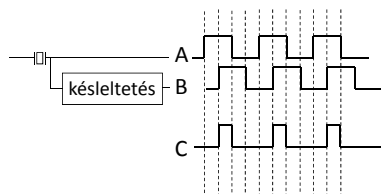


Nem kombinációs áramkörök**Óra (clock, 3.21. ábra):** ciklusidő (cycle time).

Pl.: 500 MHz - 2 nsec.

Finomabb felbontás késleltetéssel.

Aszimmetrikus óra.



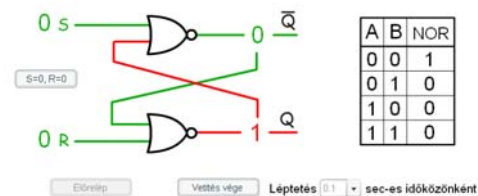
Máté: Architektúrák

3. előadás

1

Memória: „Emlékszik” az utolsó beállításra.**Tároló:** Szint vezérelt (level triggered).**SR tároló** (Set Reset latch, 3.22. ábra, 3.22.swf).Stabil állapot: a két kimenet **0, 1** vagy **1, 0**.**S** (set), **R** (reset) bemenet. ($Q\# \equiv \bar{Q}$)

3.22. ábra NEM-VAGY tároló

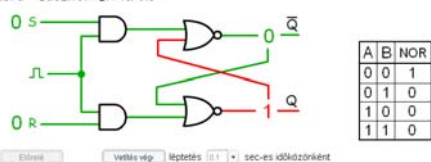


Máté: Architektúrák

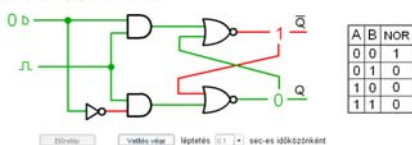
3. előadás

2

3.23. ábra Időzített SR-tároló

Mindkét **SR** tároló indeterminisztikussá válna, ha **S = R = 1** egyszerre fordulna elő.

3.24. ábra Időzített D-tároló



Máté: Architektúrák

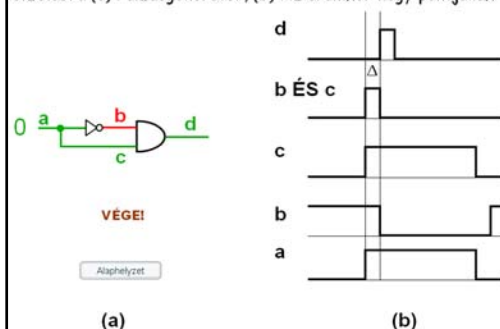
3.23.swf

3. előadás

3.24.swf

3

3.25. ábra (a) Pulzusgenerátor, (b) Az áramkör négy pontjának az idődiagramm

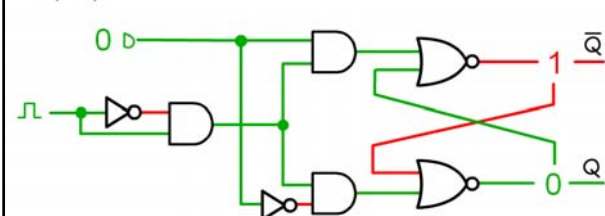
Az inverternek van egy picit (1-10 ns) késleltetése (Δ).

Máté: Architektúrák

3. előadás

3.25.swf

4

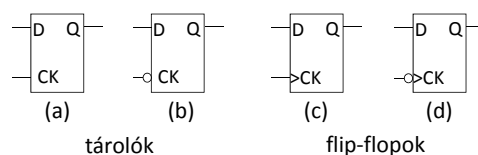
Flip-flop: élvezérelt (edge triggered), **D flip-flop: 3.26. ábra.****D-flipflop**

Máté: Architektúrák

3. előadás

3.26.swf

5

3.27. ábra: Tárolók és flip-flopok

tárolók

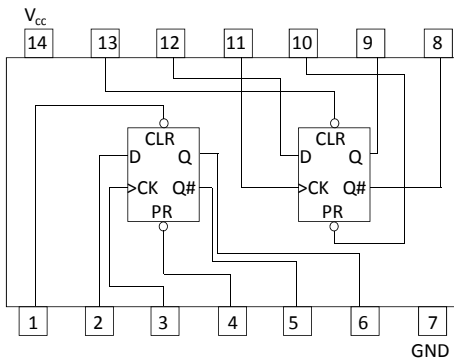
flip-flopok

CK: órajel(a) **CK=1**,(b) **CK=0** szint esetén írja be **D**-t,(c) **CK** emelkedő, (d) **CK** lefelé menő élénél.Sokszor **S** (set, **PR** preset), **R** (reset, **CLR** clear) be- és **Q#** kimenet is van.

Máté: Architektúrák

3. előadás

6

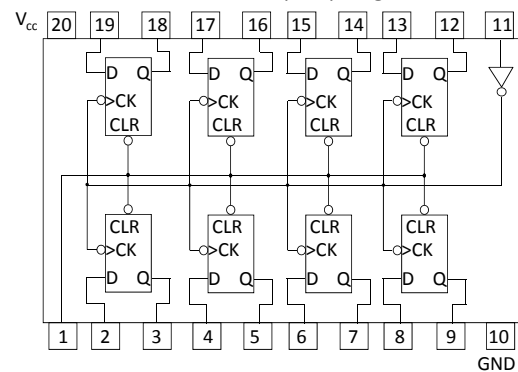
3.28. ábra: (a) 2 független D flip-flop,

Máté: Architektúrák

Nem kell

3. előadás

7

3.28. ábra: (b) közös CK-val és CLR-rel vezérelt 8 bites D flip-flop: regiszter

Máté: Architektúrák

Nem kell

3. előadás

8

Alapvető digitális logikai áramkörök

Integrált áramkör (IC, Integrated Circuit, chip, lapka)
kb. 5x5 mm² szilícium darab kerámia vagy műanyag lapon (tokban), lábakkal (pins). Négy alaptípus:

- SSI (Small Scale Integrated 1-10 kapu),
- MSI (Medium Scale ..., 10-100 kapu),
- LSI (Large Scale..., 100-100 000 kapu),
- VLSI (Very Large Scale ..., > 100 000 kapu).

Máté: Architektúrák

3. előadás

9

Memória szervezése

Elvárás: szavak címezhetősége.

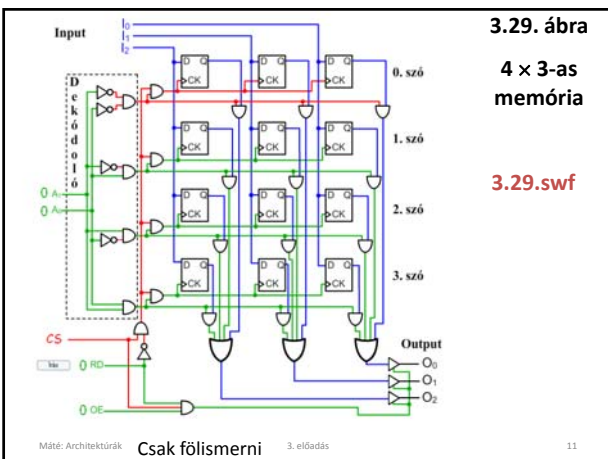
3.29. ábra: Négy db három bites szó. Bemenetek:

- három a vezérléshez,
 - CS (Chip Select): lapka választás,
 - RD (Read): 1: olvasás, 0: írás választása,
 - OE (Output Enable): kimenet engedélyezése.
- kettő a címzéshez (dekódoló),
három a bemenő adatoknak,
három adat kimenet.

Máté: Architektúrák

3. előadás

10



Máté: Architektúrák

Csak fölüsmerni

3. előadás

11

Memória szervezése

Az igazi memóriáknál a bemenet és kimenet közös (kevesebb lábra van szükség): Nem invertáló és invertáló pufferek (ezek három állapotú eszközök, **tri-state device**, 3.30. ábra).

adat be → adat ki
vezérlés

nem invertáló puffer

Ha a vezérlő jel

magas: —————

alacsony: ————/—————

adat be → adat ki
vezérlés

invertáló puffer

Ha a vezérlő jel

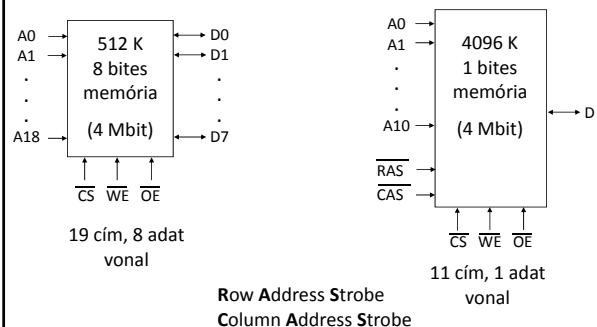
magas: ————/—————

alacsony: —————

Máté: Architektúrák

3. előadás

12

MemórialapokElőnyös, ha a szavak száma **2 hatvány**.**4 Mbit-es memória kétféle szervezése: 3.31. ábra.**

Máté: Architektúrák

3. előadás

13

MemórialapokA jel (bemenet) **beállított** (asserted) vagy **negált**.**CS** beállított: **1**, de **CS#** beállított: **0****a) 512 K** bájtos elrendezés: **19** cím, **8** adat vonal.

b) 2048*2048 bites elrendezés: **11** cím, **1** adat vonal:
Bit kiválasztás sor- (**RAS**: Row Address Strobe) és oszlopindex **CAS** (Column ...) segítségével.
Gyakran alkalmazzák nagyobb memóriáknál, bár a két cím megadása lassíthat.

Nagyobb memóriáknál **1, 4, 8, 16** bites kimeneteket is használnak.

Máté: Architektúrák

3. előadás

14

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 címevel hivatkozhatunk. A rekesz a legkisebb címezhető egység.

Máté: Architektúrák

3. előadás

15

Cím Rekesz/cella



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

n a memória cellák
száma

← Rekesz hossza →

Központi memória (2.9. ábra)

Máté: Architektúrák

3. előadás

16

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

Nem kell

3. előadás

17

Bájtsorrend

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

A legmagasabb helyértékű bájt 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 bájtjainak értéke rendre:

a, b, c, d, akkor a szó értéke:

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

Máté: Architektúrák

3. előadás

18

Bájtsorrend (2.11. ábra)

A memória címek úgy vannak fölírva, hogy a legmagasabb helyértékű bájt 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

3. előadás

19

Bájtsorrend (12. ábra)

A szövegek karaktereit mindkét esetben növekvő bájt sorrendben helyezik el

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

		0	1	2	3					
		T	E	X	T					
4		4	5	6	7					
		78	56	34	12					

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

Problémák a gépek közötti kommunikációban!

Máté: Architektúrák

3. előadás

20

RAM (Random Access Memory)

- **Statikus RAM (SRAM)**. D flip-flop elemekből épül fel. Amíg áram alatt van, tartja a tartalmát. Elérési idő: néhány nsec (cache-nek jók).
- **Dinamikus RAM (DRAM)**: minden bit egy tranzisztor és egy kondenzátor: néhány msec-onként frissíteni kell, de nagyobb adatsűrűség érhető el. Elérési idő: néhány tíz nsec (főmemóriák).
 - régi: **FPM** (Fast Page Mode) sor-, oszlopcím.
 - újabb: **EDO** (Extended Data Output) lehet új memóriahivatkozás, mielőtt az előző befejeződik.
- **SDRAM** (Synchronous **DRAM**). A központi óra vezérli. Blokkos átvitel. Újabban: **DDR** (Double Data Rate). Az órajel föl- és lefutó élénél is van adatátvitel.

Máté: Architektúrák

3. előadás

21

ROM (Read-Only Memory)

ROM: gyárilag kialakított tartalom.

PROM (Programmable **ROM**): a tartalom biztosítékok kiegészítésével alakul ki (a **PLA**-khoz hasonlóan, **3.15. ábra**).

EPROM (Erasable **PROM**): a biztosítékok speciális fénnel kiolvaszthatók és „kijavíthatók”.

EEPROM: elektromos impulzusokkal.

Flash memória: törlés és újírás csak blokkonként. Kb. 100 000 használat után „elkopnak”. Ilyen van a legtöbb MP3 lejátszóban, digitális fényképezőgépben ...

Máté: Architektúrák

3. előadás

22

512 MB-os flash memória (2006)

Máté: Architektúrák

3. előadás

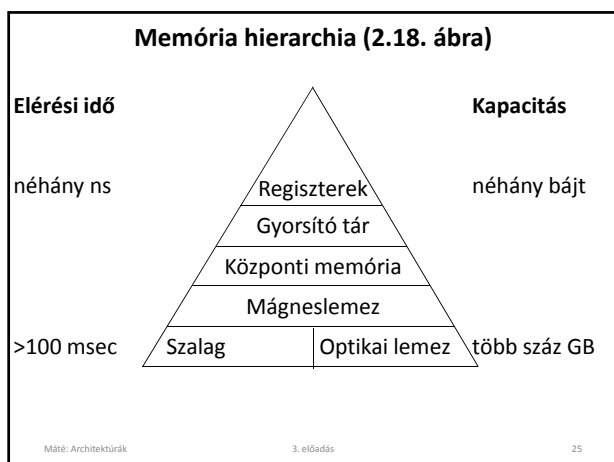
23

1 GB-os flash memória (2007)

Máté: Architektúrák

3. előadás

24



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 3. előadás 26

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 3. előadás 27

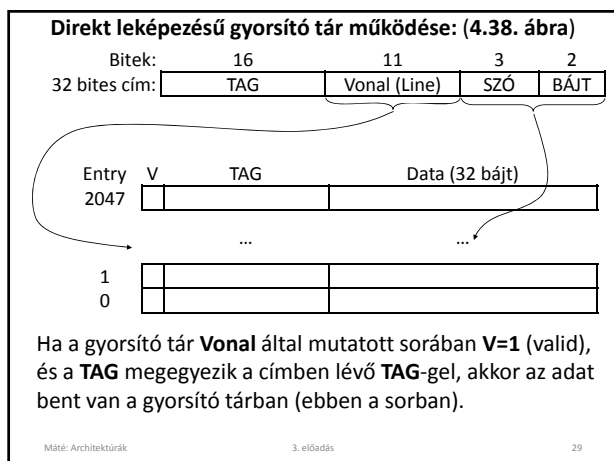
Osztott (külön utasítás és adat) gyorsító tár előnyei:

- 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 3. előadás 28



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 vonalhoz n bejegyzés van, akkor n utas halmazkezelésű gyorsító tárról beszélünk.

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

Máté: Architektúrák 3. előadás 30

Halmaz kezelési gyorsító tár (4.39. ábra)

Entry	V	Tag	Data	V	Tag	Data	V	Tag	Data	V	Tag	Data
2^k-1												
1												
0												
	A			B			C			D		
	bejegyzés			bejegyzés			bejegyzés			bejegyzés		

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).

Máté: Architektúrák

3. előadás

31

LRU (Least Recently Used) algoritmus:

gyorsító sor betöltése előtt a legrégebben használt bejegyzés kerül ki a gyorsító tárból.

Máté: Architektúrák

3. előadás

32

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

3. előadás

33

Hibafelismerő és hibajavító kódolás

Az elektromos hálózatban keletkező áramlökések miatt a memóriák néha hibázhatnak. Az ilyen hibák kivédésére sokszor hibafelismerő vagy hibajavító kódot alkalmaznak.

Kódolás: adat + ellenőrző bitek = **kódszó**.

Két kódszó Hamming távolsága: az eltérő bitpozíciók száma. Pl.:

11001 és

11011 (Hamming) távolsága = 1.

A kódrendszer Hamming távolsága: a két legközelebbi kódszó Hamming távolsága.

Máté: Architektúrák

3. előadás

34

Hibafelismerő kód: bármely két kódszó Hamming távolsága > 1 : paritás bit.

d hibás bit javítása: a kódrendszer 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 $(1 + n)2^m \leq 2^n = 2^{m+r}$,

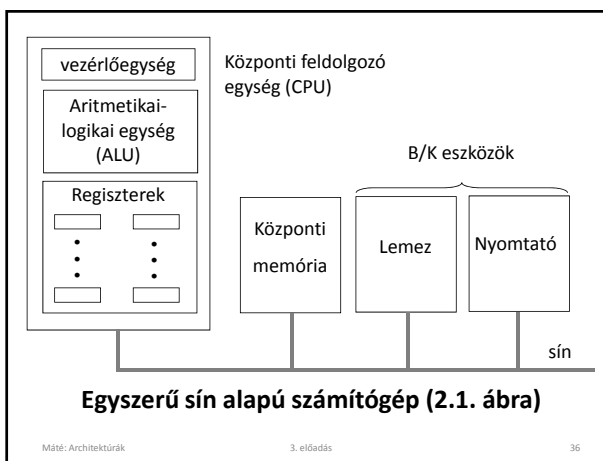
2^m -mel egyszerűsítve: $(1 + n) = 1 + m + r \leq 2^r$,

vagy másképp: $m + r < 2^r$ szükséges.

Máté: Architektúrák

3. előadás

35



Máté: Architektúrák

3. előadás

36

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

Máté: Architektúrák 3. előadás 37

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 3. előadás 38

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 3. előadás 39

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 3. előadás 40

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.

Máté: Architektúrák 3. előadás 41

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 3. előadás 42

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

3. előadás

43

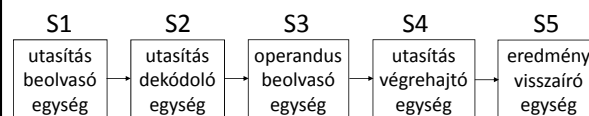
Párhuzamosítás: utasítás vagy processzor szintű.

Utasítás szintű: szállítószalag (csővezeték, pipelining).

Kezdetben:



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



Máté: Architektúrák

3. előadás

44

	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

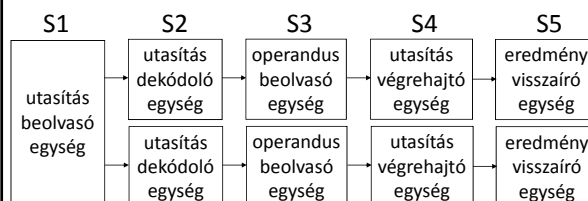
- **Késleltetés (latency):** mennyi ideig tart egy utasítás.
- **Áteresztőképesség (processor bandwidth):** hány MIPS (Million Instruction Per Second) a sebesség.

Máté: Architektúrák

3. előadás

45

Több szállítószalagos CPU



Két szállítószalag (2.5. ábra):

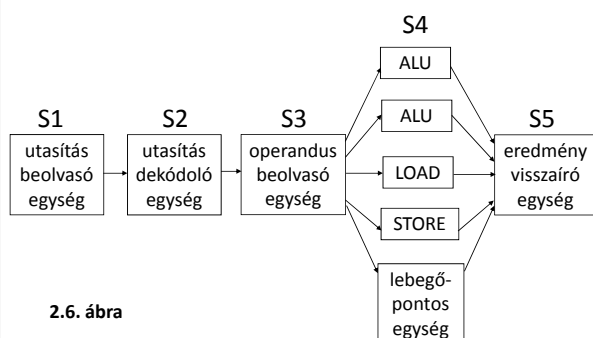
- Két végrehajtó egység, de közös regiszterek,
- A két szállítószalag lehet különböző is (**Pentium**):
fő – ez többet tud, elsőbbséget élvez – és mellék
Bonyolult szabályok a párhuzamos végrehajthatóságra (fordítók vagy hardver).

Máté: Architektúrák

3. előadás

46

Szuperskaláris processzor 5 funkcionális egységgel:



2.6. ábra

Máté: Architektúrák

3. előadás

47

Feladatok

Milyen nem kombinációs áramköröket ismer?
Kombinációs áramkör-e az **ALU**?
Hogy érhetünk el az órajelnél finomabb időzítést?
Hány stabil állapota van az **SR** tárolónak?
Rajzoljon **SR** tárolót! Hogy működik?
Rajzoljon időzített **SR** tárolót! Hogy működik?
Rajzoljon időzített **D** tárolót! Hogy működik?
Rajzoljon pulzuszgenerátort! Mi a működési elve?
Mi a különbség a tároló és a flip-flop között?
Hogy működik az invertáló és a nem invertáló puffer?
Miért használnak a memóriáknál invertáló vagy nem invertáló puffert?

Máté: Architektúrák

3. előadás

48

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

3. előadás

49

Feladatok

Hogy címezhető meg n címlábon 2^n -nél nagyobb memória?

Mit jelent, hogy a **CS/CS#** bemenet beállított/negált?

Mit tud a központi memóriáról?

Mi a memória cella/rekesz?

Mit jelent a big/little endian kifejezés?

Milyen problémát okoz az eltérő bájtrend?

Mi a **RAM**? Milyen **RAM**-okat ismer? Jellemezze ezeket!

Máté: Architektúrák

3. előadás

50

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

3. előadás

51

Feladatok

Jellemezze az **SRAM**-ot!

Jellemezze a **DRAM**-ot!

Mit jelent az **FPM** rövidítés?

Mit jelent az **EDO** rövidítés?

Jellemezze az **SDRAM**-ot!

Mit jelent az **DDR** rövidítés?

Mit jelent a **ROM** rövidítés?

Jellemezze a **PROM**-ot!

Jellemezze az **EPROM**-ot!

Jellemezze az **EEPROM**-ot!

Milyen memória van a legtöbb fényképezőgépben?

Máté: Architektúrák

3. előadás

52

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

3. előadás

53

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 tárban?

Hogy dönthető el, hogy egy memória cella bent van-e egy direkt leképezésű gyorsító tárban?

Milyen esetben hatékony/nem hatékony egy direkt leképezésű gyorsító tár?

Máté: Architektúrák

3. előadás

54

Feladatok

Rajzoljon halmazkezelésű (csoportasszociatív) gyorsító tárat! Hogy működik?

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

3. előadás

55

Feladatok

Mi a Hamming távolság?

Mekkora a hexadecimális E6 és C7 Hamming távolsága?

Mekkora a bináris 11001011 és 10011111 Hamming távolsága?

Hány ellenőrző bit szükséges 32 kódszó 1 hibát javító kódolásához?

Hány ellenőrző bit szükséges 256 kódszó 1 hibát javító kódolásához?

Máté: Architektúrák

3. előadás

56

Feladatok

Milyen részei vannak a CPU-nak?

Mi az ALU?

Mi az adatút?

Mi a regiszter?

Mik a CPU feladatai?

Mit jelent az implicit operandus kifejezés?

Máté: Architektúrák

3. előadás

57

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

3. előadás

58

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

3. előadás

59

Az előadáshoz kapcsolódó**Fontosabb témák**

Nem kombinációs áramkörök.

Óra, tárolók, flip-flop-ok

Központi memória, bájtsorrend

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

Hamming távolság. Hibaészlelő, hibajavító kódok

A **CPU** részei, feladatai, adatút

A **CISC** és a **RISC** kialakulása

Utasítás szintű párhuzamosítás

Máté: Architektúrák

3. előadás

60