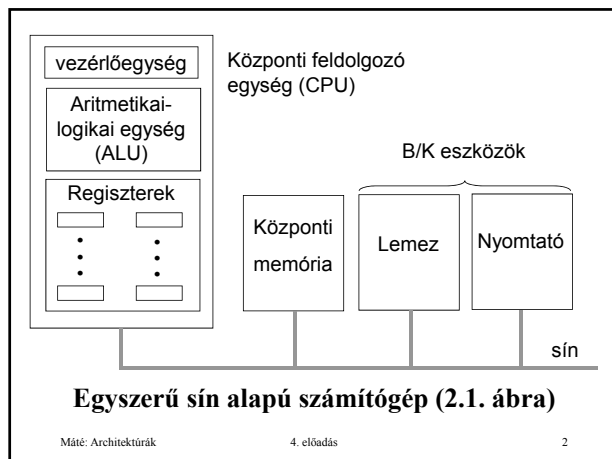


**Október 19, 20, 21, 22-én teszt
az Irinyi 227-es teremben
a MOODLE vizsgáztató programmal
az október 19-e előtt
elhangzott előadások anyagából.**

**A vizsgáztató program az október 12-ével kezdődő
héten kipróbálható, gyakorolható lesz.**

**További információt a honlapomon, az előadás
mellékletek között fogok adni.**

Máté: Architektúrák 4. előadás 1



CPU feladata: a memóriában tárolt program végrehajtása.

vezérlőegység

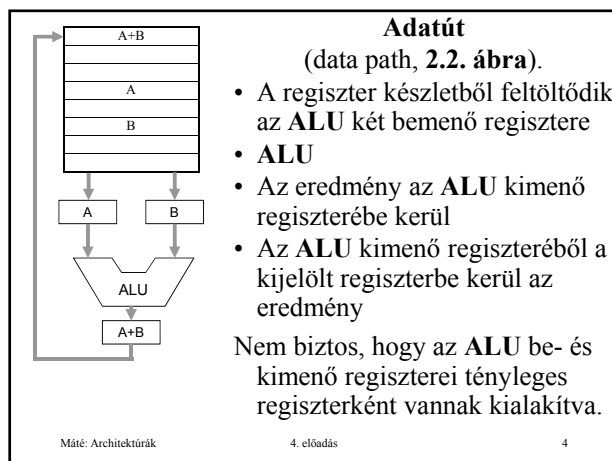
Aritmetikai-logikai egység (ALU)

Regiszterek

Részei:

- vezérlőegység, feladata:
 - a program utasításainak beolvasása,
 - az ALU, a regiszterek vezérlése,
- aritmetikai-logikai egység (ALU), feladata: az utasítások végrehajtása,
- regiszter készlet, feladata: részeredmények, vezérlő információk tárolása. A legfontosabb regiszterek:
 - utasítászámoló (Program Counter): **PC**,
 - utasításregiszter (Instruction Register): **IR**,
- adatút (data path, **2.2. ábra**).

Máté: Architektúrák 4. előadás 3



CPU (Central Processing Unit) feladatai

- a végrehajtható utasítás betöltése,
- a betöltött utasítás típusának megállapítása,
- az ezt követő utasítás címének megállapítása,
- ha kell, az operandus(ok) helyének megállapítása,
- ha kell, az operandus(ok) betöltése,
- az utasítás végrehajtása,
- ha kell, az eredmény helyének megállapítása,
- ha kell, az eredmény tárolása,
- az egész ciklus újra kezdése.

Máté: Architektúrák 4. előadás 5

RISC – CISC

RISC: Reduced Instruction Set Computer
csökkentett utasításkészletű számítógép

CISC: Complex Instruction Set Computer
összetett utasításkészletű számítógép

A 70-es években nagyon sok bonyolult utasítást építettek a gépekbe, mert a **ROM**-oknak a **RAM**-okhoz viszonyított nagy sebessége a mikroprogram gyors lefutását – a bonyolult utasítás viszonylag gyors végrehajtását eredményezte → **CISC**.
Nem volt ritka a 200-300 utasítással rendelkező gép.

Máté: Architektúrák 4. előadás 6

A RISC kialakulása

IBM-801 (John Cocke) Seymour Cray ötletei alapján nagy teljesítményű miniszámítógép. Nem került piacra, csak 1982-ben publikálták.

Berkeley 1980 (David Pettersen, Carlo Séquin) **RISC I**, később **RISC II** → **SPARC**

Stanford 1981 (John Hennessy) **MIPS**

Elv: Csak olyan utasítások legyenek, amelyek az adatút egyszeri bejárásával végrehajthatók.

Tipikusan kb. 50 utasításuk van.

Ha egy **CISC** utasítás 4-5 **RISC** utasítással helyettesíthető, és a **RISC** 10-szer gyorsabb, akkor is a **RISC** nyer.

Időközben a **RAM**-ok sebessége csaknem elérte a **ROM**-ok sebességét, ez is a **RISC** mellett szól.

KOMPATIBILITÁS

Az **Intel** túlélte: a **486**-os processzortól kezdődően minden processzora tartalmaz **RISC** magot, amely a legegyszerűbb, és egyben leggyakoribb utasításokat egyetlen adatút ciklus alatt hajtja végre, csak a többi – a ritkábban előfordulókat – interpretálja a **CISC** elvnek megfelelően → versenyképes maradt.

Korszerű számítógépek (RISC) tervezési elvei

- Minden utasítást közvetlenül a hardver hajtson végre
- Maximalizálni az utasítások kiadásának ütemét
- Az utasítások könnyen dekódolhatók legyenek
- Csak a betöltő és tároló utasítások hivatkozzanak a memóriára

→ Sok (legalább 32) regiszter kell

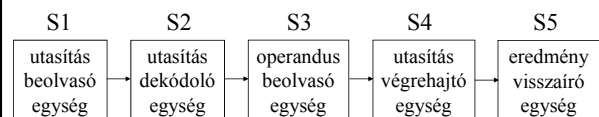
Párhuzamosítás: utasítás vagy processzor szintű.

Utasítás szintű: szállítószalag, csővezeték (pipelining).

Kezdetben:



Minden fázist külön hardver hajt végre (**2.4. ábra**), ezek párhuzamosan működhetnek (szerelő csarnok).

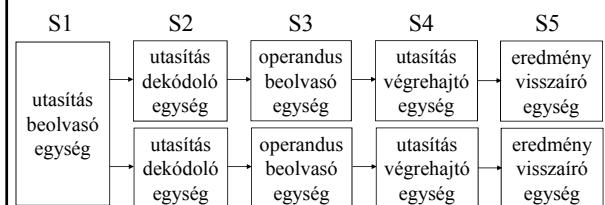


	A végrehajtás alatt lévő utasítás sorszáma									
S1:	1	2	3	4	5	6	7	8	9	
S2:		1	2	3	4	5	6	7	8	
S3:			1	2	3	4	5	6	7	...
S4:				1	2	3	4	5	6	
S5:					1	2	3	4	5	
idő	1	2	3	4	5	6	7	8	9	...

2.4. ábra

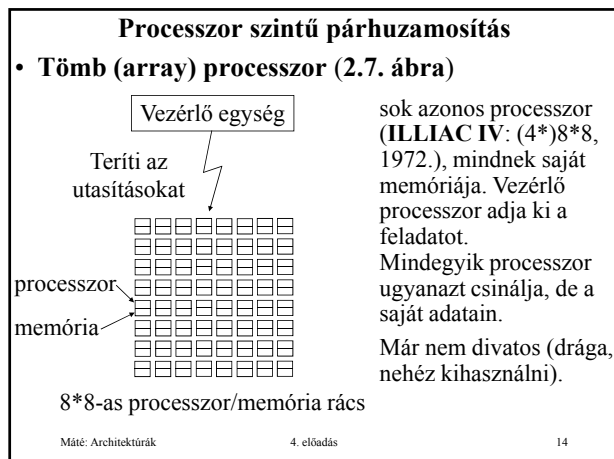
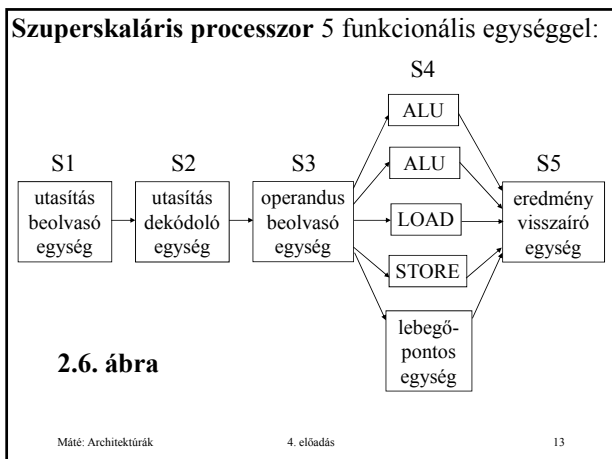
- **Késleltetés** (latency): mennyi ideig tart egy utasítás.
- **Áteresztőképesség** (processor bandwidth): hány **MIPS** (Million Instruction Per Second) a sebesség.

Több szállítószalagos CPU



Két szállítószalag (2.5. ábra):

- Két végrehajtó egység, de közös regiszterek,
- A két szállítószalag lehet különböző is (**Pentium**): fő – ez többet tud, elsőbbséget élvez – és mellék Bonyolult szabályok a párhuzamos végrehajthatóságra (fordítók vagy hardver).



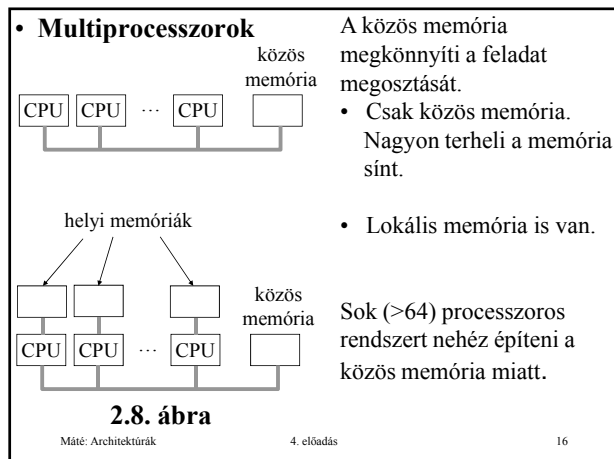
- **Vektor processzorok**

Vektor regisztereket használnak.

A vektor regiszter több hagyományos regiszterből áll. Gyors szállítószalag gondoskodik a regiszterek feltöltéséről, szintén gyors szállítószalag továbbítja a regiszterek tartamát az aritmetikai egységbe, pl. a vektor regiszterek összeadásához. Az eredmények szintén vektor regiszterbe kerülnek.

Jól kombinálhatók hagyományos processzorokkal.

Máté: Architektúrák 4. előadás 15



- **Multiszámítógépek:** Nincs közös memória: A CPU-k üzenetekkel tartják egymással a kapcsolatot. Néhány µs üzenet idő.

2-3 dimenziós hálók, fák, gyűrűk.

Közel 10 000-es rendszer is van.

Máté: Architektúrák 4. előadás 17

Adattípusok

Alapkérdés: mit támogat a hardver (milyen utasítások vannak)? Ami nincs (pl. dupla pontosságú egész aritmetika), azt szoftveresen kell megcsinálni.

Numerikus típusok:

- előjel nélküli és előjeles egész számok (**8, 16, 32, 64 bites**).
- lebegőpontos számok (**32, 64, néha 128 bites**),
- binárisan kódolt decimális számok: decimális aritmetika (**COBOL → Y2K = 2000. év probléma**).

Máté: Architektúrák 4. előadás 18

Az egyes gépek által támogatott numerikus típusok
P: Pentium 4, **U:** UltraSPARC III, **I:** I-8051

típus	1 bit	8 bit	16 bit	32 bit	64 bit	128 bit
bit	I					
előjeles egész		P U I	P U	P U	U	
előjel nélküli egész		P U	P U	P U	U	
BCD		P				
lebegőpontos				P U	P U	U

5.7-9. ábra

Máté: Architektúrák 4. előadás 19

Karakterkódolás

ASCII (American Standard Code for Information Interchanges), 7 bites: vezérlőkérekek, az angol abc kis és nagy betűi, szimbólumok, **2.43. ábra**

Latin-1 kód: 8 bites.

IS 8859: kódlap, **IS 8859-2:** magyar betűk is.

UNICODE (IS 10646), 16 bites: kódpozíciók (code point). Általában egy nyelv jelei egymás után vannak – a rendezés könnyű.

- Kínai, japán, koreai: fonetikus szimbólumok, Han ideogramok (20992 jel, nincsenek szótár szerint rendezve). ... Japán íráshoz kevés (> 50000 kanji jel van).
- Új jelek? Braille nincs benne.

Máté: Architektúrák 4. előadás 20

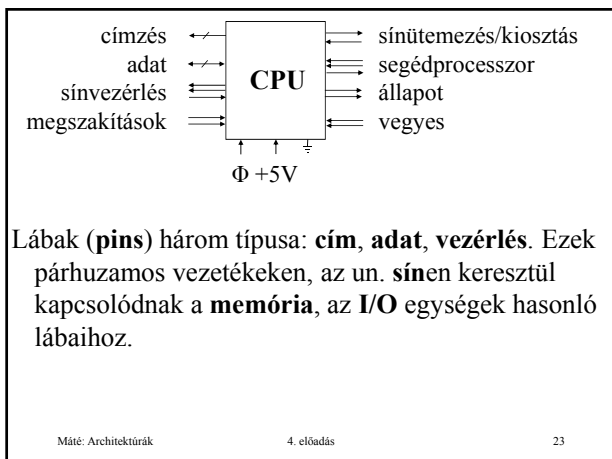
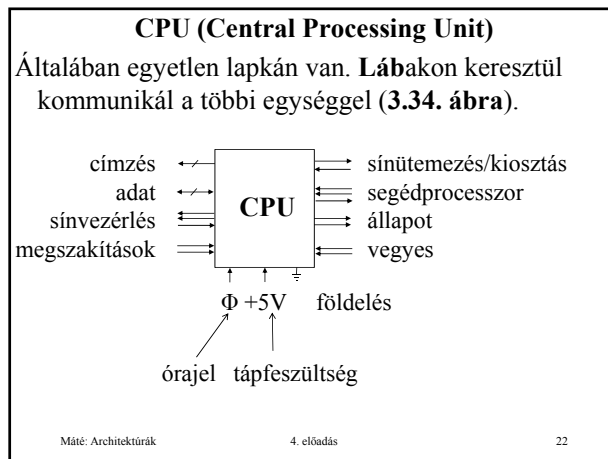
További nem numerikus típusok

Logikai érték (boolean): igaz, hamis.
 Leggyakrabban egy bájtban (szóban) ábrázolják.
 Bit térkép.

Mutató (pointer): memória cím.

Bit: kapcsolók, lámpák beállítására, lekérdezésére beágyazott rendszerekben.

Máté: Architektúrák 4. előadás 21



Lényeges a cím- és adatlábak száma (**3.34. ábra**):

- Ha m címláb van, akkor 2^m memóriarekesz érhető el (tipikus $m = 16, 20, 32, 64$).
- Ha n adatláb van, akkor egyszerre n bit olvasható illetve írható (tipikus $n = 8, 16, 32, 36, 64$).

Máté: Architektúrák 4. előadás 24

Óra, áram (3.3 v. 5V), föld, továbbá **vezérlőlábak**:

- sín vezérlés (bus control): mit csináljon a sín,
- megszakítások,
- sín kiosztás (ütemezés, egyeztetés – bus arbitration): kinek dolgozzon a sín,
- segéd processzor vezérlése, jelzései,
- állapot,
- egyéb.

Máté: Architektúrák 4. előadás 25

Pl. utasítás betöltése:

- A CPU kéri a sín használat jogát,
- Az utasítás címét a cím lábakra teszi,
- vezérlő vonalon informálja a memóriát, hogy olvasni szeretne,
- a memória a kért szót az adat vonalakra teszi, kész jelzést tesz egy vezérlő vonalra,
- a CPU végrehajtáshoz átveszi az utasítást.

Máté: Architektúrák 4. előadás 26

Sín (bus): Korai személyi számítógépeknél egyetlen (külső) rendszersín, manapság legalább kettő van: egy belső és egy külső (I/O), **3.35. ábra.**

Máté: Architektúrák 4. előadás 27

Sínprotokoll: a sín működésének + a csatlakozások mechanikai, elektronikus definíciója

Mesterek (masters): aktív (kezdeményező) berendezések (CPU, lemez vezérlő).

Szolgák (slaves): passzív (végrehajtó) berendezések (lemez vezérlő, CPU), **3.35. ábra.**

Ez a szereposztás tranzakciónként eltérő lehet.

Mester	Szolga	példa
CPU	Segéd proc.	CPU felkínálja az utasítást
Segéd proc.	CPU	Segéd proc. kéri az operandusokat

A memória sohasem lehet mester!

Máté: Architektúrák 4. előadás 28

A sínhez kapcsolódó lapkák lényegében erősítők.

Mester – **sín vezérlő (bus driver)** – sín.

Sín – **sín vevő (bus receiver)** – szolga.

Mester–szolgáknál: **sín adó-vevő (bus transceiver)**.

A csatlakozás gyakran **tri-state device** vagy **open collector** – **wired-OR** segítségével történik.

Sávszélesség: (továbbítható bitek száma) / sec.

Sávszélesség növelése:

Gyorsítás: probléma a sín aszimmetria (skew), kompatibilitás.

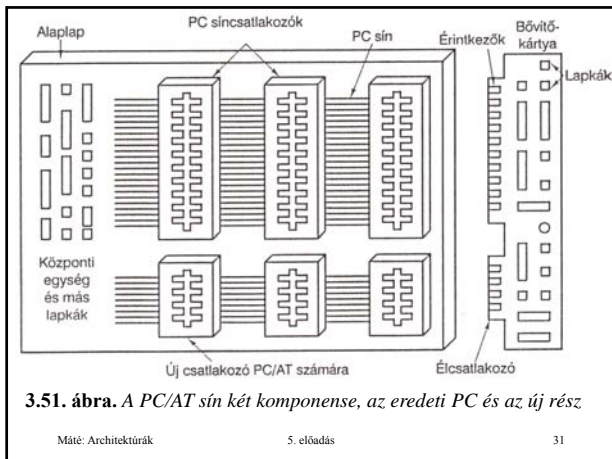
Sínszélesség: több vezeték → drágább, kompatibilitás.

Máté: Architektúrák 4. előadás 29

Sínszélesség (pl. IBM PC: 3.37., 3.51. ábra).

3.37. ábra. A cím szélességének növekedése az elmúlt időszakban

Máté: Architektúrák 4. előadás 30



Alaplap (motherboard, parentboard, **3.51. ábra**)
 Rajta van a CPU, sín(ek), ezen illesztő helyek (slots) a memória és a **beviteli/kiviteli (Input/Output – I/O)** eszközök számára (**3.51., 2.28. ábra**).

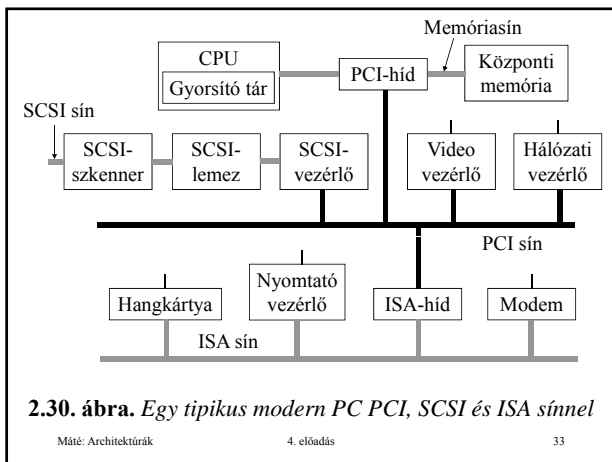
I/O eszköz: maga az eszköz + vezérlő (controller) külön kártyán vagy az alaplapon (**2.29. ábra**).

Gyorsabb CPU gyorsabb sínt igényel!

Kívánság: PC cseréjénél megmaradjon a régi perifériák egy része: az új gépben is kell a régi sín!
 Sínek szabványosítása.

Egy gépen belül több sín is használható: **2.30. ábra**.

Máté: Architektúrák 4. előadás 32



Sokszorozott (multiplexed) sín: pl. először a cím van a sínen, majd az adat (ugyanazokon a vezetékeken). Ilyenkor a sín szélessége lényegesen csökken (olcsóbb, kevesebb láb szükséges a sínhez való csatlakozáshoz), csökken a sáv szélesség is, de nem olyan mértékben. Általában bonyolultabb a sín protokoll.

Máté: Architektúrák 4. előadás 34

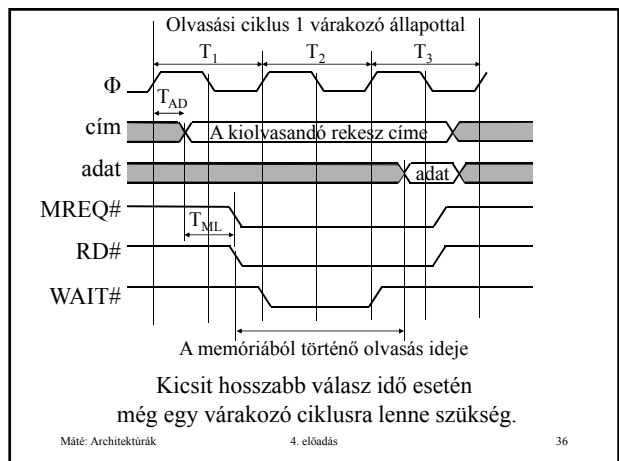
Sínek időzítése

Szinkron sín: 5 – 100 MHz-es órajel van a sín egy vezetékén. Minden síntevékenység az órajelhez van igazítva.

Síntevékenységek: cím megadása, vezérlőjelek (**MREQ#, RD#, WAIT#**), adat megérkezése, ... (**3.38. ábra**)

Jelölés	Tevékenység	min	max	idő
T_{AD}	Cím megérkezési ideje a sínen		11	ns
T_{ML}	Cím a sínen van MREQ# előtt	6		ns
...

Máté: Architektúrák 4. előadás 35



Minden sínművelet a ciklusidő (sín ciklus) egész számú többszöröséig tart:
pl. 2.1 ciklusidő helyett 3 ciklusidő kell.

A leglassabb eszközhöz kell a sín sebességét igazítani, a gyors eszköz is lassan fog működni.

Máté: Architektúrák 4. előadás 37

Aszinkron sín:

Minden eseményt egy előző esemény okoz!

Nincs órajel, **WAIT**, van viszont:

MSYN# (kérés - Master SYNchronization),
SSYN# (kész - Slave SYNchronization).

Ugyanazon a sínen gyors és lassú mester - szolga pár is lehet.

Máté: Architektúrák 4. előadás 38

Aszinkron sín működése (3.39. ábra)
Akkor indulhat újabb tranzakció, ha **SSYN#** negált.

Ugyanazon a sínen gyors és lassú mester - szolga pár is lehet.

Máté: Architektúrák 4. előadás 39

Teljes kézfogás (full handshake):

Akkor indulhat, ha **SSYN#** negált!

- Mester: kívánságok beállítása, majd **MSYN#**, vár,
- Szolga: látja **MSYN#**-t: dolgozik, majd **SSYN#**, vár,
- Mester: látja **SSYN#** -t (kész), dolgozik, ha kell, majd negálja **MSYN#** -t,
- Szolga: látja **MSYN#** negálását, negálja **SSYN#** -t.

Máté: Architektúrák 4. előadás 40

Sínütemezés (kiosztás)

Ha egyszerre többen is igénylik a sítet (CPU, I/O vezérlő), akkor a **sínütemező** (bus arbiter) dönt.

Általában I/O elsőbbséget kap (cikluslopás).

Máté: Architektúrák 4. előadás 41

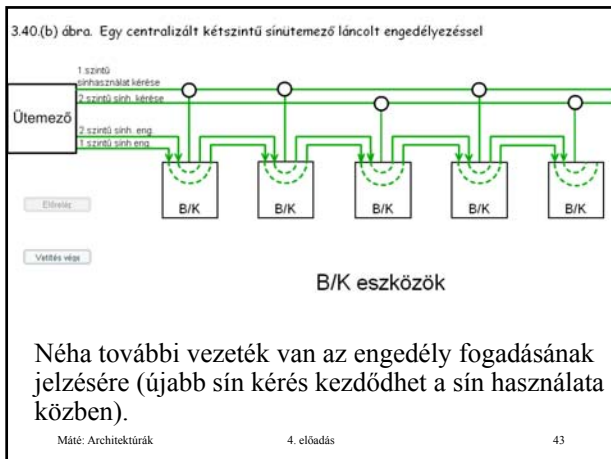
Sínütemezés (kiosztás – bus arbitration)

- **Centralizált (3.40. (a) ábra):** (margaréta) láncolás (daisy chaining), egy vagy többszintű lehet.

3.40.(a) ábra. Egy centralizált egyszintű sínütemező láncolt engedélyezéssel

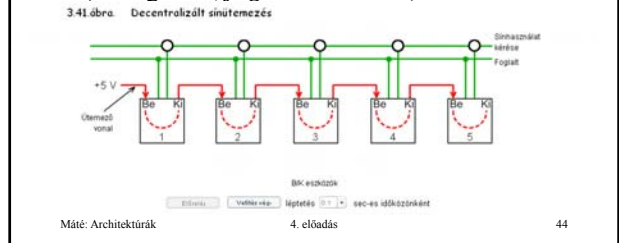
Ha van **kérés** és a **sín szabad: sín foglalási engedély**.

Máté: Architektúrák 4. előadás 42



• decentralizált

- pl. 16 prioritású: 16 eszközhez 16 kérés vonal, minden eszköz minden kérés vonalat figyel, tudja, hogy a saját kérése volt-e a legmagasabb prioritású.
- **3.41. ábra:** ha nem *foglalt* és *be*, akkor kérheti a sít (*ki* negálása, *foglalt* beállítása).



Feladatok

- Mi a CPU?
- Mi az ALU?
- Mi az adatút?
- Milyen részei vannak a CPU-nak?
- Mi a regiszter?
- Mit jelent az implicit operandus kifejezés?

Feladatok

- Mit jelent a **CPU** rövidítés?
- Hogy tartja a kapcsolatot a CPU a környezetével?
- Milyen lábai vannak egy CPU-nak?
- Miért lényeges a cím és adat lábak száma?
- Hány cím adható meg *k* címvezetéken?

Feladatok

- Mit jelent a **RISC** rövidítés?
- Mit jelent a **CISC** rövidítés?
- Mi segítette elő a **CISC** gépek kialakulását?
- Miért előnyös a **RISC** architektúra?
- Miért nem tért át az Intel **RISC** processzorok gyártására?
- Hogyan alkalmazza az Intel a **RISC** elveket?
- Melyek a modern számítógép tervezés legfontosabb elvei?
- Miért van szükség sok regiszterre a **RISC** gépeken?

Feladatok

- Milyen párhuzamosítási lehetőségeket ismer?
- Mi az utasítás szintű párhuzamosítás?
- Szemléltesse az utasítás szintű párhuzamosítást!
- Mit jelent a csővezeték (pipelining)?
- Mi a késleltetés (latency)?
- Mi az áteresztő képesség?
- A késleltetés vagy az áteresztő képesség a fontosabb a gép teljesítménye szempontjából?
- Mi az előnye/hátránya a több szállítószalagos CPU-nak?
- Mi a szuperskaláris architektúra lényege?

Feladatok

Hogy működik a tömb (array) processzor?
Mi a tömb (array) processzor előnye/hátránya?
Hogy működik a vektor processzor?
Mi a vektor processzor előnye/hátránya?
Mi a multiprocesszorok lényege?
Mi a közös/helyi memóriák szerepe a multiprocesszoros rendszerekben?
Miért nehéz sok processzoros rendszert építeni?
Mi a lényege multiszámítógépeknek?
Hogy tartják a kapcsolatot egymással a multiszámítógépek CPU-i?

Máté: Architektúrák

4. előadás

49

Feladatok

Milyen adat típusokat ismer?
Milyen karakter kódolásokat ismer?
Milyen feladatai vannak a CPU-nak?
Mi a központi memória feladata?

Máté: Architektúrák

4. előadás

50

Feladatok

Mit nevezünk sínnek?
Mit nevezünk sín vezérlésnek?
Mit nevezünk sín ütemezésnek?
Hogyan történik egy adat beolvasása a memóriából?
Hogyan történik egy adat kiírása a memóriába?
Mi a sínprotokoll?
Mi a mester, és mi a szolga?
Mit nevezünk sín vezérlőnek/vevőnek/adó-vevőnek?
Mi a sáv szélesség?

Máté: Architektúrák

4. előadás

51

Feladatok

Mi a sínszélesség?
Mi a sín aszimmetria?
Hogy növelhető egy sín sáv szélessége?
Miért nem növelhető szabadon a sín szélessége?
Miért nem növelhető szabadon a sín sáv szélessége?
Miért volt szükség a sínek szabványosítására?
Mit jelent a sokszorozott (multiplexed) sín?
Milyen hatása van a sokszorozott sín használatának?
Hogy működik a szinkron/aszinkron sín?
Mire szolgál a mester/szolga szinkronizáció?
Mi a teljes kézfogás?

Máté: Architektúrák

4. előadás

52

Feladatok

Milyen sín ütemezőket ismer?
Hogy működik a centralizált sín ütemező?
Hogy működik a decentralizált sín ütemező?

Máté: Architektúrák

4. előadás

53

Az előadáshoz kapcsolódó

Fontosabb tételek

A CPU részei, feladatai, adatút
A CISC és a RISC kialakulása
Utasítás és processzor szintű párhuzamosítás
Adat típusok, karakter kódolás
CPU, Sínek. Sín protokoll. Mester – szolga. Sín vezérlő, vevő, adóvevő. Sáv szélesség, sín szélesség.
Sokszorozott sín.
Sín időzítés: szinkron, aszinkron sín, teljes kézfogás.
Sín ütemezés.

Máté: Architektúrák

4. előadás

54