

Ha egy mikroművelet minden inputja rendelkezésre áll, akkor az esetleges **WAR** vagy **WAW** függőséget a 120 firkáló regiszter segítségével kiküszöböli. **RAW** függőség esetén a mikroműveletet várakoztatja, és a rákövetkező mikroműveleteket kezdi feldolgozni. Egyszerre akár 126 utasítás feldolgozása is folyamatban lehet, köztük 48 betöltés és 24 tárolás. Az utasítások a programnak megfelelő sorrendben kerülnek az ütemezőbe, eltérő sorrendben kezdődhet a végrehajtásuk, de az előírt sorrendben fejeződnek be.

**Pontos megszakítás:** a megszakítás előtti összes utasítás befejeződött, az utána következőkből egy sem kezdődött el.

Máté: Architektúrák 9. előadás 3

A **Lefoglaló/átnevező egység** a két sor megfelelőjébe teszi a végrehajtható mikroutasításokat. Az ALU-k az órajel kétszeres sebességével dolgoznak, nehéz folyamatosan munkát adni nekik.

Minden órajel ciklusban egy betöltés és egy tárolás is végrehajtható.

Máté: Architektúrák 9. előadás 4

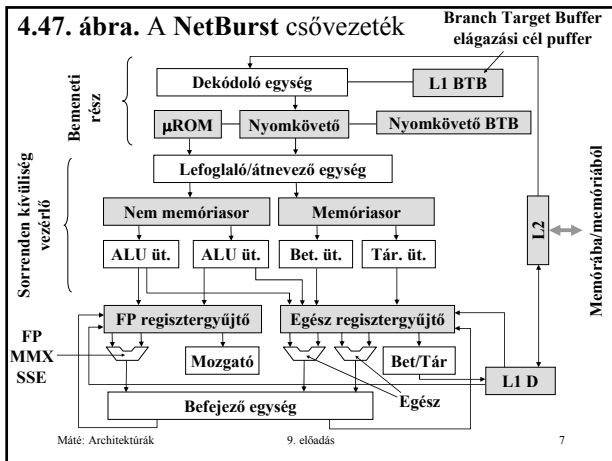
Az egyik egész aritmetikájú **ALU** az összes logikai, aritmetikai, és elágazó, a másik csak az összeadó, kivonó, léptető és forgató utasítás végrehajtására képes.

Mindkét regisztergyűjtő 128 regisztert tartalmaz, időben változik, hogy melyikben van **EAX**, ...

Máté: Architektúrák 9. előadás 5

A befejező egység feladata, hogy az utasítások a programnak megfelelő sorrendben fejeződjenek be. **L1** 4 utas halmazkezelésű, írás áteresztő gyorsítótár 64 bájtos gyorsító sorral. Nem lehet **L1**-et módosítani, amíg a tárolást megelőző műveletek be nem fejeződtek (24 bejegyzéses tároló puffer), de ha egy betöltés utasítás onnan akar olvasni, ahova egy korábbi tárolt, akkor a tárolások pufferéből megkaphatja a kért adatot (tárolás utáni betöltés).

Máté: Architektúrák 9. előadás 6



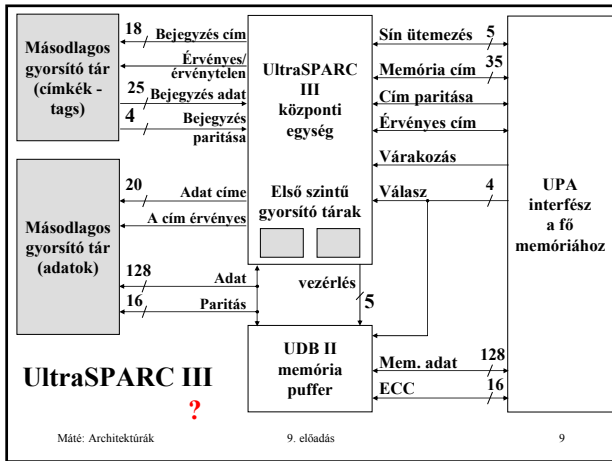
**UltraSPARC III (2000)**

64 bites RISC gép, felül kompatibilis a 32 bites SPARC V8 architektúrával és az UltraSPARC I, II-vel. Új a VIS 2.0 utasításkészlet (3D grafikus alkalmazásokhoz, tömörítéshez, hálózat kezeléshez, jelfeldolgozáshoz, stb.).

Több processzoros alkalmazásokhoz készült. Az összekapcsoláshoz szükséges elemeket is tartalmazza.

2000-ben 0.6, 2001-ben 0.9, 2002-ben 1.2 GHz, órajel ciklusonként 4 utasítás **elvégzését tudja indítani.**

Máté: Architektúrák 9. előadás 8



**UltraSPARC III**

CPU 29 millió tranzisztor, 4 CPU közös memóriával használható. 1368 láb (3. 47. ábra). 64 (jelenleg csak 43) bites cím és 128 bites adat lehetséges.

**Belső gyorsító tár (32 KB utasítás + 64 KB adat). 2 KB előre betöltő és tároló gyorsítótár L2 eléréséhez.**

A gyorsító sor (cache line) mérete **64 (32?) B**.

Az L2 gyorsítótár osztott, külső **1, 4 vagy 8 MB**

A gyorsító sor mérete 64, 256 illetve 512 B

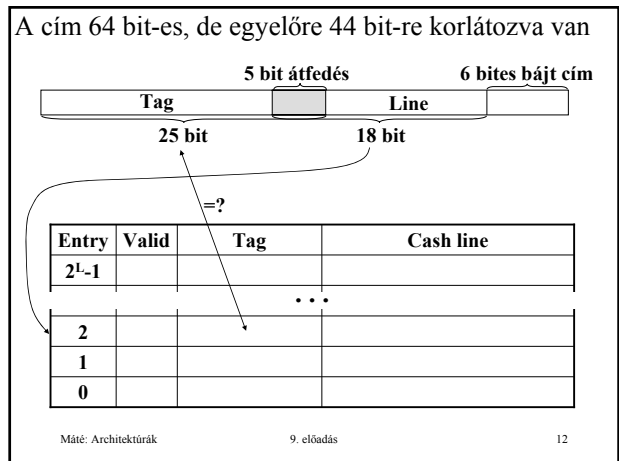
Máté: Architektúrák 9. előadás 10

Az **UltraSPARC II-nél 0.5-16 MB-os az L2 tár és 8 K - 256 K db 64 B-os gyorsító sor (cache line)** lehet.

A címezéséhez **13 – 18 bit** szükséges.

A CPU mindig **18 bites Line címet (Bejegyzés cím)** ad át. Csak maximális méret esetén van mind a **18 bit** címezésre kihasználva.

Máté: Architektúrák 9. előadás 11

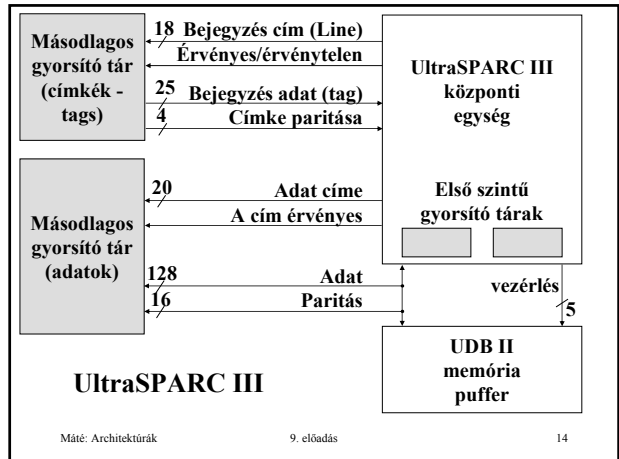


**512 KB-os gyorsító tár esetén a 44 bites cím felosztása:**  
**Tag: 25 bit, Line: 13 bit, bájt cím: 6 bit = 44 bit.**  
**16 MB-os tár esetén 18 bites Line kell, és 20 bites Tag (Bejegyzés adat) is elég lenne, de ilyekor – hogy a CPU egységesen működhessen – a gyorsító tárban tárolt 20 bites Tag-et a gyorsító tár kiegészíti Line 5 legmagasabb helyértékű bitjével.**  
 Az **Adat címe** a gyorsító sor címén (**Bejegyzés cím, Line**) kívül még **2** bitet tartalmaz, mert egy átvitel során a gyorsító sornak csak negyed része (**16 bájt**) mozgatható.

Máté: Architektúrák

9. előadás

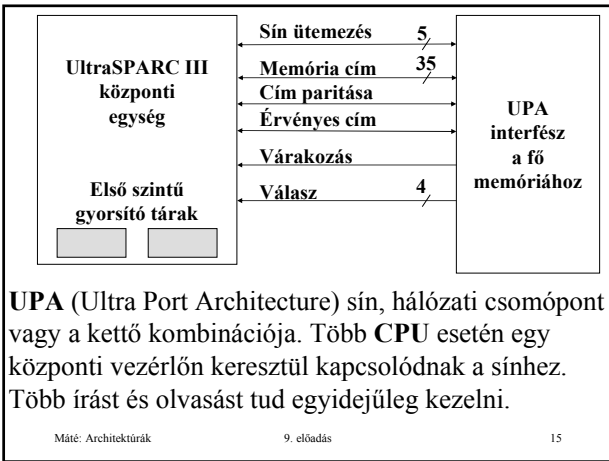
13



Máté: Architektúrák

9. előadás

14

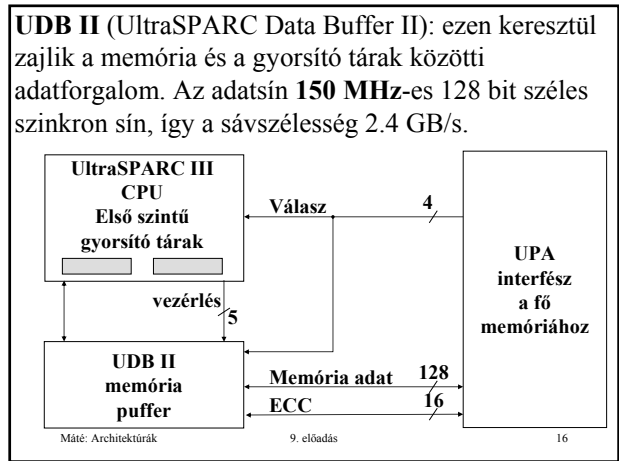


**UPA (Ultra Port Architecture)** sín, hálózati csomópont vagy a kettő kombinációja. Több CPU esetén egy központi vezérlőn keresztül kapcsolódnak a sínhez. Több írást és olvasást tud egyidejűleg kezelni.

Máté: Architektúrák

9. előadás

15

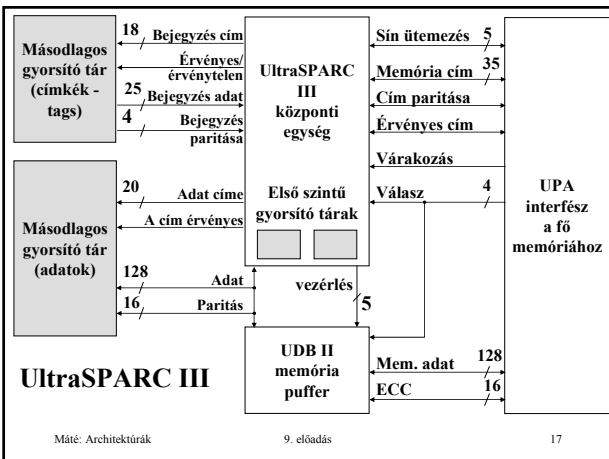


**UDB II (UltraSPARC Data Buffer II):** ezen keresztül zajlik a memória és a gyorsító tárok közötti adatforgalom. Az adatsín **150 MHz-es** 128 bit széles szinkron sín, így a sávszélesség 2.4 GB/s.

Máté: Architektúrák

9. előadás

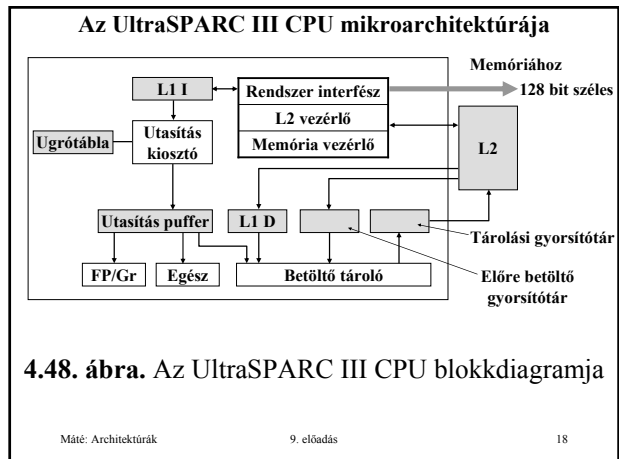
16



Máté: Architektúrák

9. előadás

17



**4.48. ábra.** Az UltraSPARC III CPU blokkdiagramja

Máté: Architektúrák

9. előadás

18

**L1 I** 32 KB 4 utas halmazkezelésű  
Az utasítás kiosztó ciklusonként 4 utasítást tud kiosztani  
**Ugrótábla** 16K bejegyzés elágazás jövendöléshez (cél cím is)  
16 elemű **Utasítás puffer**

Két egész aritmetikájú ALU + regiszterek + firkáló regiszterek  
Lebegőpontos ALU-k: összeadó/kivonó, szorzó/osztó + grafikai utasítások + 32 regiszter

Máté: Architektúrák 9. előadás 19

**L1 I** Utasítás kiosztó  
**Rendszer interfész** L2 vezérlő Memória vezérlő  
Memóriához 128 bit széles  
**L2**  
Tárolási gyorsítótár 2 KB  
Előre betöltő gyorsítótár 2 KB  
**L1 D** Utasítás kiosztó  
Betöltő tároló

**L1 D** 64 KB-os 4 utas halmazkezelésű, írás áteresztő, 32 bájtos gyorsító sor. Feltételezett betöltésre 2 KB előre betöltő gyorsítótár. 2 KB tárolási gyorsítótár.  
**Memória vezérlő**: virtuális → fizikai cím.

Máté: Architektúrák 9. előadás 20

**UltraSPARC III CPU mikroarchitektúrája**  
A SPARC sorozat RISC elgondoláson alapul. A legtöbb utasításnak két forrás és egy cél regisztere van.  
Előre betöltés: speciális utasításokkal, és a visszafelé kompatibilitás miatt hardveresen is.  
2 bites elágazás jövendölő + statikus elágazás jövendölés.

Máté: Architektúrák 9. előadás 21

**UltraSPARC III csővezetéke (4.49. ábra)**

Address generation, cím generáló. Ugrás, csapda, ...  
Az eltolás részben lévő utasítást mindig végrehajtja!  
**Preliminari Fetch**, előzetes betöltő. Legfeljebb 4 utasítást képes betölteni L1 I-ből, nézi, hogy van-e közöttük elágazó, elágazás jövendölés.  
**Branch target**, elágazási cél. Ha kell ugrani, → A

Máté: Architektúrák 9. előadás 22

**UltraSPARC III csővezetéke**

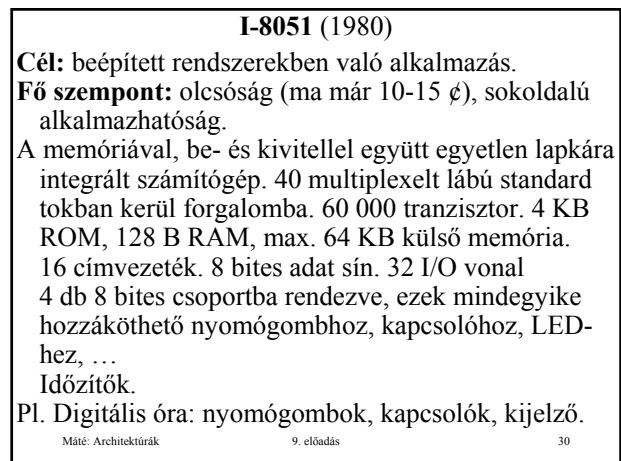
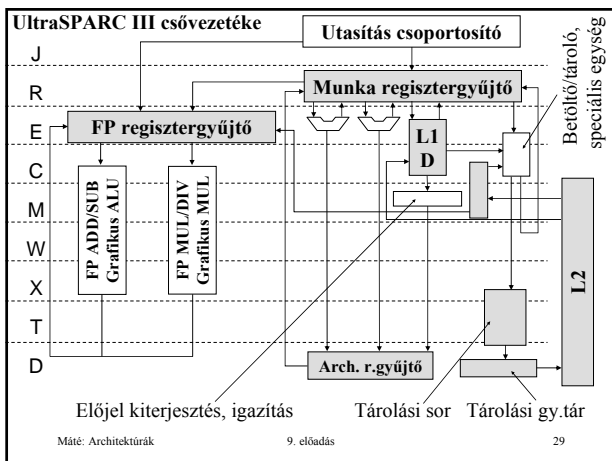
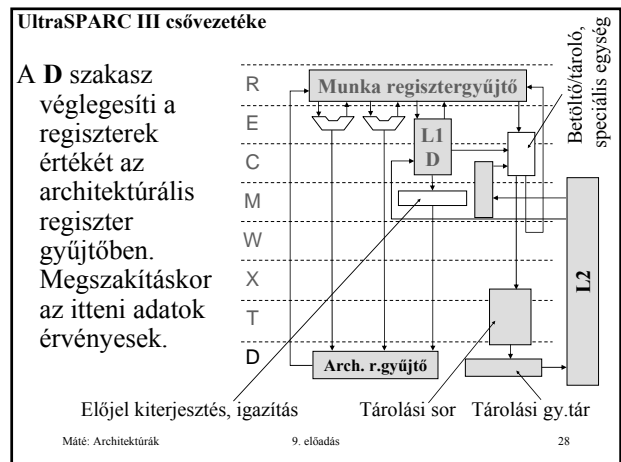
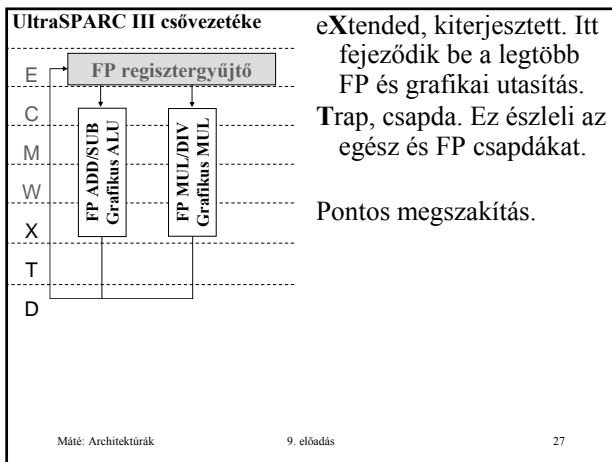
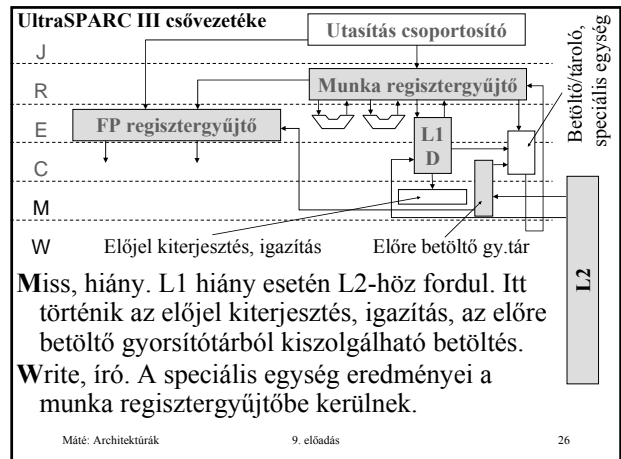
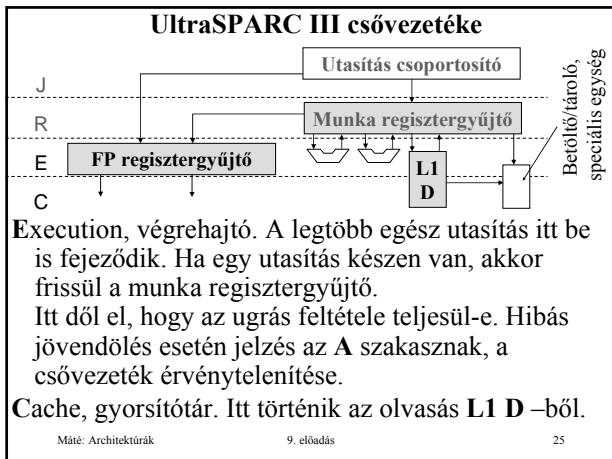
**Instruction group formation**, utasítás csoportosító. Aszerint csoportosítja az utasításokat, hogy melyik működési egységet használják.

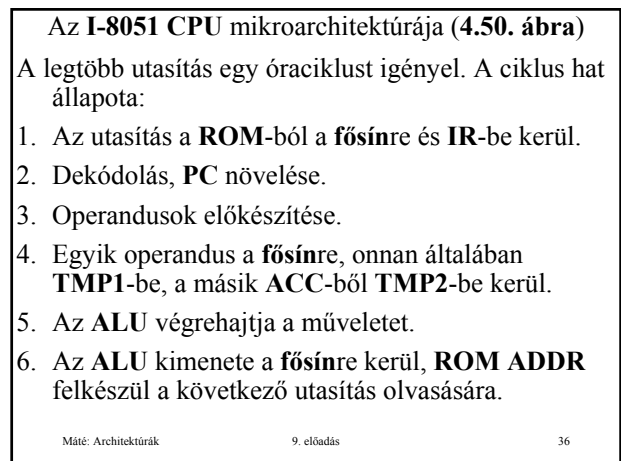
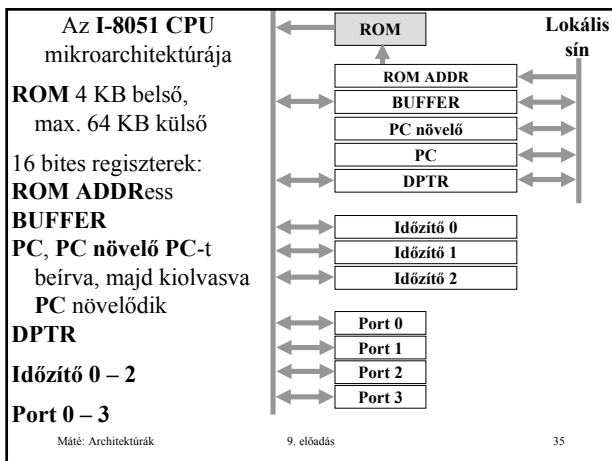
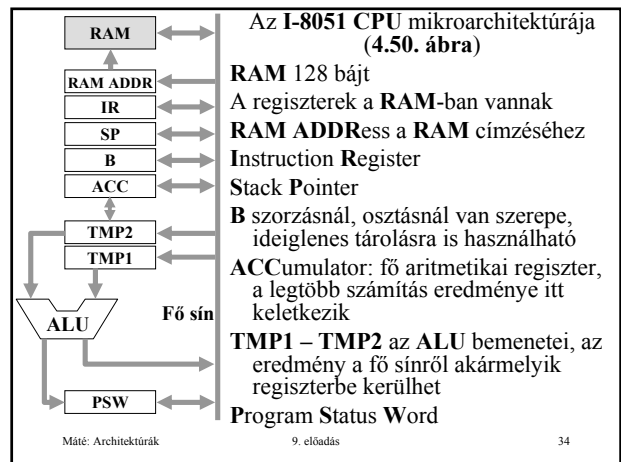
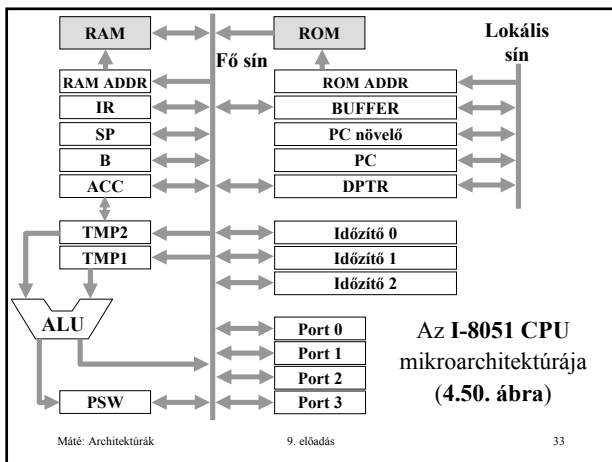
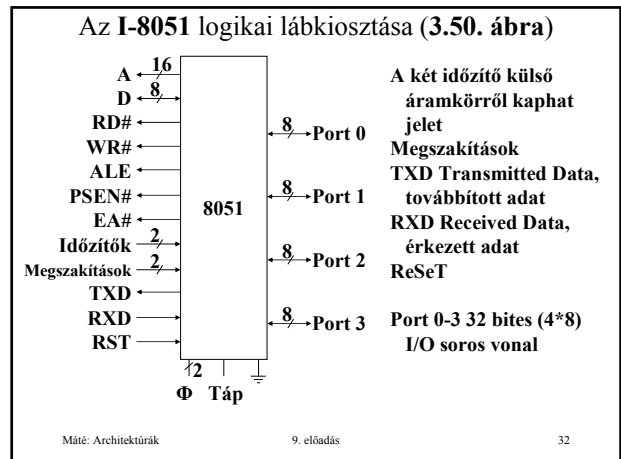
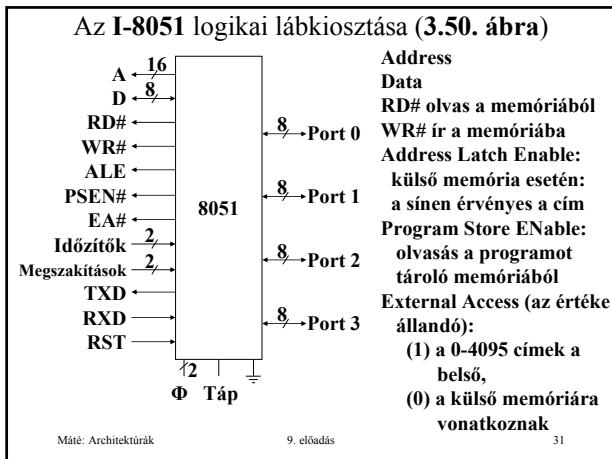
Máté: Architektúrák 9. előadás 23

**UltraSPARC III csővezetéke**

**J** instruction stage grouping, utasítás kiosztó. Az elérhető működési egységektől függően akár 4 utasítást is továbbít az R szakasznak.  
**Register**, függőség esetén vár, nincs sorrenden kívüli végrehajtás.

Máté: Architektúrák 9. előadás 24





### Összehasonlítás

**Pentium 4** CISC gép  
egy CISC utasítás → több RISC mikROUTASÍTÁS

**UltraSPARC III** RISC gép

**I-8051** inkább RISC, mint CISC gép

**picoJava II** verem gép, sok memória hivatkozás  
több CISC utasítás → egy RISC mikROUTASÍTÁS

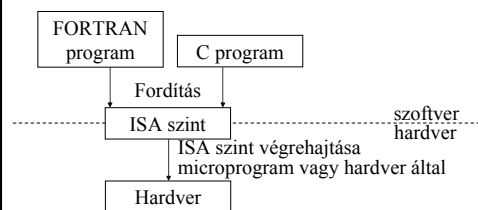
Máté: Architektúrák

9. előadás

37

### Utasításrendszer-architektúra szintje (ISA)

Amit a fordító program készítőjének tudnia kell:  
memóriamodell, regiszterek, adattípusok, utasítások.  
A hardver és szoftver között helyezkedik el, **5.1 ábra**.



Általában a mikroarchitektúra nem tartozik hozzá.

Máté: Architektúrák

9. előadás

38

### Az ISA szint tervezési szempontjai

- **Hosszú távú:** később is jó legyen az architektúra,  
**Rövid távú:** addig is piacon kell maradni.
- **Rövidebb utasítások:** kevesebb helyet foglalnak el,  
gyorsabban betölthetők.  
**Hosszabb utasítások:** több lehetséges műveleti kód,  
nagyobb memória címezhető.
- **Bájt címzés:** hatékonyabb szöveg feldolgozásnál,  
**Szó címzés:** nagyobb memória címezhető.
- ...

Máté: Architektúrák

9. előadás

39

### Utasítások szintje (ISA)

A jóság két kritériuma:

- hatékony hardver megvalósítási lehetőség,
- jó médium a fordítóknak.

Továbbfejlesztéseknél ügyelni kell a kompatibilitásra!

Nyilvános definíció:

van: **SPARC, JVM** (tervezők);  
nincs: **Pentium 4** (gyártók).

kernelmód ↔ (user) felhasználói mód

Máté: Architektúrák

9. előadás

40

### Memória modellek

Néha (pl. **8051**) külön memória az adatoknak és az utasításoknak (Harvard-architektúra, nem ugyanaz, mint az osztott gyorsítótár!).

Máté: Architektúrák

9. előadás

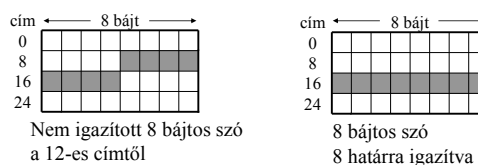
41

### Memória modellek

ASCII kód 7 bit + paritás → **Byte (bájt)**

**Szó:** 4 vagy 8 byte.

Igazítás (alignment), **5.2. ábra:** hatékonyabb, de probléma a kompatibilitás (a **Pentium 4**-nek két ciklusra is szüksége lehet egy szó beolvasásához).



Máté: Architektúrák

9. előadás

42

### Memória modellek

**Memória szemantika:** **STORE A** -t közvetlenül követő **LOAD A** mit ad vissza?

**A memória műveletek végrehajtása:**

- kötött sorrendben,
- definiálatlan sorrendben (ez a trend, mert hardver szinten egyszerűbb és gyorsabb). A hardver segítséget nyújthat:
  - **SYNC** utasítás: befejeztet minden megkezdett memória műveletet,
  - függőség esetén a hardver vár.

Máté: Architektúrák

9. előadás

43

### Regiszterek

**ISA**-szinten a mikroszint nem minden regisztere látszik (**TOS**, **MAR**), de van közös is (**PC**, **SP**).

**Speciális regiszterek:** **PC**, **SP**, ...

**Általános célú regiszterek:** a gyakran használt adatok gyors elérésére.

Jó, ha szimmetrikusak: fordítók, konvenciók.

**RISC** gépen általában legalább 32 általános célú.

**Kernelmódban továbbiak:** gyorsítótár vezérlés, memória védelem, ...

**PSW** (Program Status Word): az eredmény negatív, nulla, ... mód, prioritásszint, megszakítás-állapot, ...

Máté: Architektúrák

9. előadás

44

### Utasításkészlet

**LOAD**, **STORE**,

**MOVE**, aritmetikai, logikai,

feltétlen, feltételes elágazó utasítások,

...

Máté: Architektúrák

9. előadás

45

### Pentium 4

Nagyon sok előd (kompatibilitás!), a fontosabbak:

- **4004**: 4 bites,
- **8086**, **8088**: 16 bites, 8 bites adat sín.
- **80286**: 24 bites (nem lineáris) címtartomány (16 K darab 64 KB-os szegmens).
- **80386**: **IA-32** architektúra, az Intel első 32 bites gépe, lényegében az összes későbbi is ezt használja.
- **Pentium II** –től **MMX** utasítások.

Máté: Architektúrák

9. előadás

46

### A Pentium 4 üzemmódjai

**real** (valós): az összes **8088** utáni fejlesztést kikapcsolja (valódi **8088**-ként viselkedik). Hibánál a gép egyszerűen összeomlik, lefagy.

**virtuális 8086**: a **8088**-as programok védett módban futnak (ha **WINDOWS**-ból indítjuk az **MS-DOS**-t, és ha abban hiba történik, akkor nem fagy le, hanem visszaadja a vezérlést a **WINDOWS**-nak).

**védett**: valódi **Pentium 4**. 4 védelmi szint (**PSW**):

**0**: kernelmód (operációs r.), **1**, **2**: ritkán használt, **3**: felhasználói mód.

Máté: Architektúrák

9. előadás

47

### Memóriaszervezés:

- **16 K db szegmens** lehetséges, de a **WINDOWS**-ok és **UNIX** is csak **1** szegmenst támogatnak, és ennek is egy részét az operációs rendszer foglalja el,
- minden szegmensen belül a címtartomány: **0 - 2<sup>32</sup>-1**
- **Little endian** tárolási mód: az alacsonyabb címen van az alacsonyabb helyértékű bájt.

Máté: Architektúrák

9. előadás

48



**Regiszterek (5.3. ábra):**

- (majdnem) általános regiszterek:

<b>EAX</b>		AH AX AL	<b>Accumulator</b>
<b>EBX</b>		BH BX BL	<b>Base index</b>
<b>ECX</b>		CH CX CL	<b>Count</b>
<b>EDX</b>		DH DX DL	<b>Data</b>

Ezek 8 és 16 bites részei önálló regiszterként használhatók.

Máté: Architektúrák 9. előadás 49

**Regiszterek (5.3. ábra):**

- **ESI, EDI** (mutatók tárolására, szöveg kezelésre),
- **EBP** (keretmutató, verem kezelésre),
- **ESP** (verem mutató),
- **EIP** (utasítás számláló),
- **EFLAGS (PSW)**,
- **CS, SS, DS, ES, FS, GS** (16 bites regiszterek. A kompatibilitást biztosítják a régebbi gépekkel. Mivel a **Windows, Unix** csak egy címtartományt használ, ezekre csak a visszafelé kompatibilitás miatt van szükség).

Máté: Architektúrák 9. előadás 50

**Feladatok**

Mire szolgál a lefoglaló/átnevező egység?  
 Mire szolgálnak a regiszter gyűjtők?  
 Milyen sorrendben fejeződik be az utasítások végrehajtása a Pentium 4-en?  
 Mi a különbség a Pentium 4 két egész aritmetikájú **ALU**-ja között?  
 Miért nem írható azonnal az eredmény L2-be?  
 Mit jelent a pontos megszakítás kifejezés?  
 Milyen problémát okozhat a tárolás utáni betöltés?

Máté: Architektúrák 9. előadás 51

**Feladatok**

Hogy működik az UltraSPARC III másodlagos gyorsítótára?  
 Mire szolgál az UPA (Ultra Port Architecture)?  
 Mire szolgál az UDB II (UltraSPARC Data Buffer II)?  
 Milyen szervezésű az UltraSPARC III **L1 I** gyorsítótára?  
 Mire szolgál a munka regisztergyűjtő?  
 Mire szolgál az architektúrális regisztergyűjtő?  
 Mire szolgál az előre betöltő gyorsítótár?  
 Mire szolgál a tárolási sor?  
 Mire szolgál a tárolási gyorsítótár?

Máté: Architektúrák 9. előadás 52

**Feladatok**

Mire szolgál az UltraSPARC III ugrótáblája?  
 Milyen elágazás jövedőlést használ az UltraSPARC III?  
 Mit nevezünk eltolás résznek?  
 Hogy kezeli az UltraSPARC III az eltolás részt?  
 Mire szolgál az utasítás csoportosító egység?  
 Mire szolgál a munka regisztergyűjtő?  
 Mire szolgál az architektúrális regisztergyűjtő?  
 Hány ALU van az UltraSPARC III-ban?  
 Hogy kezeli az UltraSPARC III a függőségeket?

Máté: Architektúrák 9. előadás 53

**Feladatok**

Mi az **I-8051** fő alkalmazási területe?  
 Nagyságrendileg milyen áru egy **I-8051**?  
 Jellemezze az **I-8051**-et!  
 Mi a **RAM**?  
 Mi a **ROM**?  
 Hány bites a **RAM ADDR** regiszter?  
 Hány bites a **ROM ADDR** regiszter?  
 Mekkora az **I-8051 RAM**-ja?  
 Mekkora az **I-8051 ROM**-ja?  
 Mire szolgál az **IR, SP, B, ACC, TMP1-2** regiszter?  
 Mi a **PSW**?  
 Hogy történik **PC** növelése?

Máté: Architektúrák 9. előadás 54

### Feladatok

Milyen és hány be/kimenete van az **I-8051**-nek?  
Mire használhatók az **I-8051** be/kimenetei?  
Hány időzítője van az **I-8051**-nek?  
Mire használhatók az **I-8051** időzítői?  
Mik az **I-8051** ALU-jának bemenetei?  
Milyen állapotai vannak az óraciklusának?  
Jellemezze a **CISC** gépeket!  
Jellemezze a **RISC** gépeket!  
**CISC** vagy **RISC** gép a **Pentium 4**?  
**CISC** vagy **RISC** gép az **UltraSPARC III**?  
**CISC** vagy **RISC** gép az **I-8051**?  
Hasonlítsa össze a **Pentium 4**-et, az **UltraSPARC III**-at és az **I-8051**-ez!

Máté: Architektúrák

9. előadás

55

### Feladatok

Miért kitüntetett szint a gépi utasítások szintje (**ISA**)?  
Mikor jó egy gép **ISA** szintje?  
Mi a különbség a felhasználói (user) és a kernel mód között?  
Mit jelent az igazítás 4 bájtos szavak tárolásánál?  
Mi az igazítás előnye?  
Mit jelent a memória szemantika?  
Milyen hardver megoldásokat ismer a memória műveletek végrehajtási sorrendjére vonatkozóan?  
Mi a **SYNC** utasítás hatása?  
Miért van szükség a **SYNC** utasításra?

Máté: Architektúrák

9. előadás

56

### Feladatok

Mondjon olyan regisztert, amely a mikro utasítások szintjén és **ISA** szinten is látszik!  
Mondjon olyan regisztert, amely csak a mikro utasítások szintjén látszik!  
Mondjon olyan regisztert, amely csak kernel módban érhető el!  
Milyen utasítás típusokat ismer?  
Melyek a **Pentium 4** processzor legfontosabb elődjei?  
Milyen üzemmódjai vannak a **Pentium 4**-nek?  
Milyen a **Pentium 4** memória szervezése?  
Milyen regiszterei vannak a **Pentium 4**-nek?  
Mit jelent a **Little endian** tárolási mód?

Máté: Architektúrák

9. előadás

57

### Az előadáshoz kapcsolódó

#### Fontosabb tételek

A NetBurst csővezeték  
Az UltraSPARC III processzor és az UltraSPARC III mikroarchitektúrája, csővezetéke  
Az I-8051 processzor és az I-8051 mikroarchitektúrája  
A Pentium 4, az UltraSPARC III és az I-8051 mikroarchitektúrájának összehasonlítása  
Gépi utasítás szint. Memória modellek, memória szemantika  
A Pentium 4 ISA szintje

Máté: Architektúrák

9. előadás

58