

Október 13, 14, 15, 16-án teszt
az Irinyi 216-os teremben
a MOODLE vizsgáztató programmal
az október 13-a előtt
elhangzott előadások anyagából.

A vizsgáztató program az október 6-ával kezdődő
héten kipróbálható, gyakorolható lesz.

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

Máté: Architektúrák

4. előadás

1

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

2

A RISC kialakulása

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

Berkeley 1980 (David Petterson, 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.

Máté: Architektúrák

4. előadás

3

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.

Máté: Architektúrák

4. előadás

4

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

Máté: Architektúrák

4. előadás

5

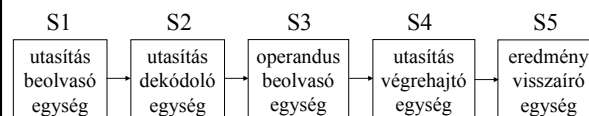
Párhuzamosítás: utasítás vagy processzor szintű.

Utasítás szintű: szállítószalag, csővezeték
(pipelining).

Kezdetben:



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



Máté: Architektúrák

4. előadás

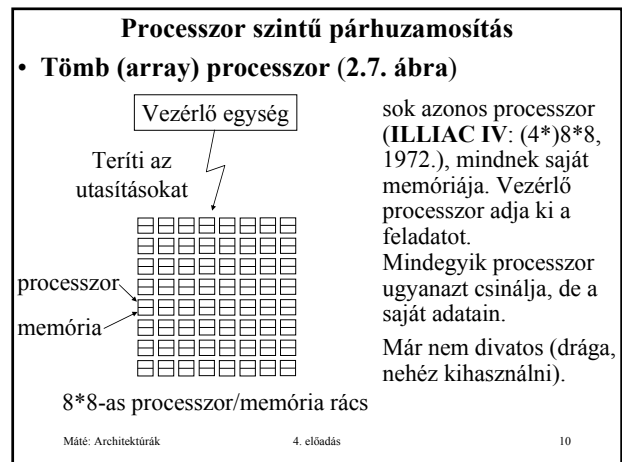
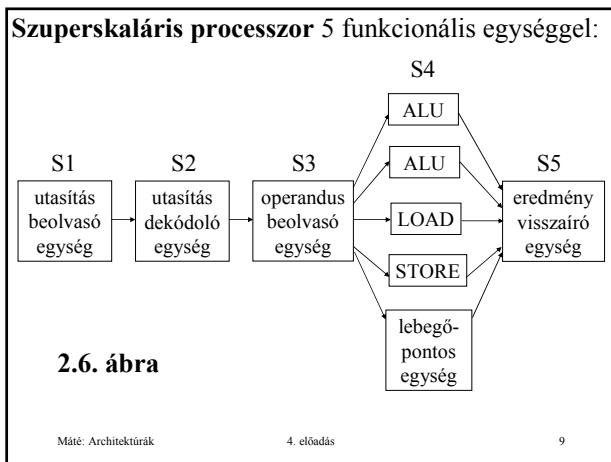
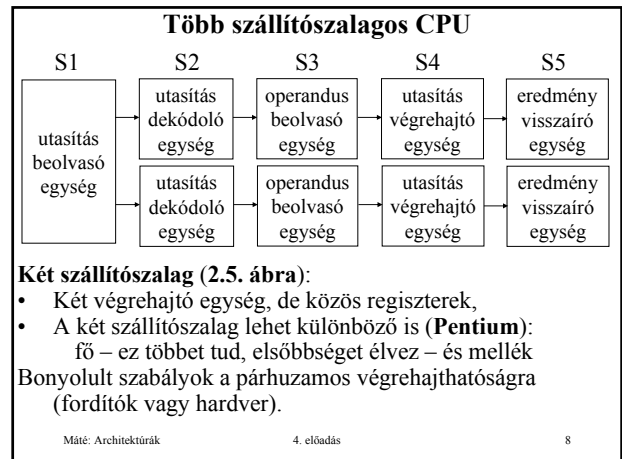
6

	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.

Máté: Architektúrák 4. előadás 7



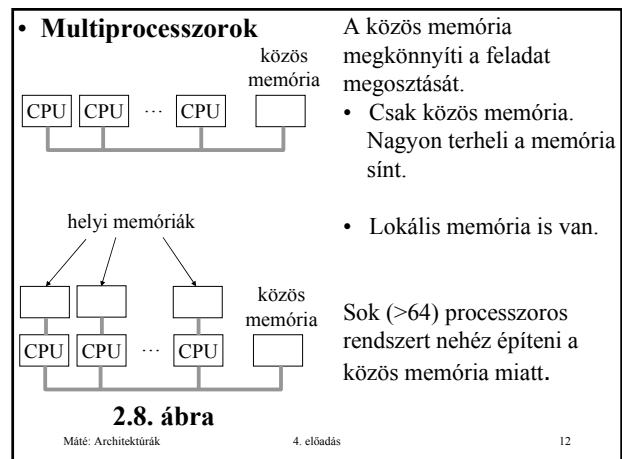
- **Vektor processzorok**

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 11



- **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 13

Központi memória (2.9. ábra)


A programok és adatok tárolására szolgál.

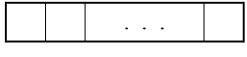
Bit: a memória alapegysége, egy 0-t vagy 1-et tartalmazhat.

Memória rekesz (cella): több bit együttese. Minden rekesz ugyanannyi bitből áll. Minden rekeszhez hozzá van rendelve egy szám, a **rekesz címe**. Egy rekeszre a címével hivatkozhatunk. A rekesz a legkisebb címezhető egység.

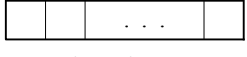
Máté: Architektúrák 4. előadás 14

Cím Rekesz/cella

0  A rekesz hossza manapság legtöbbször 8 bit (byte ~ bájtt).

1 

... n a memória cellák száma

n-1 

← Rekesz hossza →

Központi memória (2.9. ábra)

Máté: Architektúrák 4. előadás 15

A bitek száma rekeszenként néhány számítógép-történetileg érdekes, kereskedelmi forgalomba került gépen (2.10. ábra)

Számítógép	Bit
Burroughs B1700	1
IBM PC	8
DEC PDP-8	12
IBM 1130	16
DEC PDP-15	18
XDS 940	24
Electrologica X8	27
XDS Sigma 9	32
Honeywell 6180	36
CDC 3600	48
CDC Cyber	60

Máté: Architektúrák 4. előadás 16

Bájtsorrend

A legtöbb processzor több egymás utáni bájttal is tud dolgozni (**szó – word, ...**).

A legmagasabb helyértékű bájtt a szóban a legalacsonyabb címen: legmagasabb címen:

nagy (big) endian **kis (little) endian**

MSBfirst (SPARC) **LSBfirst (Pentium)**

Most/Least Significant Byte first

Ha egy 32 bites szó bájttjainak értéke rendre: a, b, c, d, akkor a szó értéke:

$$a*256^3+b*256^2+c*256+d \quad a+b*256+c*256^2+d*256^3$$

Máté: Architektúrák 4. előadás 17

Bájtsorrend (2.11. ábra)

A memória címek úgy vannak fölírva, hogy a legmagasabb helyértékű bájtt van bal oldalon.

Cím	Nagy endian				Kis endian				Cím
0	0	1	2	3	3	2	1	0	0
4	4	5	6	7	7	6	5	4	4
8	8	9	10	11	11	10	9	8	8
12	12	13	14	15	15	14	13	12	12

← 32 bites szó → ← 32 bites szó →

Máté: Architektúrák 4. előadás 18

Bájtsorrend (12. ábra)

A szövegek karaktereit mindkét esetben növekvő bájtsorrendben helyezik el

		kis endian				Cím
Cím	nagy endian					
0	0	1	2	3	3	0
	T	E	X	T	2	
4	4	5	6	7	7	4
	12	34	56	78	4	

A TEXT szöveg és az 12345678 hexadecimális szám elhelyezése a két géptípuson

Problémák a gépek közötti kommunikációban!

Máté: Architektúrák 4. előadás 19

Kódolás: adat + ellenőrző bitek = kódszó.

Két kódszó Hamming távolsága: az eltérő bitek száma.
Pl.: 11001 és 11011 (Hamming) távolsága = 1.

Hibaérzékelő kód: bármely két kódszó távolsága > 1: paritás bit.

d hibás bit javítása: a kódszavak távolsága > 2d.

Egy hibát javító kód (2.13. ábra):
m adat, r ellenőrző bit, összesen $n = m + r$.
 2^m „jó” szó, + minden „jó” szónak n db „egyhibás” szomszédja van, ezért $(1 + n)2^m \leq 2^n = 2^{m+r}$,
 2^m -mel egyszerűsítve: $m + r + 1 \leq 2^r$,
vagy másképp: $m + r < 2^r$ szükséges.

Máté: Architektúrák 4. előadás 20

Gyorsító tár (cache – 2.16. ábra)

A processzorok mindig gyorsabbak a memóriáknál.
A CPU lapkára integrálható memória gyors, de kicsi.

Feloldási lehetőség: a központi memória egy kis részét (gyorsító tár) a CPU lapkára helyezni: Amikor egy utasításnak adata van szüksége, akkor először itt keresi, ha nincs itt, akkor a központi memóriában.

Lokalitási elv: Ha egy hivatkozás a memória A címére történik, akkor a következő valószínűleg valahol A közelében lesz (ciklus, mátrix manipulálás, ...).

Ha A nincs a gyorsító tárban, akkor az A-t tartalmazó (adott méretű) blokk (gyorsító sor - cache line) kerül beolvasásra a memóriából a gyorsító tárba.

Máté: Architektúrák 4. előadás 21

Találási arány (h): az összes hivatkozás mekkora hányada szolgálható ki a gyorsító tárból.

Hiba arány: 1-h.

Ha a gyorsító tár elérési ideje: c,
a memória elérési ideje: m, akkor az
átlagos elérési idő = c + (1-h) m.

A gyorsító tár mérete: nagyobb tár – drágább.

A gyorsító sor mérete: nagyobb sor, a hivatkozott cím nagyobb környezete lesz a gyorsító tárban – nagyobb a sor betöltési ideje is. Ugyanakkora tárban kevesebb gyorsító sor fér el.

Máté: Architektúrák 4. előadás 22

Osztott (külön utasítás és adat) gyorsító tár előnyei:

- Egyik szállítószalag végzi az utasítás, másik az operandus előolvasást.
- Az utasítás gyorsító tárat sohasem kell visszaírni (az utasítások nem módosulnak).

Egyesített gyorsító tár: nem lehetséges párhuzamosítás.

Hierarchia:

- elsődleges, a CPU lapkán,
- másodlagos, a CPU-val egy tokban,
- külön tokban.

Máté: Architektúrák 4. előadás 23

Direkt leképezésű gyorsító tár működése: (4.38. ábra)

Bitek: 16 11 3 2
32 bites cím: TAG Vonal (Line) SZÓ BÁJT

Entry 2047 V TAG Data (32 bájt)

1

0

Ha a gyorsító tár Vonal által mutatott sorában V=1 (valid), és a TAG megegyezik a címben lévő TAG-gel, akkor az adat bent van a gyorsító tárban (ebben a sorban).

Máté: Architektúrák 4. előadás 24

Halmazkezelésű (csoportasszociatív) gyorsító tár

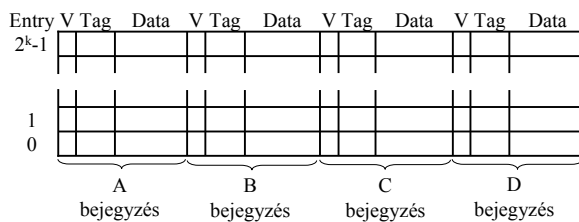
Ha egy program gyakran használ olyan szavakat, amelyek távol vannak egymástól, de ugyanoda képződnek le a gyorsító tárban, akkor sűrűn kell cserélni a gyorsító sort.

Ha minden címhez n bejegyzés van, akkor n utas halmazkezelésű gyorsító tárról beszélünk.

Gyakori a 2 és 4, újabban a 8 utas kezelés.

LRU (Least Recently Used) algoritmus: gyorsító sor betöltése előtt a legrégebben használt bejegyzés kerül ki a gyorsító tárból.

Halmaz kezelésű gyorsító tár (4.39. ábra)



Ha a gyorsító tár **Vonal** által mutatott sorában az **A, B, C** és **D** bejegyzések egyikében **TAG** megegyezik a címben lévő **TAG**-gel, és a hozzá tartozó **V=1** (valid), akkor az adat bent van a gyorsító tárban (ebben a bejegyzésben).

Memóriába írás

Stratégiák:

Írás áteresztés (write through): az írás a memóriába történik. Ha a cím a gyorsító tárban van, oda is be kell írni, különben el kellene dobni a gyorsító sort.

Késleltetett írás (write deferred, write back): ha a cím bent van a gyorsító tárban, akkor csak a gyorsító táriba írunk, a memóriába csak gyorsító sor cserénél.

Ha a cím nincs a gyorsító tárban, akkor előtte betölthetjük: **írás allokalás (write allocation)** – többnyire ezt alkalmazzák késleltetett írás esetén.

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**).

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

Karakterkódolás

ASCII (American Standard Code for Information Interchanges), 7 bites: vezérlőkarakterek, 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.

További nem numerikus típusok

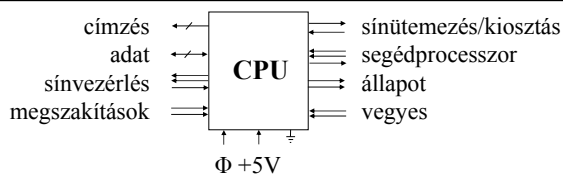
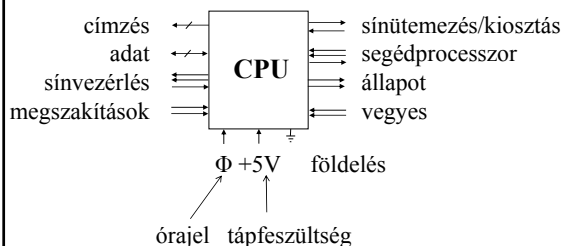
Logikai érték (boolean): igaz, hamis.
Leggyakrabban egy bájtban (szóban) ábrázolják.
Bit térkép.

Mutató (pointer): memória cím.

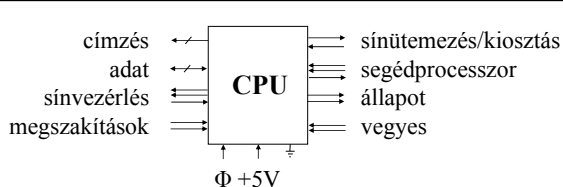
Bit: kapcsolók, lámpák beállítására, lekérdezésére beágyazott rendszerekben.

CPU (Central Processing Unit)

Általában egyetlen lapkán van. **Lábakon** keresztül kommunikál a többi egységgel (3.34. ábra).

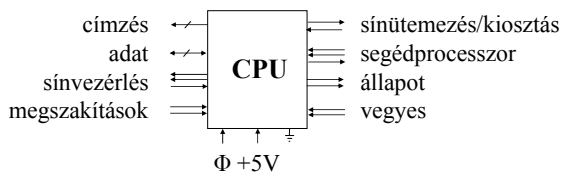


Lábak (**pins**) három típusa: **cím, adat, vezérlés**. Ezek párhuzamos vezetéseken, az ún. **sínen** keresztül kapcsolódnak a **memória**, az **I/O** egységek hasonló lábaihoz.



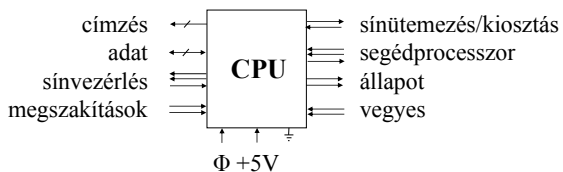
Lényeges a cím- és adatlábak száma (3.34. ábra):

- 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 $n = 8, 16, 32, 36, 64$).



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.



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.

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

CPU lapka

regiszterek

ALU

Lapkán belüli sín

sínvezérlő

memória-sín

memória

B/K sín

lemez

modem

nyomtató

Máté: Architektúrák 4. előadás 37

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

Mesterek (masters): aktív (kezdeményező) berendezések (CPU, lemez vezérlő), **3.35. ábra.**

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 38

A sínhez kapcsolódó lapkák lényegében erősítők.

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

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

Mester–szolgnál: **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 39

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

8086

20 bites cím

vezérlés

80286

4 bites

80386

8 bites

3.37. ábra. A cím szélességének növekedése az elmúlt időszakban

Máté: Architektúrák 4. előadás 40

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?

Máté: Architektúrák 4. előadás 41

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?

Máté: Architektúrák 4. előadás 42

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

43

Feladatok

Milyen adat típusokat ismer?
 Milyen karakter kódolásokat ismer?
 Milyen feladatai vannak a CPU-nak?
 Mi a központi memória feladata?
 Mi a memória cella/rekesz?
 Mit jelent a big endian kifejezés?
 Milyen problémát okoz az eltérő bájtrend?
 Mi a Hamming távolság?
 Mekkora a hexadecimális E6 és C7 Hamming távolsága?
 Hány ellenőrző bit szükséges 256 kódszó 1 hibát javító kódolásához?

Máté: Architektúrák

4. előadás

44

Feladatok

Az alábbi memóriák közül melyik lehetséges, melyik ésszerű? Indokolja meg!

10 bites címek	1024 db	8 bites rekesz
10	1024	12
9	1024	10
11	1024	10
10	10	1024
1024	10	10

Egy régi gépnek 8192 szavas memóriája volt. Miért nem 8000?

Máté: Architektúrák

4. előadás

45

Feladatok

A memória 100-adik bájtjától a 01234567H 4 bájtos számot és – folytatólagosan – az abcd szöveget helyeztük el. Mi az egyes bájtok tartalma, ha a memória big/little endian szervezésű?

Máté: Architektúrák

4. előadás

46

Feladatok

Hol helyezkedhet el a gyorsító tár?
 Mi a lokalitási elv?
 Mit nevezünk találati aránynak?
 Mi a szerepe a találati aránynak?
 Mi a hiba arány?
 Hogy határozható meg az átlagos keresési idő?
 Mi a gyorsító sor?
 Mit nevezünk osztott gyorsító tárnak?
 Mit nevezünk egyesített gyorsító tárnak?
 Mik az osztott gyorsító tár előnyei?

Máté: Architektúrák

4. előadás

47

Feladatok

Mit tartalmaz a direkt leképezésű gyorsító tár egy bejegyzése?
 Mi a TAG?
 Mire szolgál a valid (érvényes) jelzés?
 Hogy működik a direkt leképezésű gyorsító tár?
 Egy memória cella hány helyen lehet egy direkt leképezésű gyorsító tárban?
 Hogy dönthető el, hogy egy memória cella bent van-e egy direkt leképezésű gyorsító tárban?
 Milyen esetben nem hatékony egy direkt leképezésű gyorsító tár?

Máté: Architektúrák

4. előadás

48

Feladatok

Milyen a halmazkezelésű gyorsító tár felépítése?
Hogy működik a halmazkezelésű gyorsító tár?
Mi a halmazkezelésű gyorsító tár előnye a direkt leképezésével szemben?
Mi az **LRU** algoritmus?
Milyen memóriába írási stratégiákat ismer gyorsító tár esetén?
Mit nevezünk írás áteresztésnek (write through)?
Mit nevezünk késleltetett írásnak (write deferred, write back)?
Mit nevezünk írás allokálásnak (write allocation)?

Máté: Architektúrák

4. előadás

49

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?
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?

Máté: Architektúrák

4. előadás

50

Feladatok

Mi a sínprotokoll?
Mi a mester, és mi a szolga?
Mit nevezünk sín vezérlőnek/vevőnek/adó-vevőnek?
Mi a sáv szélesség?
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?

Máté: Architektúrák

4. előadás

51

Az előadáshoz kapcsolódó

Fontosabb tételek

A CISC és a RISC kialakulása
Utasítás és processzor szintű párhuzamosítás
Központi memória, bájtrend
Hamming távolság. Hibaészlelő, hibajavító kódok
Gyorsító tár (cache). Találati és hiba arány. Egyesített és osztott gyorsító tár. Direkt leképezésű és halmaz kezelésű gyorsító tár. Memóriába írás
Adat típusok, karakter kódolás
CPU, Sínek. Sín protokoll. Mester – szolga. Sín vezérlő, vevő, adóvevő. Sáv szélesség, sín szélesség.

Máté: Architektúrák

4. előadás

52