

A számítógépek felépítése 4.a: CISC és RISC processzorok

Markó Tamás
PTE TTK, 2003

2003.07.17.

Markó Tamás, PTE TTK

1

A rádiótelefonokat kérem KIKAPCSOLNI!

2003.07.17.

Markó Tamás, PTE TTK

2

A RISC architektúra létrejötte

- CISC: Complex Instruction Set Computer
- RISC: Reduced Instruction Set Computer
- Megfigyelések:
 - Az utasításokat nem egyforma gyakorisággal használjuk
Példa: IBM 370. 200 különböző utasítás, de
 - 10 adja a felhasználás (hely és idő) 80 %-át
 - 21 utasítás adja a 95 %-át
 - 30 utasítás adja a 99 %-át.
 - Gyakran több egyszerű utasítás egymás után gyorsabb, mint egy bonyolult
Példa: IBM 370
A LOAD MULTIPLE utasítás $n \geq 4$ esetén lassabb, mint n db egyszerű LOAD utasítás

4

A RISC elterjedéséhez vezető korlátok

1. A chipek mérete:
 - CISC rendszerű számítógépet nehezebb összezsúfolni a korlátozott méretre
2. A technológia gyors fejlődése
 - rövid a tervezésre fordítható idő
 - egyszerűbb eszközöket kell előállítani
3. Az erőforrások eltérő mértékben „értékesek”
 - a legértékesebb a tárolóhely (a regiszterek és a cache)
 - viszonylag sok regiszter ? viszonylag kevés logika

2003.07.17.

Markó Tamás, PTE TTK

5

Az első RISC processzorok

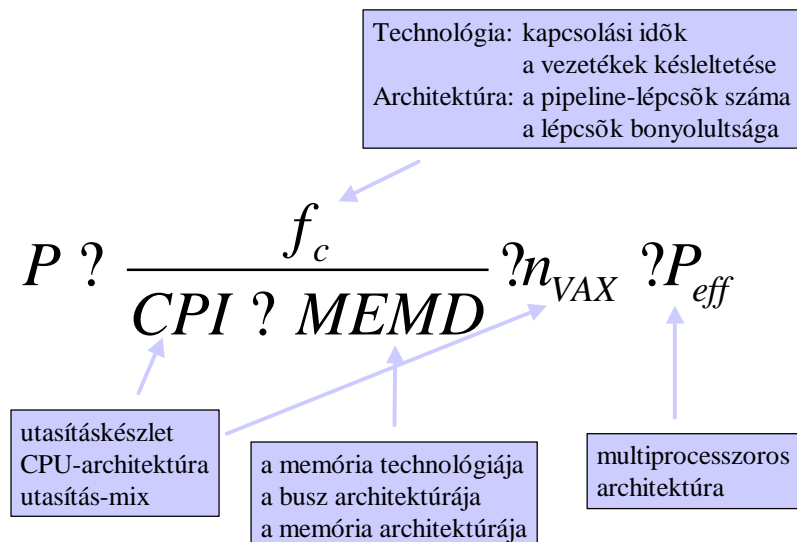
- Az 1980-as évek elején különböző RISC processzorok jöttek létre
- Berkeley RISC I/RISC II (-> SPARC)
- Stanford MIPS
- IBM 801 (a legelső RISC, -> Power PC)

2003.07.17.

Markó Tamás, PTE TTK

6

A teljesítményt befolyásoló tényezők



2003.07.17.

Markó Tamás, PTE TTK

15

A RISC módszer 1.

Magas órajel-frekvencia:

- egyszerű architektúra
- pipeline használata (ezáltal egyszerű műveletek egy-egy lépcsőben)

Alacsony CPI:

- egyszerű utasítások
- pipeline használata (látható)
- koncentráció a fontos műveletekre

2003.07.17.

Markó Tamás, PTE TTK

16

A RISC módszer 2.

Alacsony MEMD:

- lehetőleg regiszterben lévő operandusokat használunk
- load/store architektúra
- sok regiszter ? egyszerű utasításokra van szükség
- (register windowing)

Magas n_{VAX} :

- a szubrutinhívások támogatása
- (register windowing)
- célszerű utasítások

Magas P_{eff} :

- szuperskalár architektúra
- a cache hatékony használata

2003.07.17.

Markó Tamás, PTE TTK

17

Példa az egyszerű RISC architektúrára

Berkeley RISC I/RISC II

(a RISC processzorok egyik őse, a SPARC elődje)

Célkitűzések:

1. 1 utasítás végrehajtása órajelciklusonként
2. az utasítások kb. olyan bonyolultak legyenek, mint a CISC gépek mikROUTASÍTÁSAI
3. minden utasítás azonos formátumú legyen
4. csak a *load* és a *store* parancs férjen a memóriához
5. a magasszintű programozási nyelvek támogatása
6. 32 bites címek, 32 bites adatok

2003.07.17.

Markó Tamás, PTE TTK

18

C és Pascal programok vizsgálata (Berkeley)

	Call/return	Loop	Értékkadás	If	Egyéb
Forrásnyelvi utasítások (%)	12	3	38	43	4
VAX gépi utasítások (%)	33	32	13	21	1
Memória-hozzáférés (%)	45	26	15	12	1

Operanduszok:

- 60 % skalár (ebből 80 % lokális)
- 20 % egész konstans
- 20 % tömb, rekord, valós, pointer ..

? szükség van gyors call/return mechanizmusra

2003.07.17.

Markó Tamás, PTE TTK

19

A Berkeley RISC regiszterei

- **Emlékeztető:**

- A szubrutinhívások igen gyakoriak, és ott paraméterátadás is van
- A memória-hozzáférés drága (a RISC-nél: egy helyett két órajel-ciklus)
- ? Megpróbálják szubrutinhívásnál a paramétereket memória-hozzáférés nélkül átadni

- **Ki kell küszöbölni:**

- Szubrutinhívásnál a regiszterek tartalmának elmentését (alaphelyzetben szükséges, mert a szubrutinnak saját regiszter-tartalmakkal kell rendelkeznie)
- A memórián keresztül történő paraméterátadást

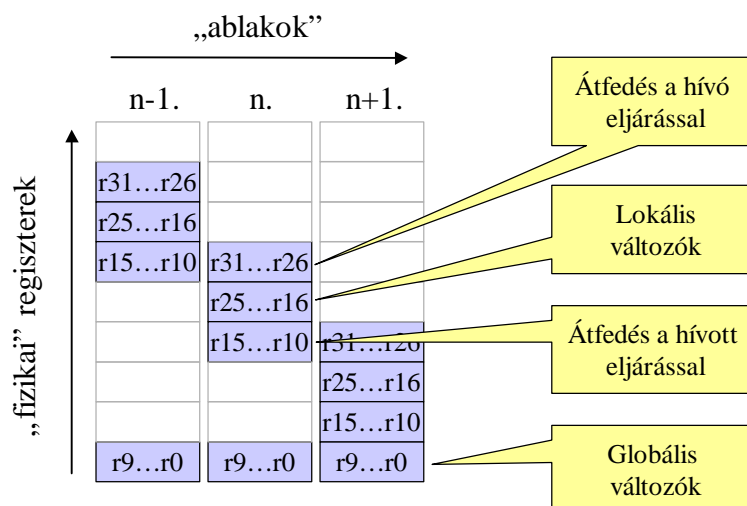
- **Megoldás:**

- Fizikailag több regiszterkészlet (ablak), minden programnak egy. Így szubrutinhíváskor nem kell menteni, csak az ablak mutatóját (CWP, Current Window Pointer) átkapcsolni.
- A regiszter-ablakok átlapolják egymást, a közös területen történik a paraméterátadás

Markó Tamás, PTE TTK

25

Register Windowing a Berkeley RISC-nél



2003.07.17.

Markó Tamás, PTE TTK

26

Hány regiszter-ablak kell?

- Kísérleti adatok:
 - 4-8 ablak: az ablakok kifogyásának (window overflow, amikor a regisztereket a memóriába kell menteni) valószínűsége $< 1\%$
- Innen származik a SPARC megoldása is:
 - 7 ablak, mindegyikben 8 in, 8 local, 8 out
 - + 8 globális regiszter
 - ? $7 * 16 + 8 = 120$ regiszter

Ami az egyikben out,
az a másikban in

2003.07.17

Markó Tamás, PTE TTK

28

CISC ? RISC

CISC

- „könnyebb programozás”
- sok (~200) utasítás
- összetett utasítások, sokféle címezési mód
- utasítás végrehajtás több órajel alatt
- mikroprogramozott
- ...

RISC

- egyszerűbb, gyorsabb
- kevés (~50) utasítás és címezési mód
- rögzített utasításforma
- utasításvégrehajtás 1 órajel alatt
- kevés memória-hozzáférés
- sok regiszter
- huzalozott vezérlő
- pipeline feldolgozás

Dr. Istenes Zoltán, ELTE

30