

Október 18, 19, 20, 21-én teszt
az Irinyi 227-es teremben
a MOODLE vizsgáztató programmal
az október 18-a előtt
elhangzott előadások anyagából.

A vizsgáztató program az október 11-ével kezdődő
héten kipróbálható, gyakorolható lesz.

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

Máté: Architektúrák

4. előadás

1

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

2

Találati 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

3

Osztott (külön utasítás és adat) gyorsító tár előnyei:

- Egyik szállítószalag végzi az utasítás,
másik az operandus előolvasást.
- Az utasítás gyorsító tárát 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.

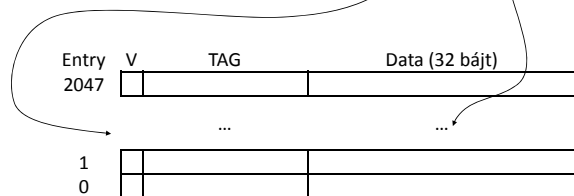
Máté: Architektúrák

4. előadás

4

Direkt leképezésű gyorsító tár működése: (4.38. ábra)

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



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

5

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.

Máté: Architektúrák

4. előadás

6

Halmaz kezelésű gyorsító tár (4.39. ábra)

Entry	V	Tag	Data	V	Tag	Data	V	Tag	Data	V	Tag	Data
2^k-1												
1												
0												

A bejegyzés
B bejegyzés
C bejegyzés
D bejegyzés

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

Máté: Architektúrák 4. előadás 7

Memóriába írás

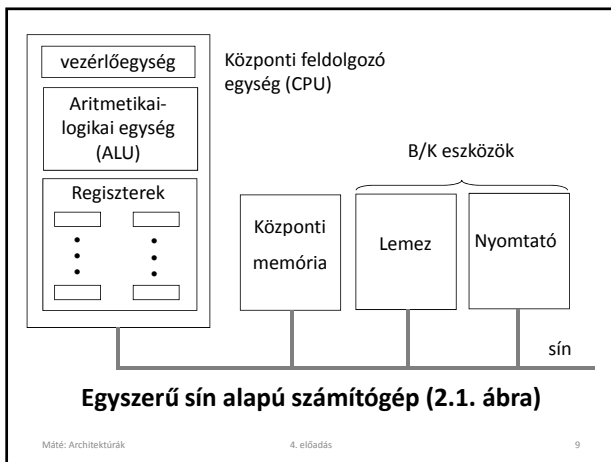
Stratégiák:

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 tárba írunk, a memóriába csak gyorsító sor cserénél.

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

Máté: Architektúrák 4. előadás 8



CPU feladata: a memóriában tárolt program végrehajtása

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:
 - utasításszámláló (Program Counter): **PC**,
 - utasításregiszter (Instruction Register): **IR**,
- adatút (data path, 2.2. ábra).

Máté: Architektúrák 4. előadás 10

Adatút (data path, 2.2. ábra).

- A regiszter készletből feltöltődik az **ALU** két bemenő regisztere
- ALU**
- Az eredmény az **ALU** kimenő regiszterébe kerül
- Az **ALU** kimenő regiszteréből a kijelölt regiszterbe kerül az eredmény

Nem biztos, hogy az **ALU** be- és kimenő regiszterei tényleges regiszterként vannak kialakítva.

Máté: Architektúrák 4. előadás 11

CPU (Central Processing Unit) feladatai

- a végrehajtandó 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 12

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 13

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 14

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 15

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 16

Párhuzamosítás: utasítás vagy processzor szintű.
Utasítás szintű: szállítószalag, csővezeték (pipelining).
Kezdetben:

Utasítás beolvasás → Utasítás végrehajtás

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

S1
utasítás beolvasó egység

S2
utasítás dekódoló egység

S3
operandus beolvasó egység

S4
utasítás végrehajtó egység

S5
eredmény visszairó egység

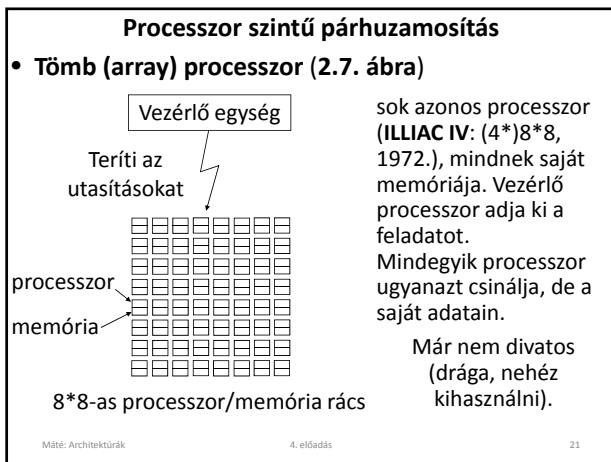
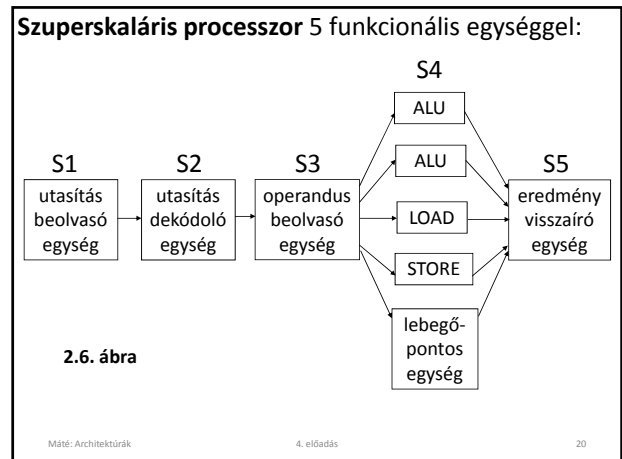
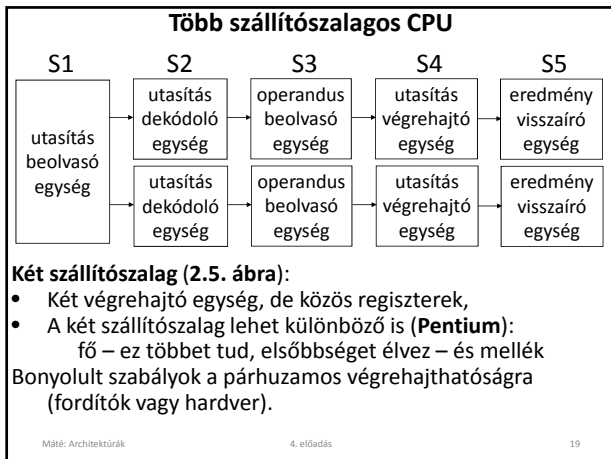
Máté: Architektúrák 4. előadás 17

	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 18



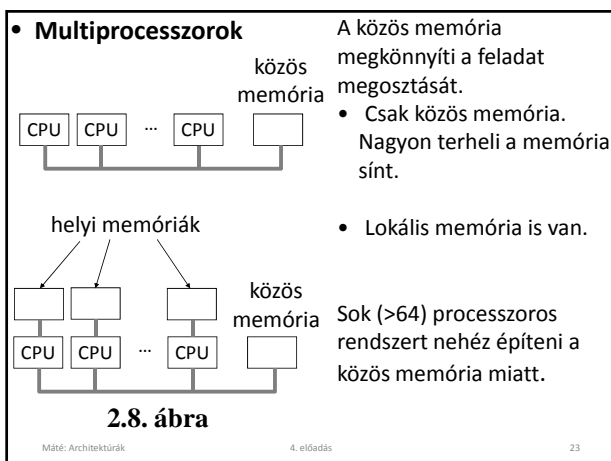
- **Vektor processzorok**

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 22



- **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 24

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

25

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

26

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.

Máté: Architektúrák

4. előadás

27

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.

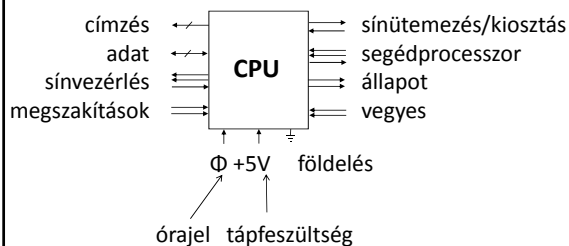
Máté: Architektúrák

4. előadás

28

CPU (Central Processing Unit)

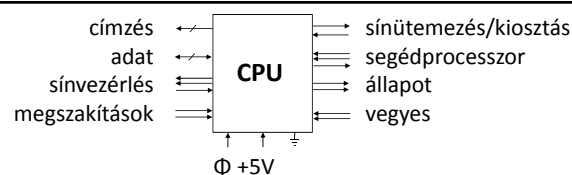
Általában egyetlen lapkán van. **Lábakon** keresztül kommunikál a többi egységgel (**3.34. ábra**).



Máté: Architektúrák

4. előadás

29



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

Máté: Architektúrák

4. előadás

30

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 31

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 32

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 33

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

Máté: Architektúrák Csak az elv kell 4. előadás 34

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 35

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.

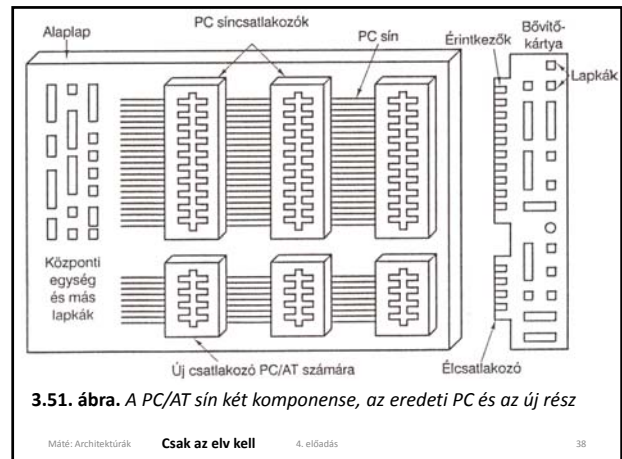
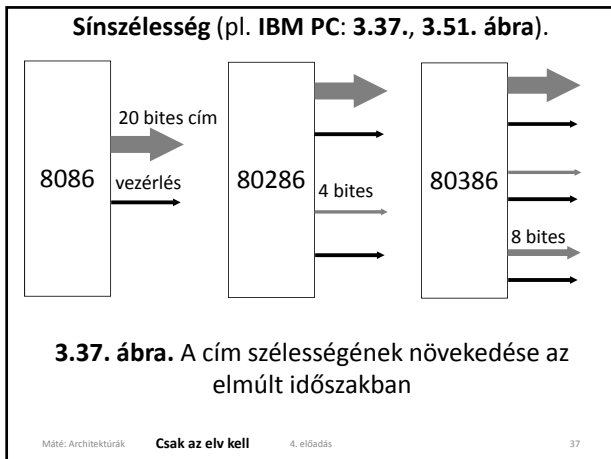
Sávszélesség: (továbbítható bitek száma) / sec.

Sávszélesség növelése:

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

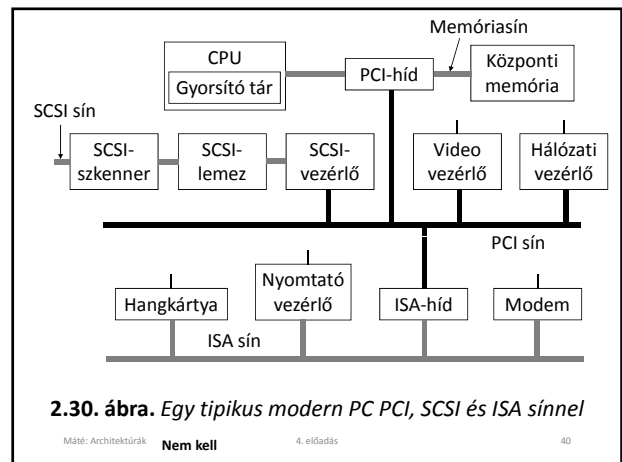
Sínszélesség: több vezeték → drágább, kompatibilitás.

Máté: Architektúrák 4. előadás 36



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 sítet igényel!
Kívánság: PC cseréjénél megmaradhasson 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 39



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 41

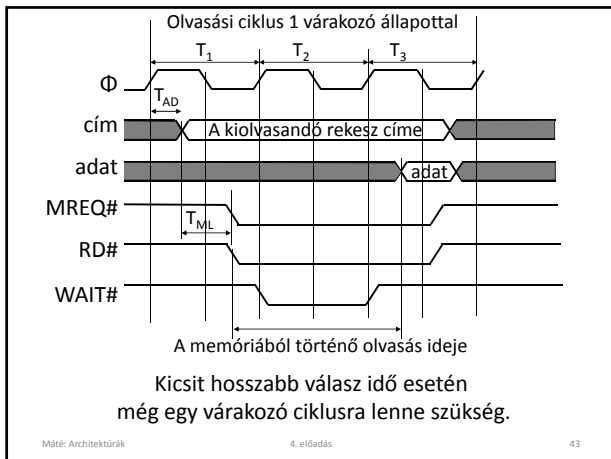
Sínek időzítése

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 sínre		11	ns
T_{ML}	Cím a sínen van MREQ# előtt	6		ns
...

Máté: Architektúrák Csak az elv kell 4. előadás 42



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 44

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 45

Feladatok

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

Máté: Architektúrák 4. előadás 46

Feladatok

- Rajzoljon halmazkezelésű gyorsító tárat! Hogy működik?
- Mi a halmazkezelésű gyorsító tár előnye a direkt leképezésűvel 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 47

Feladatok

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

Máté: Architektúrák 4. előadás 48

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

49

Feladatok

Milyen párhuzamosítási lehetőségeket ismer?
 Mi az utasítás szintű párhuzamosítás?
 Szemléltesse az utasítás szintű párhuzamosságot!
 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

50

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

51

Feladatok

Milyen adat típusokat ismer?
 Milyen karakter kódolásokat ismer? Ismertesse ezek lényegét!
 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?
 Milyen lépésekből áll egy utasítás betöltése?
 Mi a központi memória feladata?

Máté: Architektúrák

4. előadás

52

Feladatok

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

Máté: Architektúrák

4. előadás

53

Feladatok

Mi a sínszélesség?
 Mi a sín aszimmetria?
 Hogy növelhető egy sín sávszélessége?
 Miért nem növelhető szabadon a sín szélessége?
 Miért nem növelhető szabadon a sín sávszélessége?
 Miért volt szükség a sínek szabványosítására?
 Mi az alaplap?
 Mit jelent a sokszorozott (multiplexed) sín?
 Milyen hatása van a sokszorozott sín használatának?

Máté: Architektúrák

4. előadás

54

Feladatok

Mi a sín időzítés, és miért fontos?

Hogy működik a szinkron sín?

Máté: Architektúrák

4. előadás

55

Az előadáshoz kapcsolódó**Fontosabb témák**

Gyorsító tár (cache). Találati és hiba arány. Egyesített és osztott gyorsító tár. Direkt leképezésű és halmazkezelésű gyorsító tár. Memóriába írás.

Máté: Architektúrák

4. előadás

56

Az előadáshoz kapcsolódó**Fontosabb témák**

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 sín.

Máté: Architektúrák

4. előadás

57