	

FIFO laphelyettesítés 4 lapkerettel

lapkérelmek	3	2	1	0	3	2	4	3	2	1	0	4
lapkeret 1	3	3	3	3	3	3	4	4	4	4	0	0
lapkeret 2		2	2	2	2	2	2	3	3	3	3	4
lapkeret 3			1	1	1	1	1	1	2	2	2	2
lapkeret 4				0	0	0	0	0	0	1	1	1
laphibák	1	2	3			4	5	6	7	8	9	10

További memóriakezelési lehetőségek, tulajdonságok

- többszintű lapozás
- szegmentált lapcímzés kezelése...
  szegmentálás + lapozás
- Translation Lookaside Buffer...
  a legutóbbi logikai-fizikai (lap)címmegfelelés tárolása
- "overhead"
- "working set", "demand paging"...
- "trashing" - multiprogramozás

Összefoglalás, fogalmak

- memória-hierarchia
- tárolókezelő egység
- virtuális memória kezelés
- szegmentálás, szegmenstábla
- lapozás, laptábla, laphiba, laphelyettesítési eljárás

Számítógépek architektúrája

7. előadás

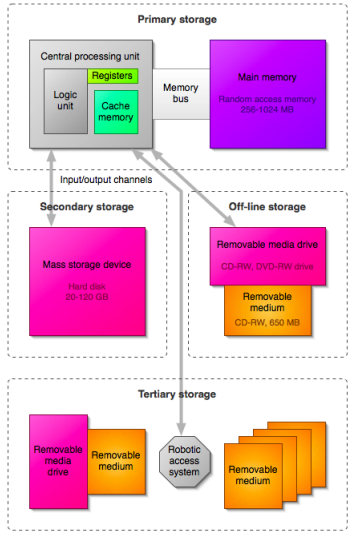
Istenes Zoltán

Eötvös Loránd Tudományegyetem
Informatikai Kar
Programozáselmélet és Szoftvertechnológiai Tanszék

Programtervező Informatikus BSc - B szakirány
2007 / Budapest

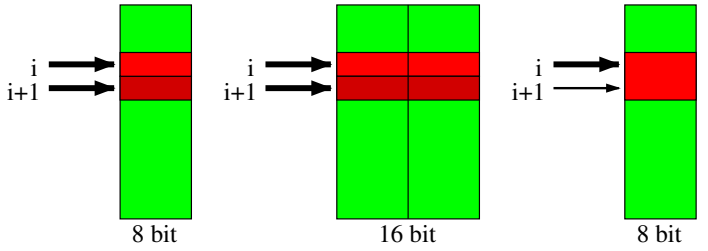
Memória-hierarchia

- CPU, regiszterek,
  1 CPU ciklus, ∼100B
- gyorsítótárak (cache) (L1-3),
  ∼ CPU ciklus, ∼10kB
- központi memória, elsődleges memória ∼100 CPU ciklus, ∼GB
- lemez háttértár, szalagos, optikai, másodlagos memória ∼ms, ∼100GB
- "harmadlagos" háttértár, lemezegységek ∼10s, ∼TB-PB
- "offline" tár, emberi beavatkozás szükséges - $$$



Memóriák "mennyiségi" jellemzése

- kapacitás: tárolható adat mennyisége
- legkisebb címezhető egység
- hozzáférési szélesség
- átlapolhatósági fok: egyszerre működhető modulok

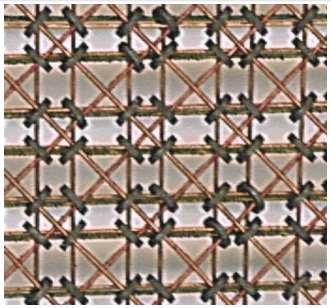


Memóriák "időbeli" jellemzése

- elérési idő: időtartam az olvasási igénytől az adat megérkezéséig
- ciklusidő: két hozzáférés közti minimális idő
- átlapolhatósági fok: egyszerre működhető modulok
- adatátviteli sebesség

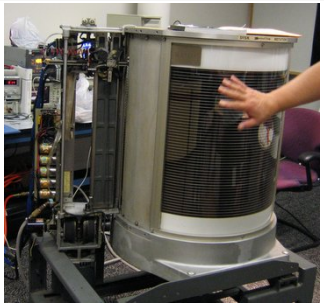
"Változékonyság" (volatility)

- változékony (pld.: központi memória)
- nem változékony (pld.: háttértároló)
- dinamikus: változékony és folyamatos frissítést igényel (pld.: DRAM)

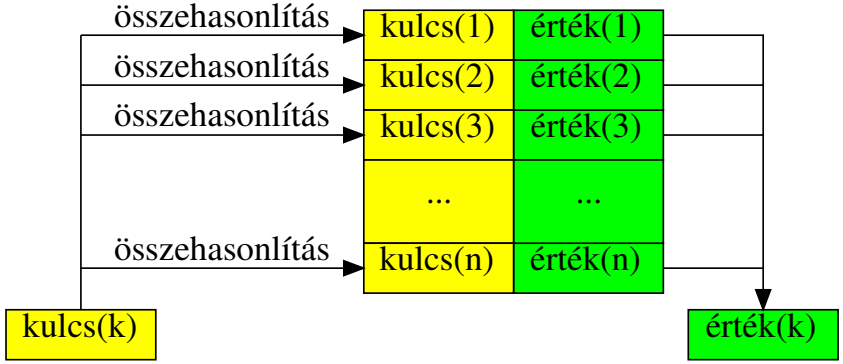


Hozzáférés

- véletlenszerű elérés (random access), "címfüggetlen"
- soros elérés (pld.: szalag)
- ciklikus elérés (pld.: merevlemez)
- tartalom szerint elérhető, asszociatív (CAM)
- egyéb: verem (LIFO), cső (FIFO)



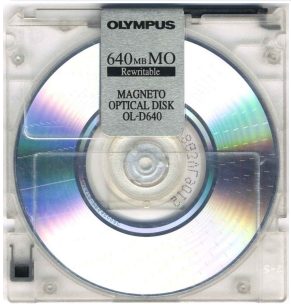
Tartalom szerint elérhető memória



Tartalom szerint elérhető memória - Content addressable memory (CAM)

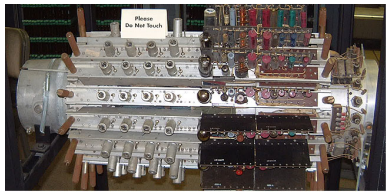
Megváltoztathatóság

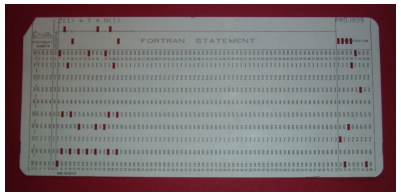
- olvasható, írható (RWM)
- csak olvasható (ROM), csak egyszer írható (pld.: WORM, CD-R)
- lassan írható, gyorsan olvasható (pld.: CD-RW)



Megvalósítás technológiája

- mágneses
- félvezetős
- optikai, mágneses-optikai
- régi technológiák: papír, elektroncső, késleltető vonal
- egyéb: holografikus, molekuláris, ...



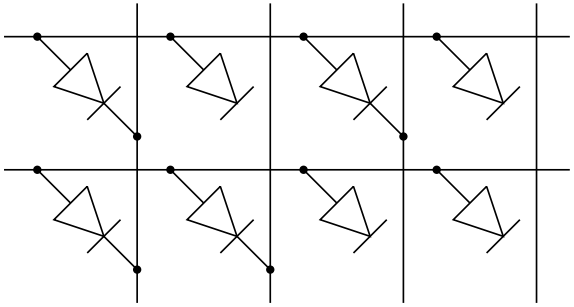


"Bithiba"

- Információ mennyisége? GiB? TiB?
- Információ "változása"? CPU-MEM GiB/s?
- 1 bit hibája:
  "ha nagyon mindegy"
  "ha nagyon nem mindegy"
- hibadetektálás, hibajavítás, többszörös bithiba

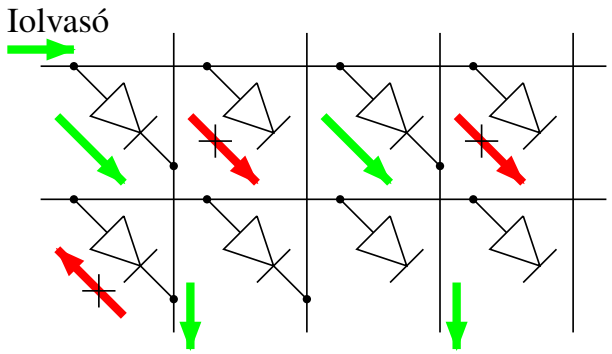
memória tartalom	paritás bit	
10011010	0	
11100000	1	
10110101	1	
00110000	0	
11010110	1	
11001111	1	paritás hiba
00101100	1	
10000101	1	

ROM megvalósítás



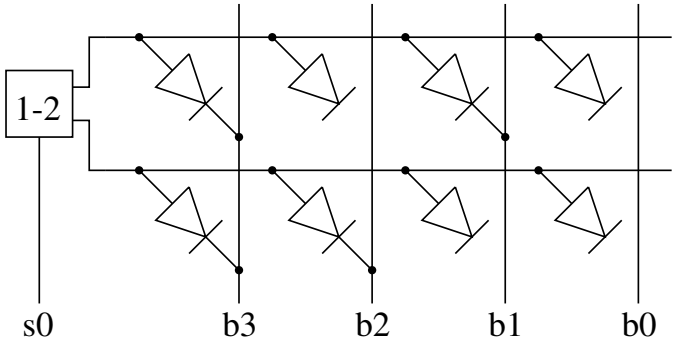
maszkolt ROM

ROM megvalósítás



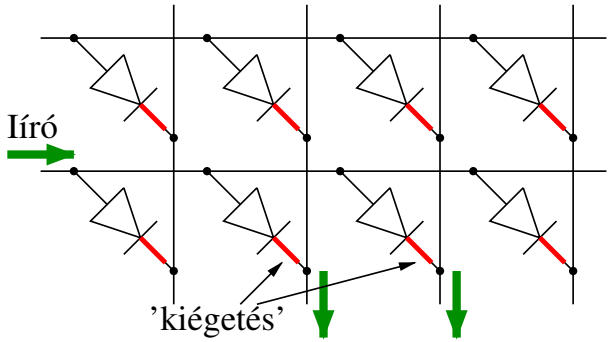
ROM olvasás

ROM megvalósítás



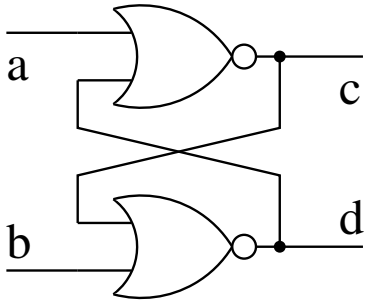
s0	b3	b2	b1	b0
0	1	0	1	0
1	1	1	0	0

PROM megvalósítás



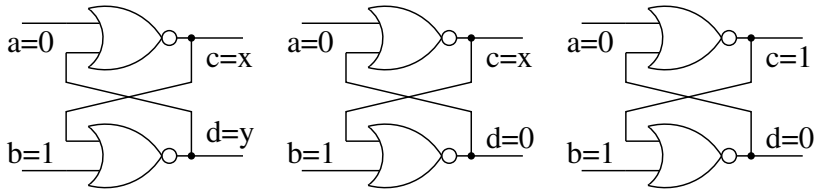
iíró>>Iolvasó

Egy "furcsa" kapcsolás

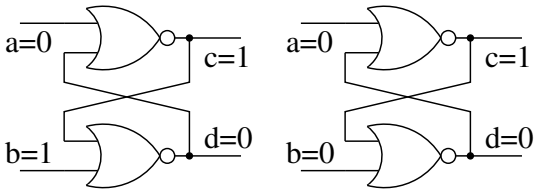


visszacsatolás

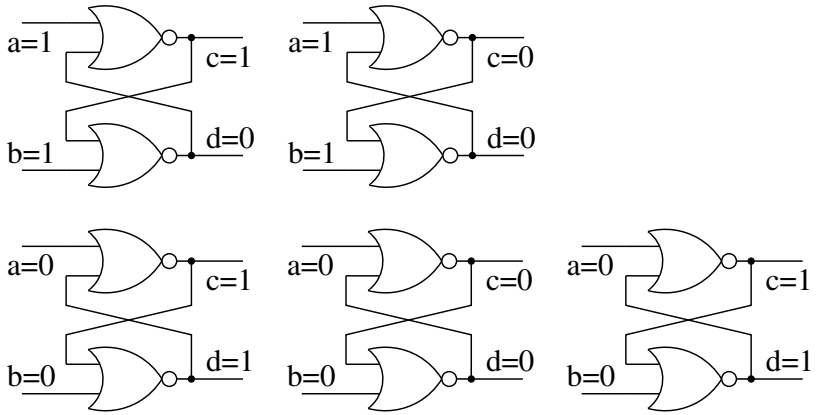
Érték beírása...



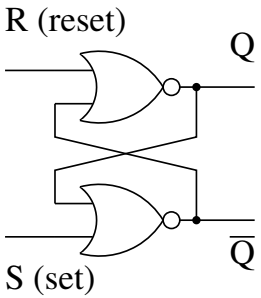
Érték tárolása...

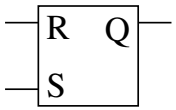


Instabil állapot...



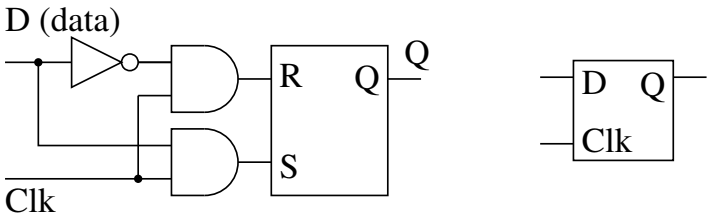
Reset-Set Flip-Flop





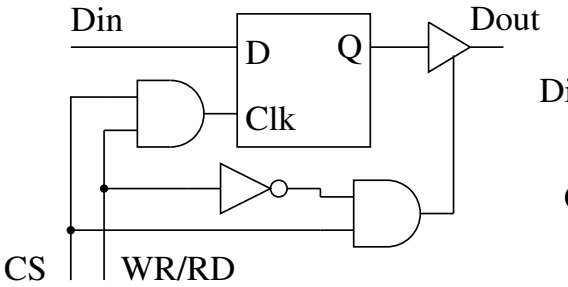
R	S	Q_t	Q_{t+1}	megjegyzés
1	0	x	0	törlés
0	1	x	1	beállítás
0	0	x	x	megőrzés
1	1	x	???	instabil

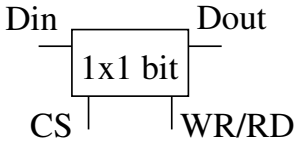
Data Flip-Flop



Clk	D	Q_t	Q_{t+1}	megjegyzés
0	x	y	y	megőrzés
1	x	y	x	tárolás

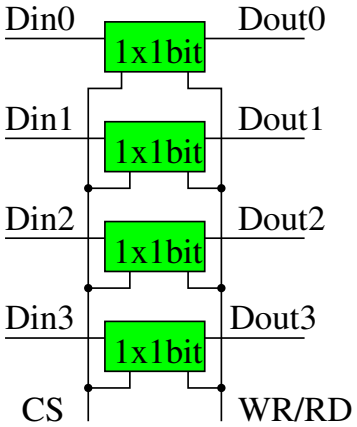
1x1 bit

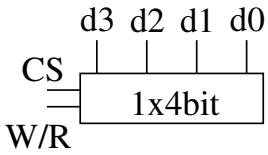




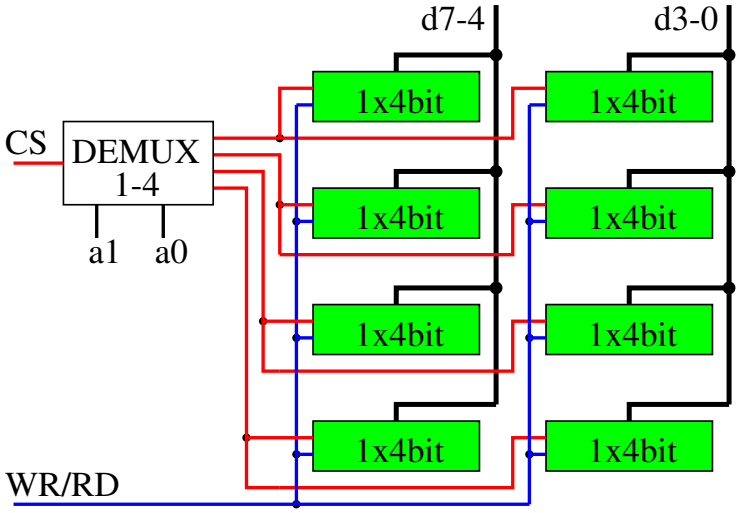
Din	CS	WR/RD	Q_t	Q_{t+1}	Dout	megjegyzés
x	0	y	z	z	hi-Z	inaktív
x	1	1	z	x	hi-Z	tárolás
x	1	0	z	z	z	olvasás

1x4 bit

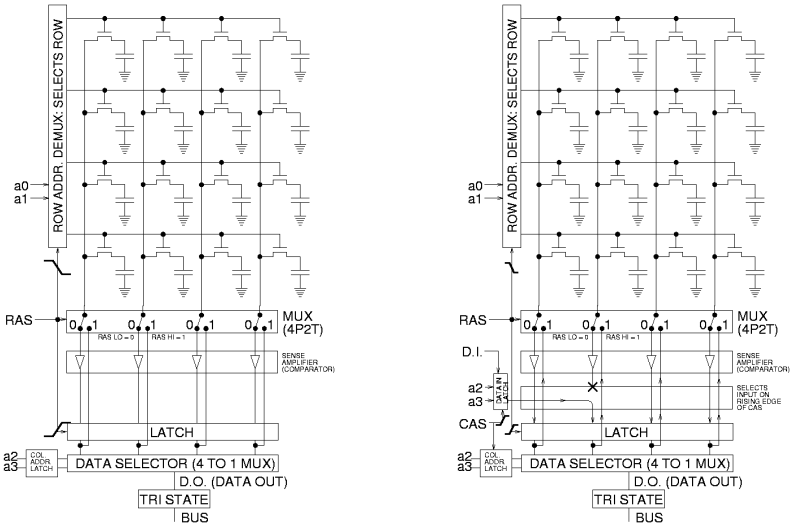




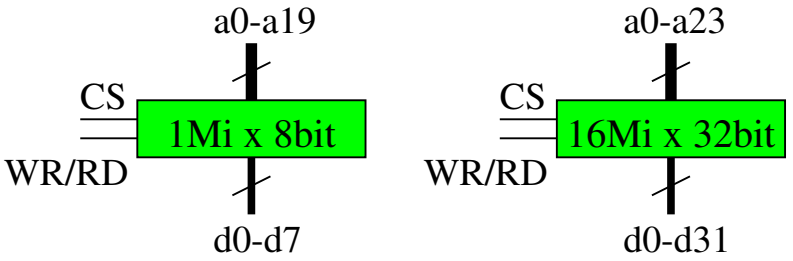
4x8 bit



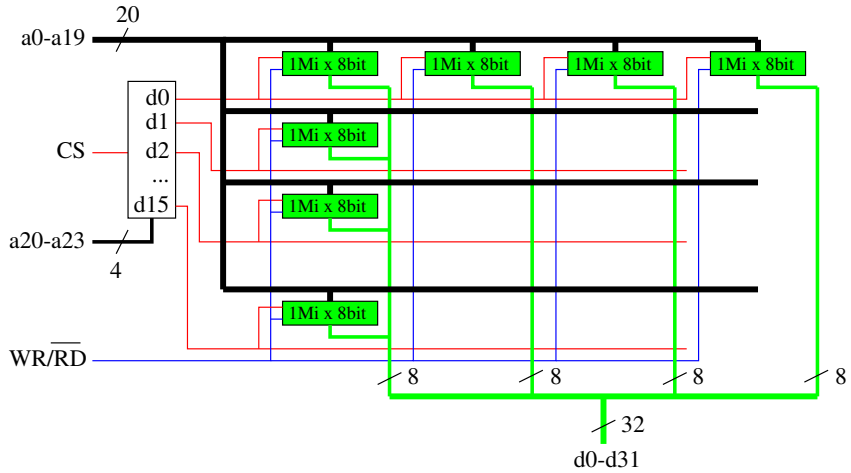
DRAM 4x4 bit



1Mi x 8bit → 16Mi x 32bit

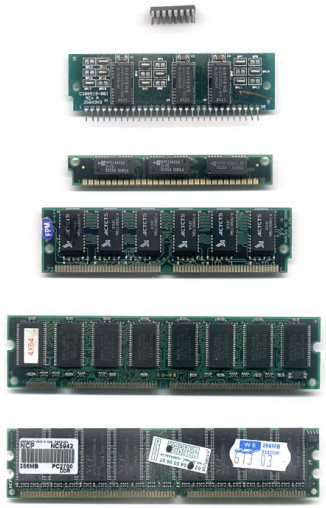


1Mi x 8bit → 16Mi x 32bit



Összefoglalás, fogalmak

- memóriák csoportosítása, jellemzése
- bithiba
- ROM, RWN,RAM, CAM, PROM
- 1x1 bites memória, RSFF, DFF
- "többbites" memóriák



Számítógépek architektúrája

8. előadás

Istenes Zoltán

Eötvös Loránd Tudományegyetem
Informatikai Kar
Programozáselmélet és Szoftvertechnológiai Tanszék

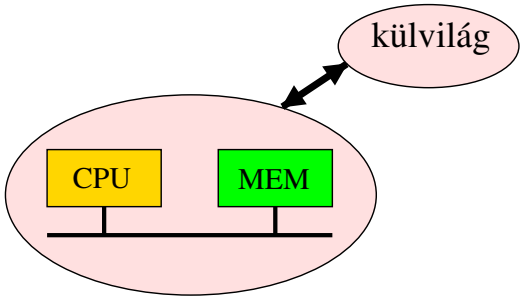
Programtervező Informatikus BSc - B szakirány
2007 / Budapest

CPU, Memória, és még valami...

Számítógép fő számítási funkciója csak a CPU-t és a Memóriát érinti: CPU beolvassa az utasításokat és az adatokat a memóriából, feldolgozza majd az eredményt a memóriában tárolja...

Kérdés

Hogyan kerülnek az adatok a memóriába?



Kapcsolat a "külvilággal"

A kimeneti és bemeneti egység feladata:

információcsere a CPU vagy a központi memória és a külvilág között...

Példa

Elhanyagolható a külvilág?

Példa

Nagyon fontos a külvilág?

A ki/bemeneti egység feladata

- kapcsolatok kezelése
- adatátvitel

processzor - memória - I/O eszköz

- ki/bemeneti eszköz
- input/output device
- I/O eszköz
- periféria

Példák perifériákra?

Célok és problémák...

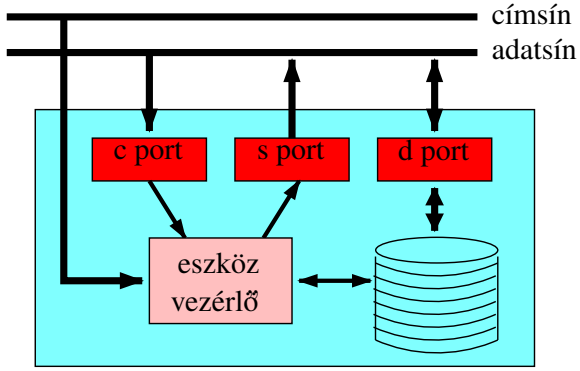
Problémák:

- eszközök (CPU - perifériák) eltérő sebessége...
- eszközök különbözősége (kezelési mód)...

Célok:

- eszközök minél gyorsabb, „jobb” kiszolgálása...
- a CPU minél kisebb leterhelése...

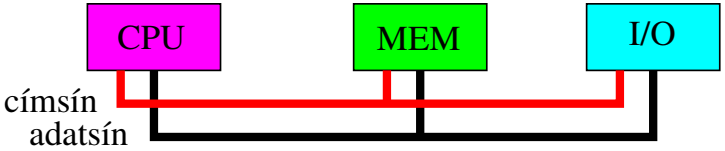
I/O eszközök portjainak az elérése



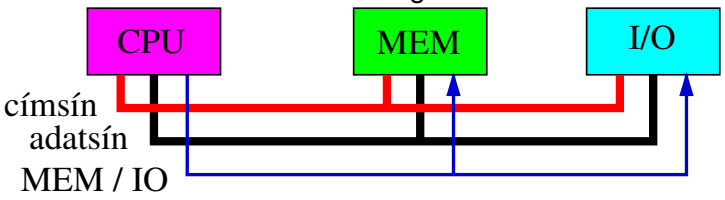
c port: parancsregiszter
s port: állapotjelző regiszter
d port: adat ki/bemeneti regiszter

Perifériák kezelése portokon keresztül

tárolóhoz rendelt módon (memory mapped addressing):
a központi memórián keresztül



közvetlen I/O utasítások:
közvetlenül az eszközvezérlő regiszterébe



Sínrendszer feladata

A sínrendszer feladata:

adatok, vezérlőjelek továbbítása

Átvitel létrehozásakor

- eszközök kijelölése ("cím" megadás...)
- adatátvitel iránya
- eszközök szinkronizálása (működésének összehangolása)

Sínrendszer struktúrája

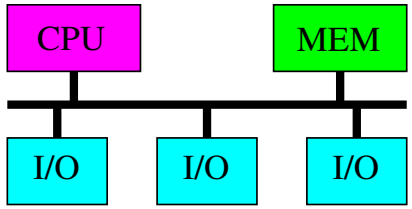
Külső / belső sínrendszer (CPU-hoz képest)

- belső: (pld. 3 sín, külön adatsín írásra, olvasásra...)
- külső:
  - helyi sín (local bus) (pld. co-processzor)
  - rendszersín (system bus) (pld. I/O)
  - memóriasín (memory bus)

Sínrendszer részei:

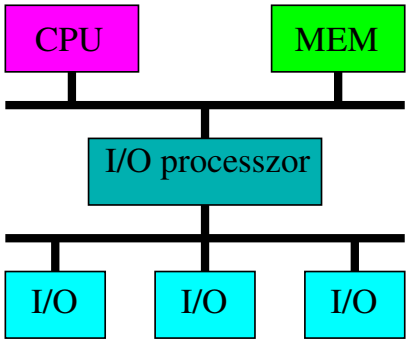
- címsín
- adatsín
- vezérlősín

Sínrendszerek - példa



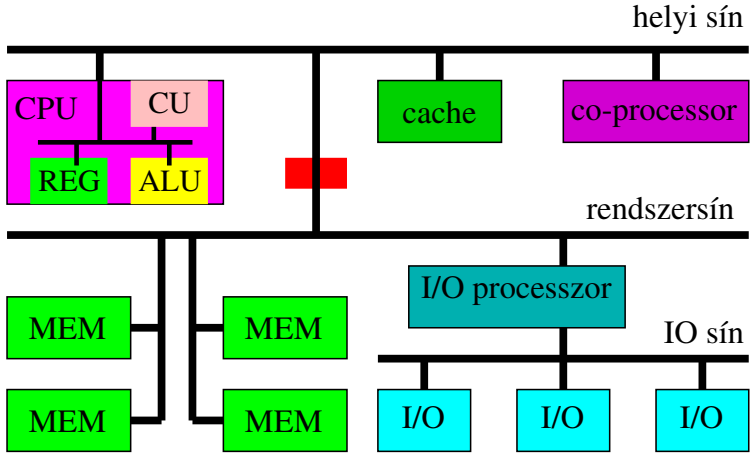
- osztott rendszersín (system bus)

Sínrendszerek - példa



- renszersín, I/O sín

Sínrendszerek - példa

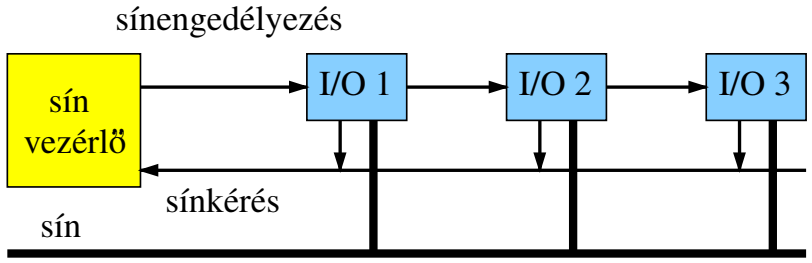


- belső sín, helyi sín, rendszersín, IO sín

Vezérlő jelek

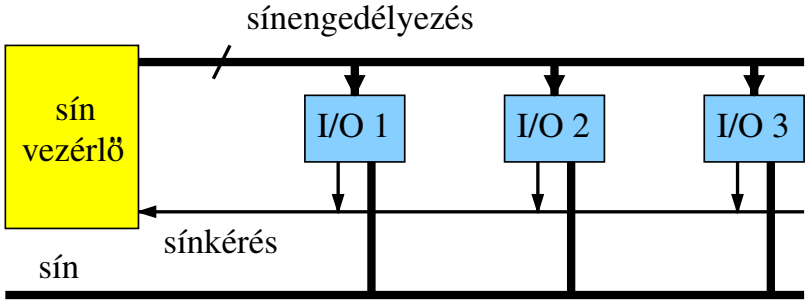
- adatátvitelt vezérlő jelek:
  - memória /periféria M/IO
  - írás / olvasás R/W
  - szó / byte átvitel WD/B
  - cím a sínen
  - adat a sínen
  - átvitel vége
- megszakítást vezérlő jelek
- sínvezérlő jelek (kérés, foglalás, visszaigazolás)
- egyéb... (órajel, ütemezés, táp,...)

Soros kiszolgálás (daisy chain)



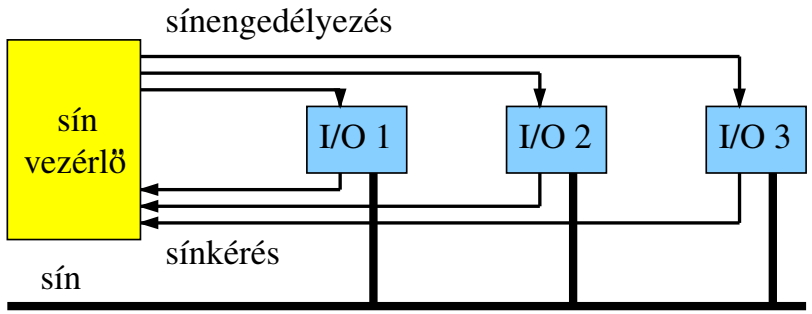
- sínkérés (BREQ - bus request)
- sínengedélyezés (BG - bus grant)

Lekérdezéses kiszolgálás (polling)



- sínkérés (BREQ - bus request)
- lekérdező szám (polling count)

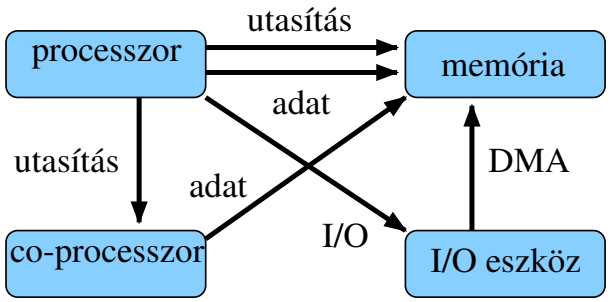
Párhuzamos kiszolgálás (independent requesting)



Fogalmak

- sínprotokoll (mechanikus, elektromos, logikai)
- átlapolódó sínciklusok
- blokk sínciklus (burst cycle)
- sínfoglalás (bus arbitration)
- sínvezérlő (sínmeghajtó) egység (bus interface)
- master / slave

Master / slave sínhasználat



- sínhasználat kezdményezése, aktív eszköz (master)
- sínhasználat végrehajtása, passzív eszköz (slave)

PCI

- 33.33 MHz clock with synchronous transfe, 32-bit or 64-bit bus width, 32-bit address space, 32-bit port space, 256-byte configuration space, 3.3 or 5-volt signaling
- peak transfer rate of 133 MB per second for 32-bit bus width (33.33 MHz $\times$ 32 bits $\times$ 1 byte/8 bits = 133 MB/s)

AGP

- AGP 8x, using a 32-bit channel operating at 66 MHz, strobing eight times per clock, delivering an effective 533 MHz resulting in a maximum data rate of 2133 MB/s (2 GB/s); 0.8 V signaling.

PCI Express, PCIe, or PCI-E

- PCIe transfers data at 250 MB/s per lane to a maximum of 32 lanes, a total combined transfer rate of 8 GB/s. It must be noted that PCIe is able to transfer data in both directions at once (full-duplex). This effectively doubles the data transfer rate allowing 500 MB/s per lane giving a total combined transfer rate of 16 GB/s when 32 lanes are employed.

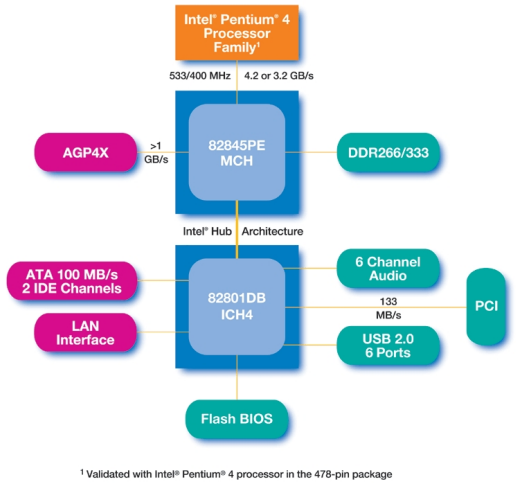
DDR-SDRAM

- PC-3200: DDR-SDRAM memory module specified to operate at 200 MHz using DDR-400 chips, 3.200 GByte/s bandwidth

DDR2-SDRAM

- Standard name: DDR2-1066, Module name: PC2-8500, Memory clock: 266MHz, I/O Bus clock: 533MHz, Peak transfer rate: 8500MB/s

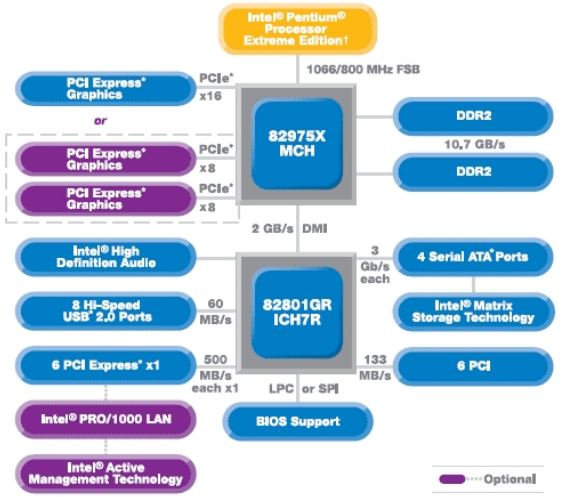
Sínrendszerek - "laptop" példa



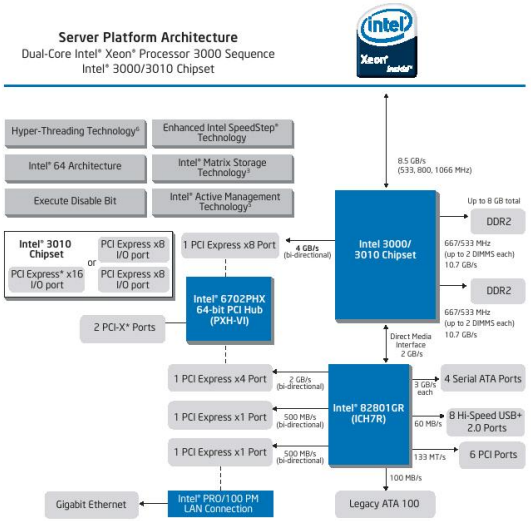
¹ Validated with Intel® Pentium® 4 processor in the 478-pin package

- PCI: 133MB/s
- AGP4x: 1066MB/s
- DDR333: 2667MB/s
- PCI-E: 250MB/s/lane (max 8GB/s)

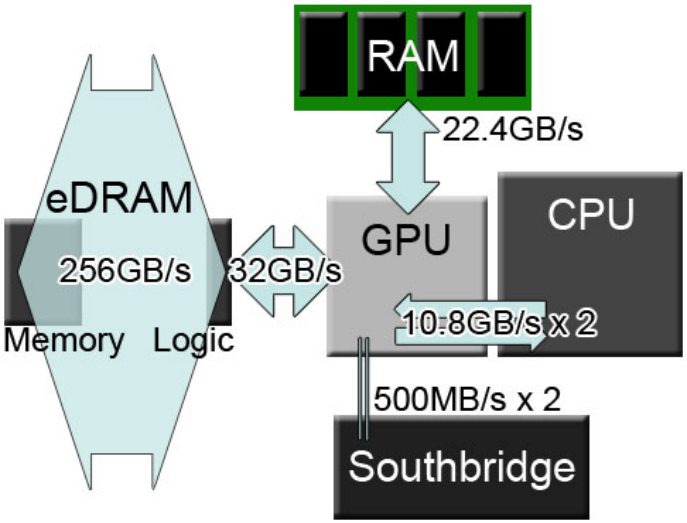
Sínrendszerek - "desktop" példa



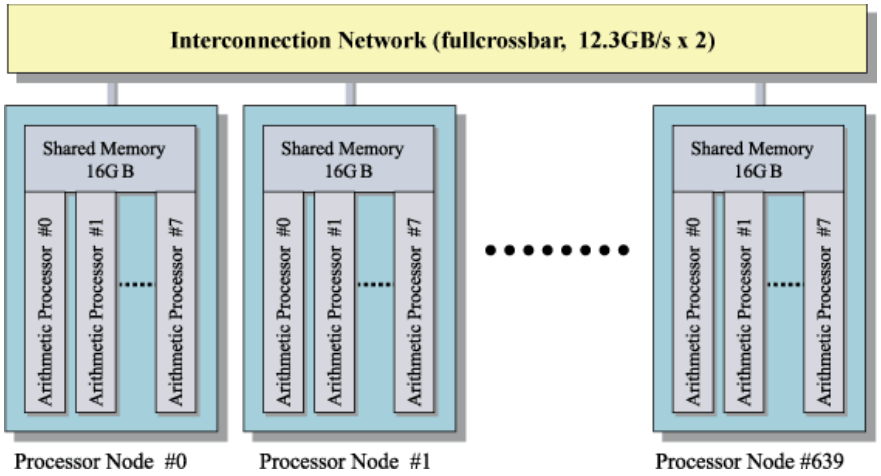
Sínrendszerek - "szerver" példa



Sínrendszerek - "játékgép" példa



Sínrendszerek - "szuperszámítógép" példa



Összefoglalás, fogalmak

- kapcsolatok, ki/bemenet, perifériák
- portok, eszközvezérlő
- sínrendszer, sínrendszer-architektúra
- sínfoglalás, soros kiszolgálás, lekérdezéses kiszolgálás, párhuzamos kiszolgálás
- master/slave sínhasználat

Számítógépek architektúrája

9. előadás

Istenes Zoltán

Eötvös Loránd Tudományegyetem
Informatikai Kar
Programozáselmélet és Szoftvertechnológiai Tanszék

Programtervező Informatikus BSc - B szakirány
2007 / Budapest

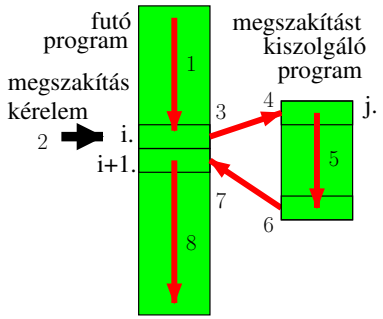
- "Váratlan események"...
- megszakítás (interrupt)
- megszakítási rendszer
- megszakítási kérelem (IRQ - Interrupt Request)
- megszakítási kérelem kiszolgálása

Cél:

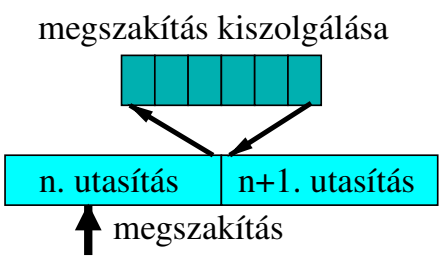
- események gyors kezelése...
- minél kevésbé zavarja a feldolgozást...

Megszakítás folyamata

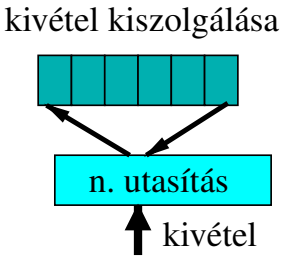
1. "futó" program
2. megszakítás
3. a "futó" program megszakítása
4. "kiszolgáló" program indítása
5. "kiszolgáló" program lefutása
6. "kiszolgáló" program befejezése
7. "visszatérés" a megszakított programhoz
8. a "futó" program folytatása



Megszakítás és kivétel



- hardver működés
- perifériára várakozás
- asszinkron esemény
  - várható "kért" működés
  - váratlan "hiba"



- programműködés
- szinkron esemény
  - hibás utasítás
  - 0-val osztás
  - túlcsordulás
  - laphiba

Megszakítás-kezelés folyamata

1. megszakítás engedélyezése
   - maszkolás
   - prioritás
   - megszakítható pont
2. megszakítás analízisa (kiszolgáló rutin?)
   - kód
   - tárcím
   - utasítás
3. állapotmentés
4. kiszolgálás
5. állapot visszaállítása

Megszakítás engedélyezése - maszkolás, prioritás

Maszkolás:

- bizonyos megszakítások "figyelmen kívül hagyása"
- megszakítási kérelmek engedélyezése / tiltása
- maszkolható / nem maszkolható kérelmek (NMI)

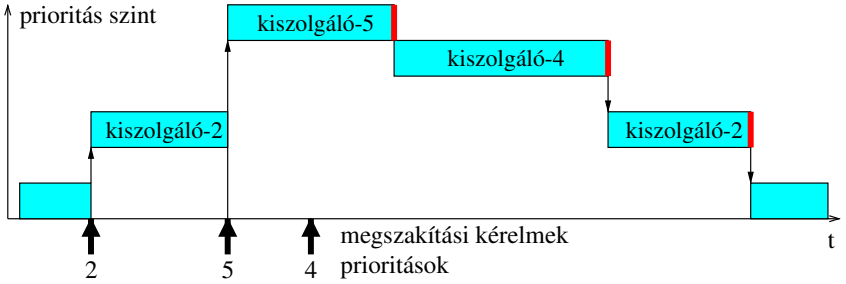
Prioritás:

- prioritások kezelése:
  több megszakítás egyidőben?
- többszörös megszakításkezelés:
  újabb megszakítás a megszakítás kezelése közben?

- megszakítható pont...

Többszörös megszakítás

- egyszintű megszakítási rendszer:
  a kiszolgáló rutin nem megszakítható
- többszintű megszakítási rendszer:
  prioritási szintnek megfelelően



Megszakítás-analízis - a megszakítást kezdeményező eszköz

Szoftver módszer

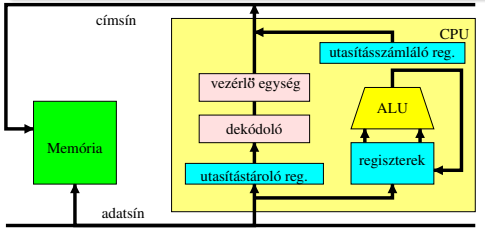
- operációs rendszer időnként megvizsgálja az eszközöket "lekérdezéses megszakításkezelés" (polling interrupt)

Hardver módszer

- 1 megszakítási vonal...
- több megszakítási vonal...
- vektoros módszer: kiszolgáló rutin kezdőcíme...

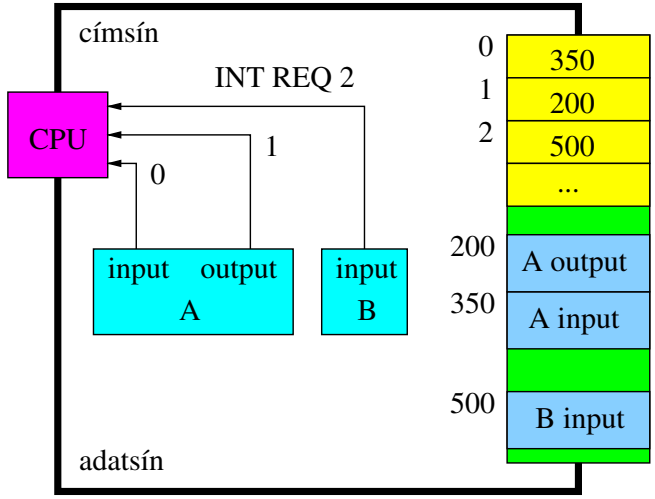
Megszakítás-analízis - rutin kezdőcím meghatározás

A megszakítást kezdeményező eszköz kiszolgáló rutinjának a kezdőcímének a meghatározása.



- kiszolgáló rutint elindító utasítás átadása
- kiszolgáló rutint elindító utasítás címének az átadása
- kiszolgáló rutin címének az átadása
- sorszám átadása: megszakítási vektortáblában rutinok kezdőcímei, sorszám mint index a táblázatban

Megszakítás vektortábla



Állapotmentés - visszaállítás

"Mit kell elmenteni? Kinek, hogyan? Hova?"

Mit:

- regiszterek
- PC
- PSW

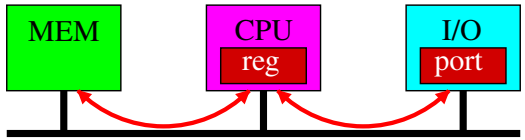
Kinek, hogyan:

- hardver
- szoftver

Hova:

- verem
- memória

Programozott I/O



- az I/O eszköz memóriához hasonló kezelése
- bájtonkénti (szavankénti) adatátvitel
- a CPU-n keresztül
  - I/O - CPU , CPU - MEM
  - MEM - CPU , CPU - I/O

Programozott I/O utasítások

- `IN X`
  bájt (szó) átvitele az `X` portról az akkumulátor-regiszterbe
- `OUT X`
  bájt (szó) átvitele az akkumulátor-regiszterből az `X` portra

Példa

státuszinformáció az 1-es port-ról olvasható,
ha kész az eszköz, az adat a 2-es port-ról olvasható

```
WAIT:
  IN 1
  CPI ready
  JNZ WAIT
  IN 2
  ...
```

1-es port-ról státuszinformáció beolvasása
az akkumulátorba összehasonlítása a
„ready” konstanssal, ha egyenlő Zflag=1
ha Zflag<>1 ugrás WAIT-re
adat beolvasása a 2-es port-ról ...

A programozott I/O tulajdonságai

- az IO átvitel sebességét attól függ, hogy a CPU milyen gyorsan tudja az I/O eszközt vizsgálni és kiszolgálni...
- a CPU (feleslegesen) sok időt „veszít” az eszköz állapotának a vizsgálatával és az adatátvitellel...
- ha több (vizsgálandó) eszköz van...
- az adat a CPU-n halad keresztül, ahelyett hogy közvetlenül a memóriába jutna...
- egyszerű, "rugalmas", "pontosan" kézben lehet tartani az adatátvitelt...

Megszakításos I/O

Probléma:

a CPU sokszor feleslegesen várakozik az IO műveletre...

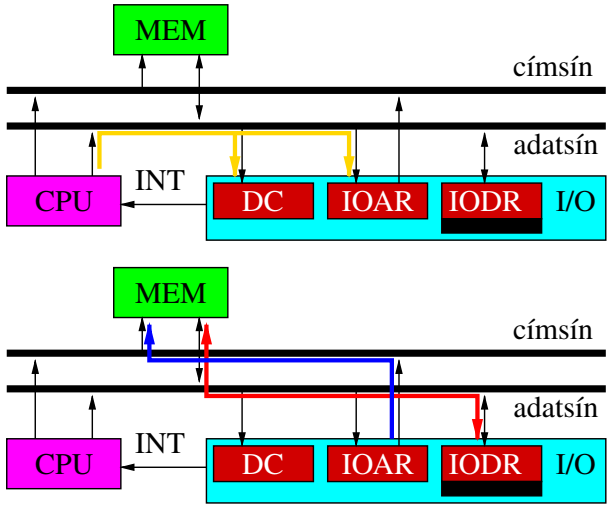
Ötlet:

az eszköz megszakítás-kérelemmel jelez ha készen van...
a CPU-nak nem kell várnia...

Tulajdonságok:

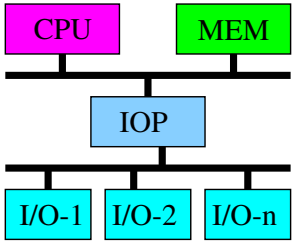
sok nagysebességű eszköznél (blokkátvitel) használhatatlan...
a megszakítások többlet terhei (overhead) miatt...

Közvetlen memória hozzáférés



I/O processzor működése

- CPU elindítja az IOP-t
- IOP utasításai a MEM-ből (CCW)
- adatáramlás
  - I/O - IOP - I/O
  - I/O - IOP - MEM
- CPU ellenőrizheti IOP-t
- IOP jelez CPU-nak ha végzett (INT)



Az IOP vezérlése

CCW - Channel Command Word: csatornavezérlő szó

Példa

Blokk írása I/O processzor-ral
IBM System 360-370 IO program példa
CCW utasítások a IOP-nak

`07h.`	`.40h.`	*szalag visszatekerése*
`37h.`	`.40h.`	*első rekord átugrása*
`01h.buffer`	`.40h.100`	*"buffer" címről 100 byte írása*
`1Fh.`	`.40h.`	*marker írása*
`07h.`	`.00h.`	*szalag visszatekerése, stop*

- multiplexer channel: egy IOP több (lassú) IO eszközt kezel
- selector channel: egy IOP egy (gyors) IO eszközt kezel

Összefoglalás, fogalmak

- megszakítás-rendszer
- megszakítás / kivétel
- megszakítási vektortábla, állapotmentés
- programozott I/O
- megszakításos I/O
- közvetlen memória hozzáférés
- I/O processzor