

**Misák Sándor**

**SZÁMÍTÓGÉPES  
ARCHITEKTÚRÁK**

**Nanoelektronikai és  
Nanotechnológiai Részleg**

**DE TTK**

v.0.1 (2007.02.20.)

## **3. előadás**

# **A SZÁMÍTÓGÉP- RENDSZEREK FELÉPÍTÉSE**

# A SZÁMÍTÓGÉPRENDSZEREK FELÉPÍTÉSE

---

## 3. előadás

### 1. Processzorok:

- **CPU** felépítése, utasítás-végrehajtás;
- **RISC** és **CISC** processzorok, **RISC** tervezési elvek;
- **Utasításszintű párhuzamosság (csővezeték, szuperskaláris architektúrák);**
- **Processzorszintű párhuzamosság.**

# A SZÁMÍTÓGÉPRENDSZEREK FELÉPÍTÉSE

---

## 3. előadás

### 2. Központi memória:

- **Bitek, memóriacímek, bájtsorrend;**
- **Hibajavító kódok;**
- **Gyorsítótár (cache-tár);**
- **Memóriatkozás és -típusok.**

# A SZÁMÍTÓGÉPRENDSZEREK FELÉPÍTÉSE

---

## 3. előadás

### 3. Háttérmemória:

- **Memóriahierarchia;**
- **Mágneslemezek;**
- **Hajlékonylemezek;**
- **IDE-lemezek, SCSI-lemezek;**
- **RAID;**
- **CD-ROM (CD-R, CD-RW, DVD, Blu-Ray).**

# A SZÁMÍTÓGÉPRENDSZEREK FELÉPÍTÉSE

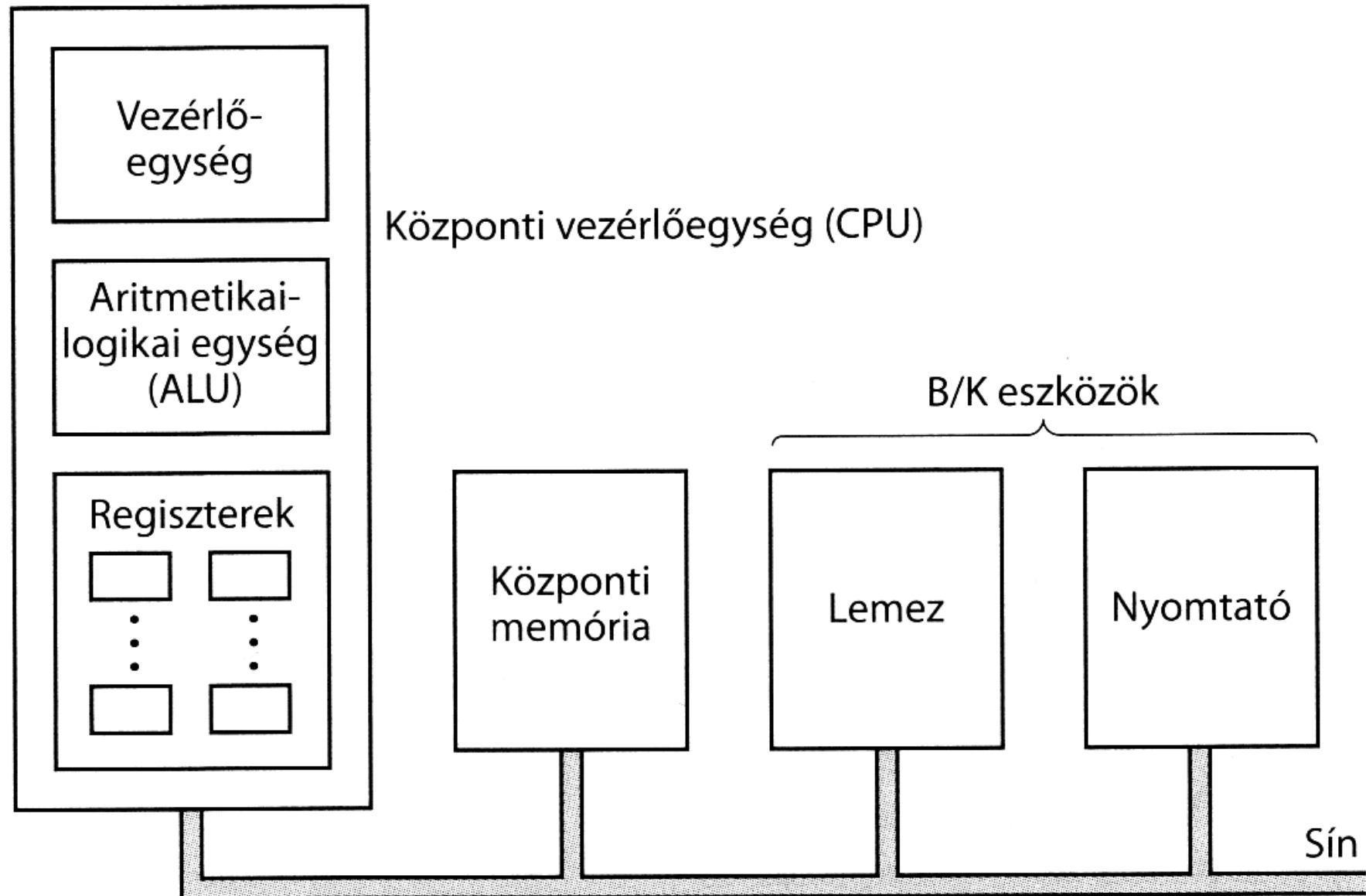
---

## 3. előadás

### 4. Bemenet / Kimenet:

- **Sínek;**
- **Terminálok (billentyűzet, monitorok, video RAM-ok);**
- **Egér, nyomtatók, digitális kamerák;**
- **Telekommunikációs berendezések (modemek, digitális előfizetői vonalak, kábeles Internet);**
- **Karakterkódok.**

# EGYSZERŰ SÍNALAPÚ SZÁMÍTÓGÉP



# EGYSZERŰ SÍNALAPÚ SZÁMÍTÓGÉP FELÉPÍTÉSE

---

## Központi feldolgozó egység (CPU):

Feladata a központi memóriában tárolt **programok végrehajtása**, azaz a program utasításainak egymás utáni beolvasása, értelmezése és végrehajtása. Különálló részekből áll:

- **Vezérlőegység (CU)**;
- **Aritmetikai-logikai egység (ALU)**;
- **Regiszttertár** (általános célú és speciális regiszterek).



# EGYSZERŰ SÍNALAPÚ SZÁMÍTÓGÉP FELÉPÍTÉSE

---

## Vezérlőegység (CU):

A **CU** feladata az **utasítások beolvasása** a központi memóriából és az **utasítások típusának megállapítása**.

## Aritmetikai-logikai egység (ALU):

Az **ALU** a program utasításainak végrehajtásához szükséges **műveleteket végez**.

## Regisztartár:

Kisméretű, gyors memória, amelyben **részeredmények** és **bizonyos** (processzorállapot, vezérlési) **információk tárolódnak**.

# EGYSZERŰ SÍNALAPÚ SZÁMÍTÓGÉP FELÉPÍTÉSE

---

## Központi memória:

A tárolóban **található** a végrehajtás alatt lévő **program** és a feldolgozásban felhasznált **adatok** is.

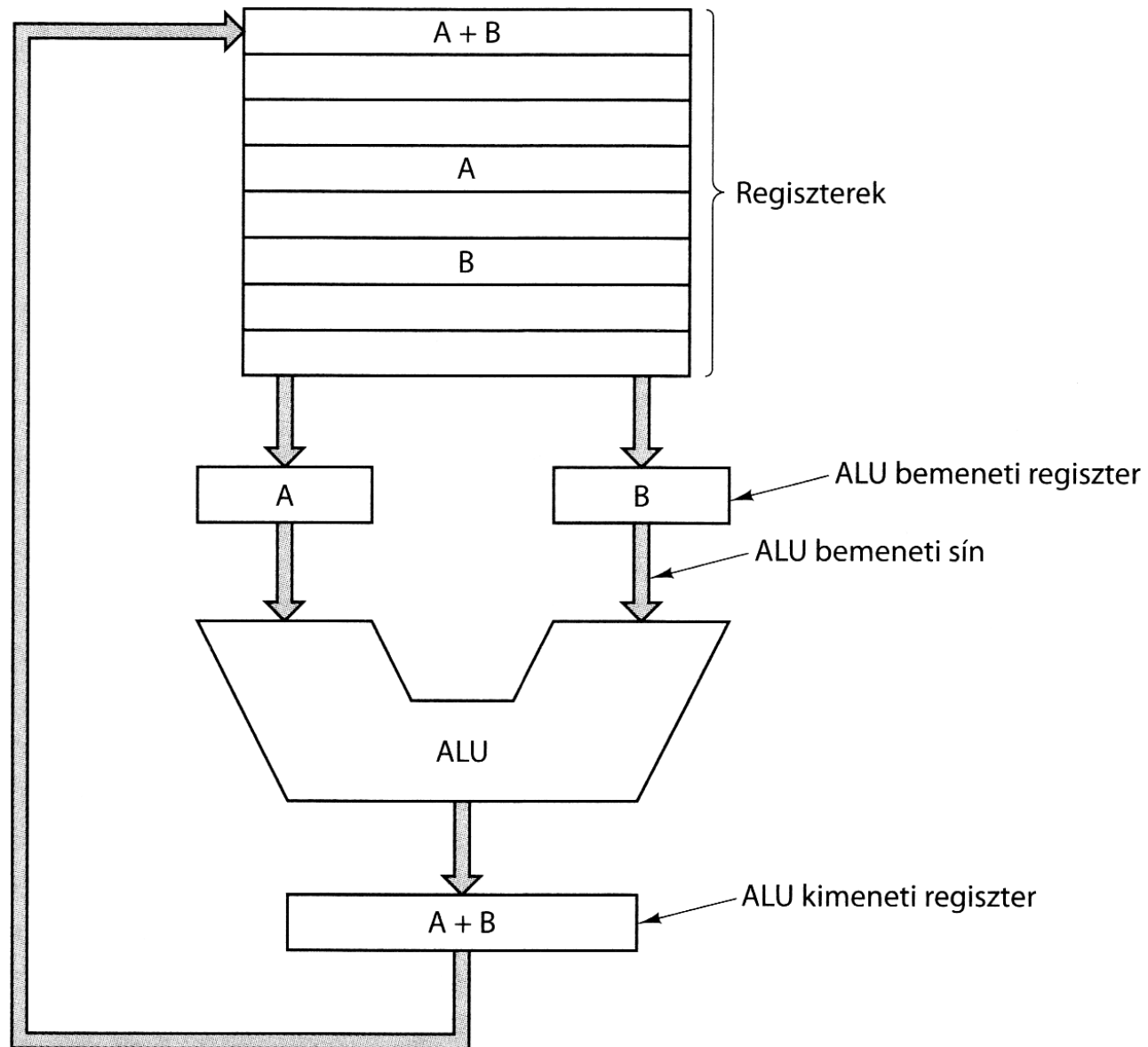
## Sín (busz):

A számítógép egyes részeit köti össze, amely **címek, adatok és vezérlő jelek** továbbítására szolgáló **vezetékkegység**.

A **CPU**-t tekintve a sín lehet:

- **belső** (a **CPU** belső egységei között; **CPU** ↔ társprocesszor);
- **külső** (**CPU** ↔ központi memória, **CPU** ↔ **Be/Ki** egységek).

# NEUMANN-ELVŰ SZÁMÍTÓGÉP ADATÚTJA



# NEUMANN-ELVŰ SZÁMÍTÓGÉP ADATÚTJA

---

Egy tipikus Neumann-elvű számítógép egyik része láttuk, az ún. **adatút (data path)**, melynek részei a **regiszterek** (általában **1÷32**), az **ALU** és az ezeket összekötő néhány **sín**.

A legtöbb **utasítás két kategóriába** sorolható:

- **Regiszter-memória utasítások** (segítségükkel tölthetünk át szavakat a memóriából a regiszterekbe, valamint a regiszterek tartalmát írhatjuk vissza a memóriába);
- **Regiszter-regiszter utasítások** (ld. az előző ábra regiszterek és **ALU** közötti adatmozgást).

# NEUMANN-ELVŰ SZÁMÍTÓGÉP ADATÚTJA

---

A két operandusnak az **ALU**-n történő átfutásából és az eredmény regiszterbe tárolásából álló folyamatot **adatútciklusnak** nevezzük.

Ez a legtöbb **CPU lelke**. Jelentős mértékben ez határozza meg, hogy a gép mire képes.

Minél **gyorsabb** az **adatútciklus**, annál **gyorsabban** dolgozik a **gép**.

# UTASÍTÁS-VÉGREHAJTÁS

---

- 1.** A soron következő utasítás beolvasása a memóriából az utasításregiszterbe.
- 2.** Az utasításslámláló regiszter beállítása a következő utasítás címére.
- 3.** A beolvasott utasítás típusának meghatározása.
- 4.** Ha az utasítás memóriabeli szót használ, a szó helyének megállapítása.
- 5.** Ha szükséges, a szó beolvasása a CPU regiszterébe.
- 6.** Az utasítás végrehajtása.
- 7.** Vissza az **1.** pontra, a következő utasítás megkezdése.

# UTASÍTÁS-VÉGREHAJTÁS

---

**Az utasítás-végrehajtás lépéssorozatát betöltő-dekódoló-végrehajtó ciklusnak nevezzük.**

**Központi szerepe van minden számítógép működésében.**

**Mikroarchitektúra-szinten az utasítások végrehajtása kétféleképpen történhet:**

- **Értelmezéssel (mikroprogrammal);**
- **Közvetlenül (elektronikus áramkörökkel)**

# UTASÍTÁS-VÉGREHAJTÁS

---

Az első számítógépeknek **kicsi, egyszerű utasításkészletük** volt.

Jött a felfedezés, hogy **összetettebb utasítások alkalmazása** esetén a programok **végrehajtási ideje** sok esetben **csökken**, annak ellenére, hogy az egyes utasítások végrehajtása több időt vehet igénybe.

**Összetett utasítások:**

- **Lebegőpontos utasítások;**
- **Tömbelemek közvetlen elérését lehetővé tevő gépi utasítások.**



# UTASÍTÁS-VÉGREHAJTÁS

---

**Az összetettebb utasítások előnyösebbek voltak, mert hardvermegoldásokkal több utasítást párhuzamosítva (átlapolva) lehetett végrehajtani.**

**Ezt a megoldást drágább, nagyobb teljesítményű gépeknél alkalmazták.**

**Olcsóbb gépeken az összetett utasítások megvalósítását (utasítás-kompatibilitás miatt) csak utasításinterpretálással (mikroprogramozott utasítás-végrehajtás) lehetett megoldani.**

**Az értelmezőalapú számítógépek elterjedése a gyors, csak olvasható táraknak, az ún. vezérlőtáraknak (control store) köszönhető, amelyekben az értelmezőt tárolták el.**

# UTASÍTÁS-VÉGREHAJTÁS

---

**Az interpretált utasításokkal ellátott egyszerű gépek előnyei:**

- **Olcsóbb hardveres kivitel.**
- **Hibásan implementált utasítások helyszíni javításának vagy akár az alaphardverben előforduló tervezési hibák áthidalásának lehetősége.**
- **Lehetőség új utasítások hozzáadására minimális költséggel, akár a számítógép leszállítása után is.**
- **Strukturált felépítés, amely lehetővé tette az összetett utasítások hatékony fejlesztését, tesztelését és dokumentálását.**

# RISC ÉS CISC PROCESSZOROK

---

**RISC = Reduced Instruction Set Computer**  
(csökkentett utasításkészletű számítógép);

**CISC = Complex Instruction Set Computer**  
(összetett utasításkészletű számítógép).

– **RISC** processzorok: **MIPS, SPARC.**

– **Elgondolások:**

- **kevés utasítás** (legalábbis az első **RISC** processzorokban);
- egyszerű, **szabályos felépítésű**, gyorsan végrehajtható utasítások;
- minél **több** utasítás **elindítása** (kiadása) **1 s** alatt.

# RISC TERVEZÉSI ELVEK

---

- Az összes utasítás **közvetlen hardveres végrehajtása**;
- Az **utasításkiadási (-indítási) ütem maximalizálása**;
- Az utasítások **könnyű dekódolhatósága**;
- **Csak a betöltő (LOAD) és tároló (STORE) utasítások hivatkozhatnak a memóriára (a többi utasítás csak regisztert használhat)**;
- **Nagy regisztertár ( $\geq 32$  regiszter).**

# RISC ÉS CISC PROCESSZOROK ÖSSZEHASONLÍTÁSA

Főbb jellemzők	CISC processzorok	RISC processzorok
Utasítás-végrehajtás	<p>Mikroprogramvezérelt:</p> <ul style="list-style-type: none"> <li>• igen bonyolult mikroeljárások;</li> <li>• nagy mikroprogram-tároló;</li> <li>• bonyolult, több gépi ciklus alatti műveletsor.</li> </ul>	<p>Huzalozott (hardveres):</p> <ul style="list-style-type: none"> <li>• Nincs mikroprogram;</li> <li>• Bonyolult fordítóprogram állítja elő a végső programkódot;</li> <li>• bonyolult utasítások elhagyása, egy órajel alatti utasítás-végrehajtási idő.</li> </ul>
Utasításkészlet	Sokféle (100÷300) utasítás.	Kevés (<128) utasítás.
Memóriacímzési módok	8-20	2-4



# RISC ÉS CISC PROCESSZOROK ÖSSZEHASONLÍTÁSA

Főbb jellemzők	CISC processzorok	RISC processzorok
Memóriakezelő utasítások	Sokféle, tárolót közvetlenül igénybevevő, megcímző utasítás használati lehetősége.	Memóriahasználatra csak <b>2 (LOAD és STORE)</b> utasítás áll rendelkezésre.
Utasítások szerkezete	<ul style="list-style-type: none"><li>• sok mezőből álló, bonyolult utasítások;</li><li>• <b>több utasításformátum;</b></li><li>• változó hosszúságúak.</li></ul>	<ul style="list-style-type: none"><li>• kevés mezőből álló, egyszerű utasítások;</li><li>• <b>keves utasításformátum;</b></li><li>• rögzített hosszúságúak.</li></ul>
Hardver	<ul style="list-style-type: none"><li>• bonyolult hardver;</li><li>• <b>VLSI gyártási technológia;</b></li><li>• nagy regisztertár;</li><li>• <b>pipeline-technika alkalmazása.</b></li></ul>	<ul style="list-style-type: none"><li>• egyszerűbb hardver;</li><li>• <b>VLSI gyártási technológia;</b></li><li>• nagy regisztertár;</li><li>• <b>pipeline-technika alkalmazása.</b></li></ul>

# PROCESSZORTELJESÍTMÉNY-NÖVEDELÉS

---

- **Órajel-frekvencia növelése** (a gyártási technológia korlátozza);
- **Párhuzamosítás** (több utasítás végrehajtása egyszerre):
  - **Utasításszintű párhuzamosság** (kihasználja az egyes utasításokban rejlő párhuzamosságot, több utasítás kiadása (elindítása) másodpercenként);
  - **Processzorszintű párhuzamosság** (több processzor dolgozik egyszerre ugyanazon a feladaton).

# UTASÍTÁSSZINTŰ PÁRHUZAMOSSÁG

---

- Csővezeték-technika (**pipeline-technika**);
- **Szuperskaláris processzor-felépítés.**

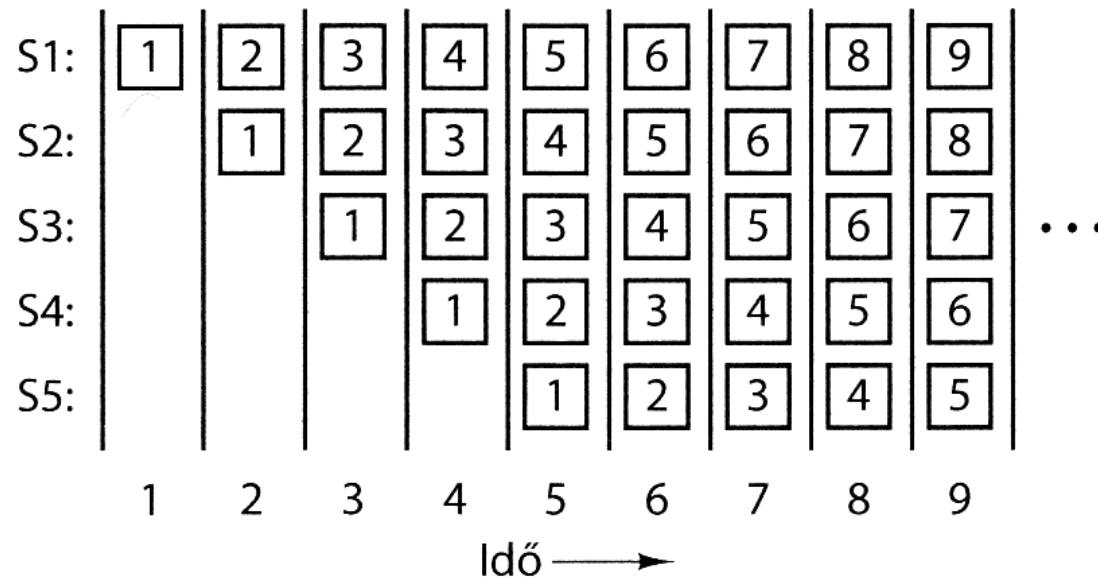
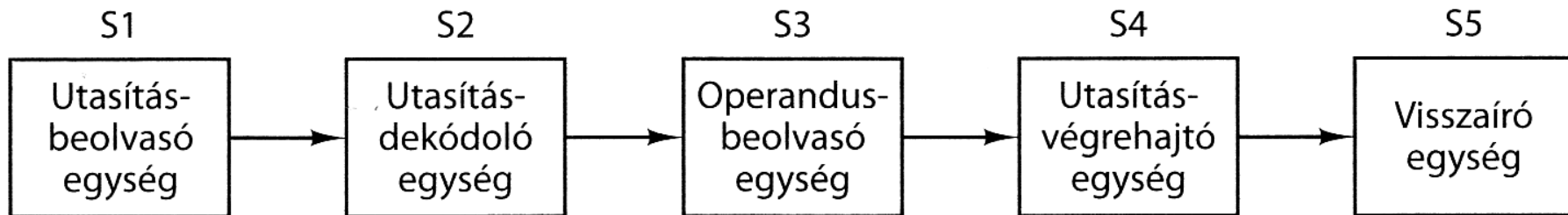
A csővezeték egységeit **fázisoknak** nevezzük.

A csővezeték lehetővé teszi, hogy kompromisszumot kössünk **késleltetés** (mennyi ideig tart egy utasítás végrehajtása) és **áteresztőképesség** (hány **MIPS** a processzor sebessége) között.

Ha az órajel **T** ns (nanosekundum), a csővezeték **n** fázisú, a késleltetés **nT** ns, mivel minden utasítás **n** állapoton halad keresztül és mindegyikben **T** ideig tartózkodik.

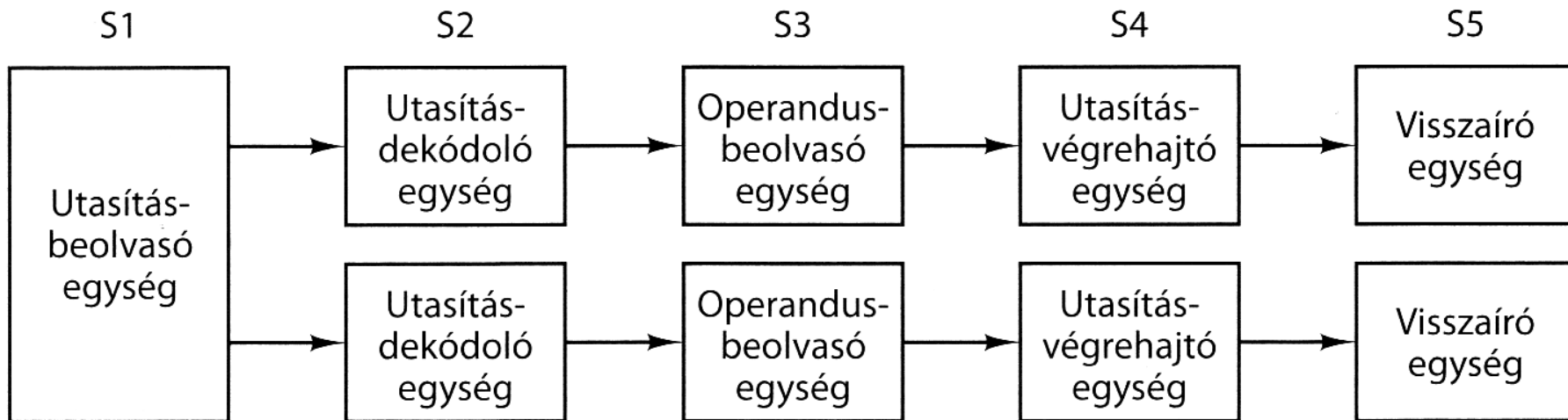


# ÖTFÁZISÚ CSŐVEZETÉK



**A fázisok állapota az idő függvényében (az ábrán kilenc órajelciklus látható)**

# SZUPERSKALÁRIS ARCHITEKTÚRÁK



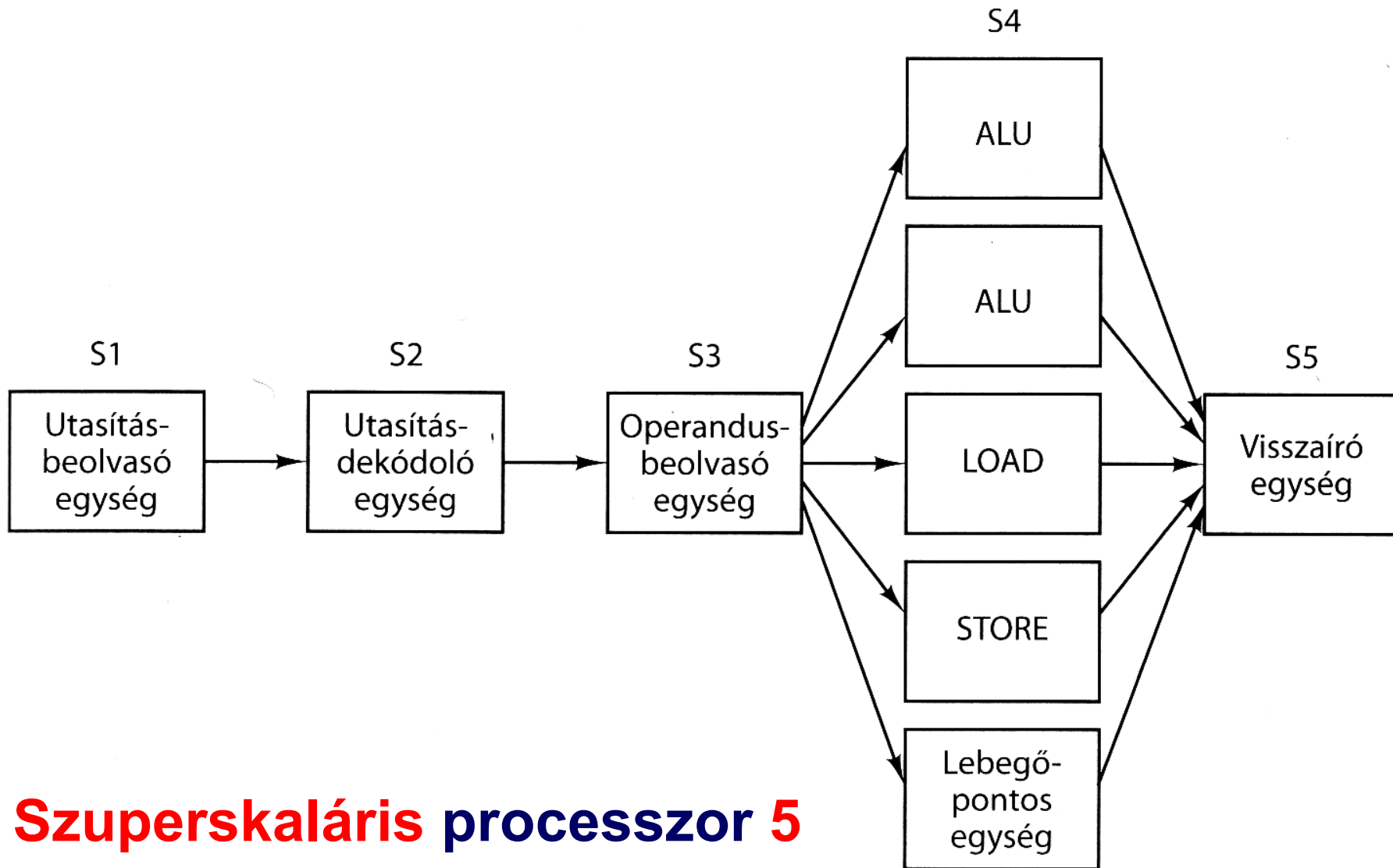
**Kettős csővezeték közös utasítás-beolvasó egységgel (pl. Intel Pentium processzor 5 fázisú, ún. U és V pipeline-ja)**

## SZUPERSKALÁRIS ARCHITEKTÚRÁK

---

- **A csővezeték számát meglehetősen növelni 4-el, de már túl sok hardverelemet kell megduplázni.**
- **Nagy teljesítményű (szuperskaláris) processzorokban csak egy csővezeték használnak, de több funkcionális egységgel.**
- **A szuperskaláris architektúra kifejezés ennek az elrendezésnek a jelölésére született.**

# SZUPERSKALÁRIS ARCHITEKTÚRÁK



**Szuperskaláris processzor 5**  
**funkcionális egységgel**

# SZUPERSKALÁRIS ARCHITEKTÚRÁK

---

- Nagyon kicsi a különbség két **CPU** között, ha az egyik **100** ns órajelenként ad ki **egy utasítást** a funkcionális egységek egy csoportja számára, a másik pedig **400** ns órajelenként **négy utasítást** ad ki ugyanennek a csoportnak.
- Mindkét esetben az az alapötlet, hogy az utasítások kiadásának sebessége **nagyobb**, mint a végrehajtás sebessége, így a **terhelés megoszlik** a funkcionális egységek között.
- Teljesítménynövelés csak abban az esetben érhető el, ha az **utasítás-előkészítési fázis lényegesen** rövidebb a végrehajtó fázisnál.

# PROCESSZORSZINTŰ PÁRHUZAMOSSÁG

---

- **Többszámítógépek:**
  - **Tömbprocesszor;**
  - **Vektorprocesszor.**
- **Multiprocesszorok.**
- **Multiszámítógépek (üzenetátadásos gépek).**
  - **Az utasításszintű párhuzamosítással 5-10-szeres processzorteljesítmény-növelés érhető el.**
  - **Viszont a processzorszintű párhuzamosítással, vagyis több CPU-t tartalmazó számítógéppel, elérhető akár az 50-100-szoros processzorteljesítmény-növelés.**

# TÖMBPROCESSZOR

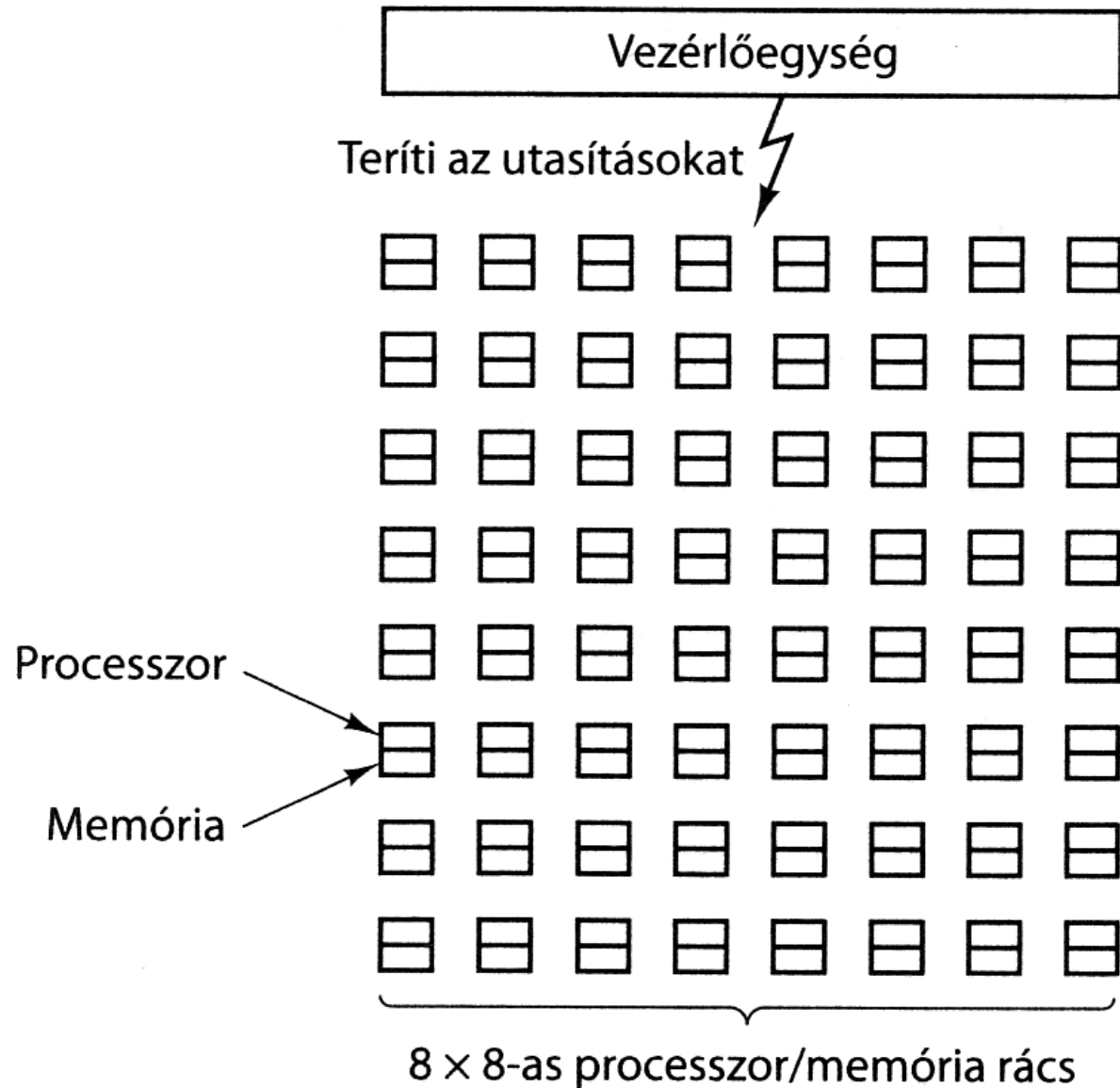
---

Egy **tömbprocesszor** nagyszámú egyforma processzorból áll, ezek ugyanazt a műveletsorozatot végzik el különböző adathalmazokon (ún. **SIMD**-processzorok).

## **ILLIAC IV** (University of Illinois):

- **4** negyedből álló gép;
- Minden negyedben **8x8**-as négyzethálóban processzor/memória párokkal.
- Negyedenként egy **vezérlőegység** adta ki az utasításokat, melyeket a hozzá tartozó processzorok **szinkronizálva** hajtottak végre.
- az adatokat mindegyik a saját memóriából vette (amit egy **inicializálási fázisban** töltöttek fel).

# ILLIAC IV TÍPUSÚ TÖMBSZÁMÍTÓGÉP





# TÖMBPROCESSZOR

---

**Tömbprocesszorokat jelenleg nem gyártanak, azonban az ötlet egyáltalán nem halt meg.**

**Az MMX és az SSE utasítások, amelyek a Pentium 4 utasításkészletében találhatóak, ezt a végrehajtási modellt használják a multimédia-szoftver felgyorsítására.**

**Ebben a tekintetben az ILLIAC IV a Pentium 4 egyik elődjének tekinthető.**

# VEKTORPROCESSZOR

---

A **tömb-** és a **vektorprocesszorok** is **adattömbökkel** dolgoznak.

Pl. két vektor elemeinek páronkénti összeadását a **tömbprocesszorok** úgy végzik, hogy a **vektor elemszámával megegyező számú összeadóegységet** tartalmaznak.

A **vektorprocesszorok** viszont ezt a műveletet úgy végzik, hogy **vektorregisztereket (regiszterláncot)** alkalmaznak.

Egy **vektorregiszter** több hagyományos regiszterből áll, ezeket a betöltő utasítás után, sorosan tölti fel a memóriából.

# VEKTORPROCESSZOR

---

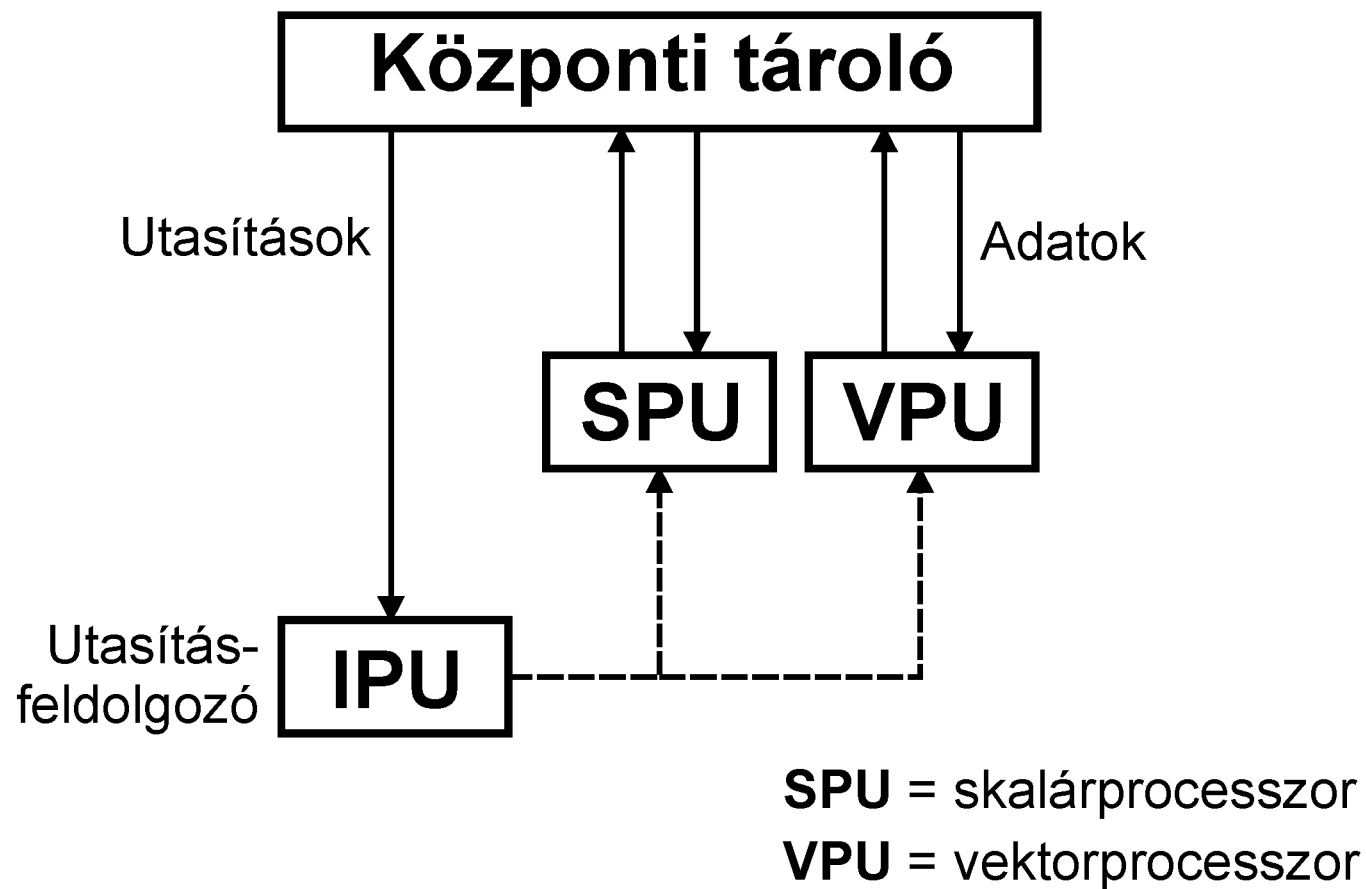
Ezután a vektorösszeadó utasítás végrehajtja két ilyen vektor elemeinek páronkénti összeadását úgy, hogy egy **csővezetékes összeadóba** irányítja a párokat a két vektorregiszterből.

A vektorösszeadás eredménye egy újabb vektor, amelyet egy vektorregiszterbe lehet tárolni, vagy közvetlenül fel lehet használni egy újabb vektorművelet operandusaként.

Skalár műveletek végrehajtásához a **vektorszámítógépben** külön egy skalárprocesszort alakítottak ki.

Első ilyen számítógép a **Cray-1 (1974)**.

# VEKTORSZÁMÍTÓGÉP



# MULTIPROCESSZOR

---

Egy **tömbprocesszor** feldolgozóegységei nem függetlenek egymástól, mert mindegyiküket **egy közös vezérlőegység** felügyeli.

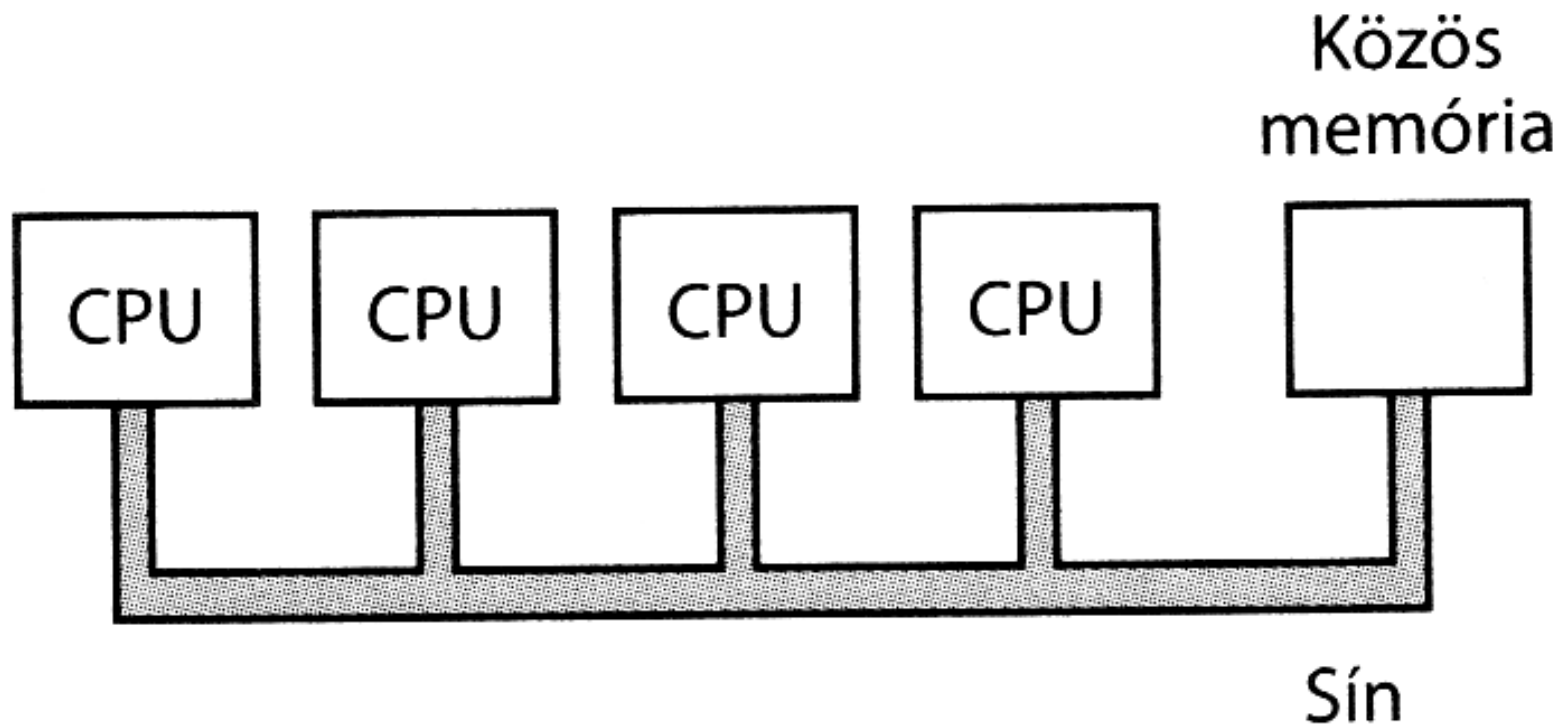
A **multiprocesszor** olyan rendszer, amelyben közös memóriát használó egynél több **CPU** található.

Mivel mindegyik **CPU** írhatja és olvashatja a memória bármely részét, **együtt kell működniük (szoftveresen)**, hogy ne legyenek egymás útjában.

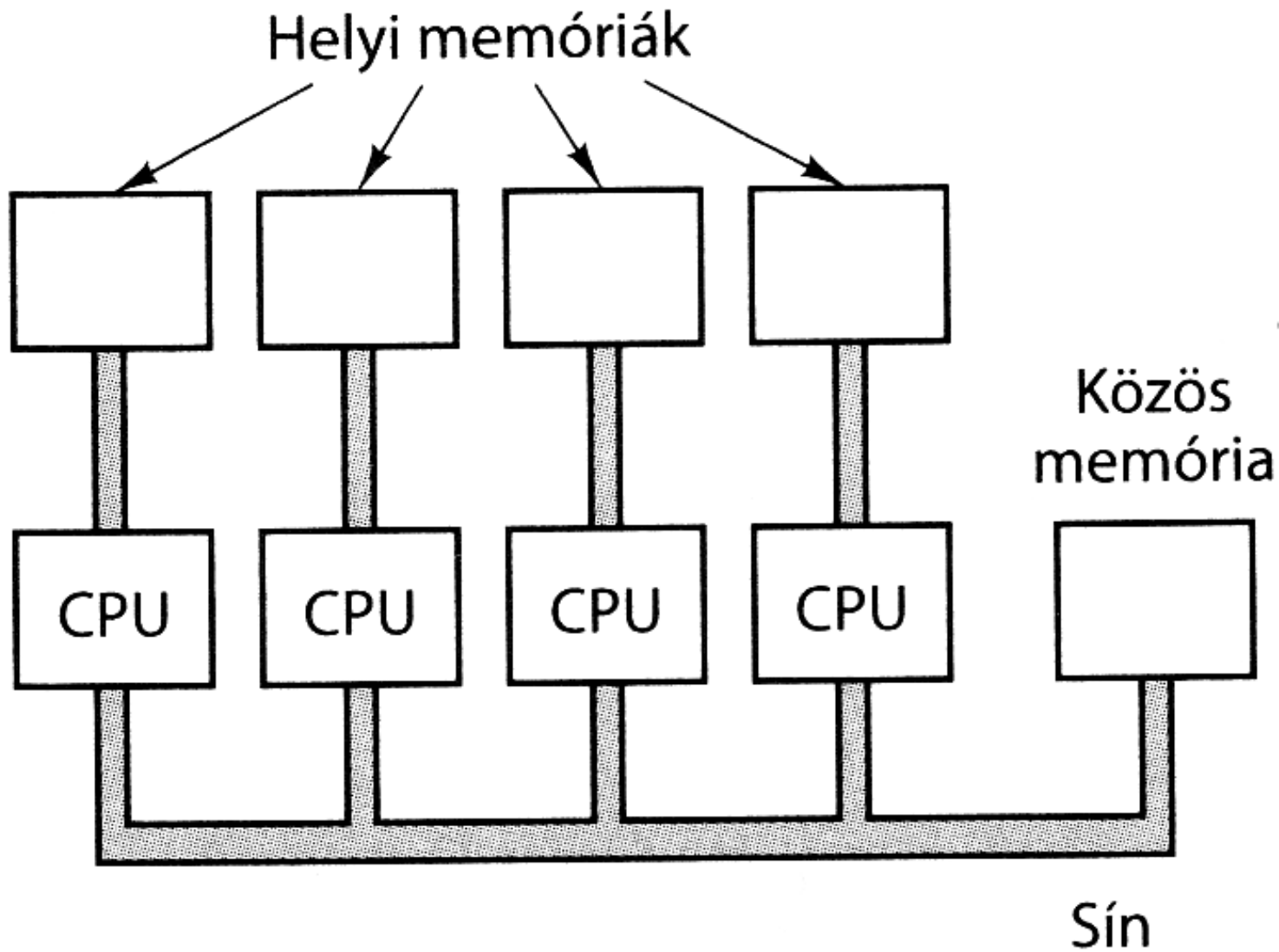
Amikor **két** vagy **több CPU** rendelkezik azzal a képességgel, hogy szorosan együttműködjenek, mint ahogyan a multiprocesszorok esetében, akkor azokat **szorosan kapcsoltaknak** nevezik.

# EGYSÍNES MULTIPROCESSZOR

---



# MULTIPROCESSZOR LOKÁLIS MEMÓRIÁKKAL



# MULTISZÁMÍTÓGÉP

---

Habár kevés ( $\leq 256$ ) processzorból álló multiprocesszorok aránylag könnyen építhetők, nagyokat meglepően nehéz konstruálni.

A nehézséget az összes processzor és a memória összekötése jelenti.

A sok összekapcsolt számítógépből álló rendszereket, amelyeknek **csak saját** memóriájuk van és **közös** memóriájuk **nincs**, **multiszámítógépeknek** nevezik.

Akár több **10000** processzort tartalmazó mutiszámítógépet is építettek már.

A **multiszámítógép** processzorai **üzenetek küldésével** kommunikálnak egymással.



# MULTISZÁMÍTÓGÉP

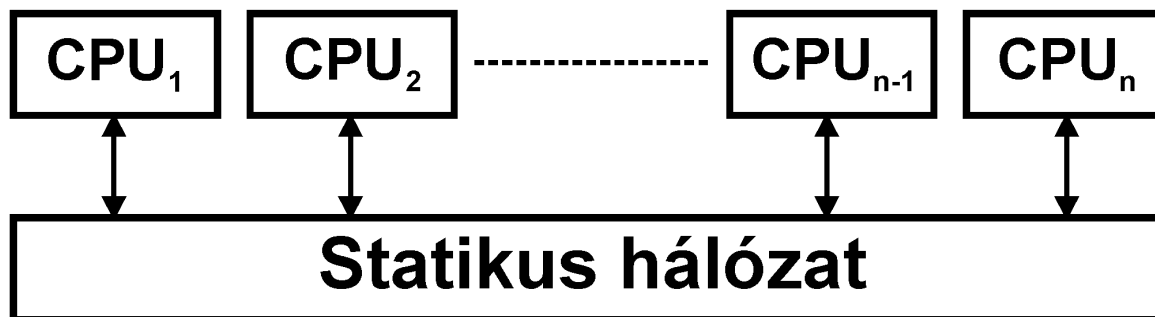
A **multiszámítógépeket 2, 3 dimenziós rácsba, fákba, gyűrűkbe** kötik (nem kötik össze mindegyiket egymáshoz).

Ennek következtében egy gép valamelyik másikhöz küldött üzeneteinek gyakran egy vagy több közbenső gépen vagy csomóponton kell áthaladniuk ahhoz, hogy a kiindulási helyükről elérjenek a céljukhoz.

Mindazonáltal **néhány mikrosekundumos nagyságrendű üzenetküldési idők** nagyobb nehézség nélkül elérhetők.

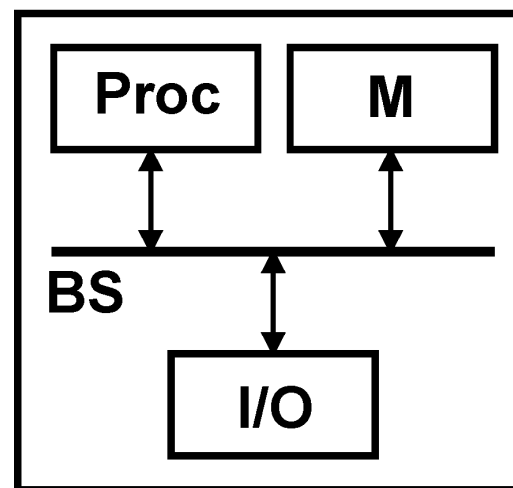
A multiszámítógépek **CPU**-it időnként **lazán kapcsoltaknak** nevezik, megkülönböztetve őket a multiprocesszorokban található szorosan kapcsolt **CPU**-któl.

# MULTISZÁMÍTÓGÉP



CPU<sub>i</sub> elem

Proc = processzor  
M = memória  
I/O = I/O egység  
BS = belső sín



**Multiprocesszoros architektúra statikus  
kapcsolati rendszerrel**