

- Verem címzés (stack addressing):** Az operandus a verem tetején van. Nem kell operandust megadni az utasításban.

**Fordított Lengyel Jelölés**  
(Postfix Polish Notation - Lukasiewicz)

**Postfix jelölés:** a kifejezéseket olyan formában adjuk meg, hogy az első operandus után a másodikat, majd ezután adjuk meg a műveleti jelet:

**infix:**  $x + y$ ,                      **postfix:**  $x y +$ .

**Előnyei:** nem kell zárójel, sem precedencia szabályok. Jól alkalmazható verem címzés esetén.

Máté: Architektúrák                      10. előadás                      1

**Dijkstra algoritmusa**

**Infix jelölés konvertálása postfix-re (5.21, 22. ábra):**

- az **infix** elemek egy változóhoz (switch) érkeznek - a változók és konstansok Kaliforniába mennek (←),
- a többi esetben a verem tetejétől függően (5.22. ábra):

váltó

- a kocsit Texas felé megy (1: ↓),
- a verem teteje Kaliforniába megy (2: ↑),
- a kocsit eltűnik a verem tetejével együtt (3: ∅),
- vége az algoritmusnak (4: ●),
- hibás az **infix** formula (5: ?).

Máté: Architektúrák                      10. előadás                      2

Minden változó és konstans menjen Kaliforniába (←), a többi esetben a döntési tábla szerint járjunk el (5.21. ábra):

váltó

**A váltó előtti kocsi**

⊥	+	-	*	/	(	)	?
●	↓	↓	↓	↓	↓	↓	↓
↑	↑	↑	↑	↑	↑	↑	↑
↑	↑	↑	↑	↑	↑	↑	↑
↑	↑	↑	↑	↑	↑	↑	↑
↑	↑	↑	↑	↑	↑	↑	↑
(	?	↓	↓	↓	↓	↓	∅

**A döntési tábla tartalmazza a prioritási szabályokat!**

Máté: Architektúrák                      10. előadás                      3

▼ **A\*(B+C)⊥** ←

⊥

A ▼ **\*(B+C)⊥** ↓

⊥

A ▼ **(B+C)⊥** ↓

\*

⊥

A ▼ **B+C)⊥** ←

(

\*

⊥

AB ▼ **+C)⊥** ↓

(

\*

⊥

← változó Kaliforniába

**A váltó előtti kocsi**

⊥	+	-	*	/	(	)	?
●	↓	↓	↓	↓	↓	↓	↓
↑	↑	↑	↑	↑	↑	↑	↑
↑	↑	↑	↑	↑	↑	↑	↑
↑	↑	↑	↑	↑	↑	↑	↑
↑	↑	↑	↑	↑	↑	↑	↑
(	?	↓	↓	↓	↓	↓	∅

**A verem teteje**

Máté: Architektúrák                      10. előadás                      4

AB ▼ **C)⊥** ←

+

(

\*

⊥

ABC ▼ **)⊥** ↑

+

(

\*

⊥

ABC+ ▼ **)⊥** ∅

(

\*

⊥

**A váltó előtti kocsi**

⊥	+	-	*	/	(	)	?
●	↓	↓	↓	↓	↓	↓	↓
↑	↑	↑	↑	↑	↑	↑	↑
↑	↑	↑	↑	↑	↑	↑	↑
↑	↑	↑	↑	↑	↑	↑	↑
↑	↑	↑	↑	↑	↑	↑	↑
(	?	↓	↓	↓	↓	↓	∅

**A verem teteje**

ABC+ ▼ **⊥** ↑

\*

⊥

ABC+\* ▼ **⊥** ●

⊥

Máté: Architektúrák                      10. előadás                      5

**Fordított lengyel jelölésű formulák kiértékelése**

Pl. (5.24. ábra):

$(8 + 2 * 5) / (1 + 3 * 2 - 4)$

$8 \ 2 \ 5 \ * \ + \ 1 \ 3 \ 2 \ * \ + \ - \ /$

// infix

// postfix

Olvassuk a formulát balról jobbra!

Ha a következő jel

- operandus:** rakjuk a verembe,
- műveleti jel:** hajtsuk végre a műveletet (a verem tetején van a jobb, alatta a bal operandus!).

Máté: Architektúrák                      10. előadás                      6

(8 + 2 * 5)/(1 + 3 * 2 - 4) // infix			
Lépés	Maradék formula	Utasítás	Verem
1	8 2 5 * + 1 3 2 * + 4 - /	BIPUSH 8	8
2	2 5 * + 1 3 2 * + 4 - /	BIPUSH 2	8, 2
3	5 * + 1 3 2 * + 4 - /	BIPUSH 5	8, 2, 5
4	* + 1 3 2 * + 4 - /	IMUL	8, 10
5	+ 1 3 2 * + 4 - /	IADD	18
6	1 3 2 * + 4 - /	BIPUSH 1	18, 1
7	3 2 * + 4 - /	BIPUSH 3	18, 1, 3
8	2 * + 4 - /	BIPUSH 2	18, 1, 3, 2
9	* + 4 - /	IMUL	18, 1, 6
10	+ 4 - /	IADD	18, 7
11	4 - /	BIPUSH 4	18, 7, 4
12	- /	ISUB	18, 3
13	/	IDIV	6

Máté: Architektúrák 10. előadás 7

**Ortogonalitási elv:** Jó architektúrában a műveleti kódok és a címzési módszerek (majdnem) szabadon párosíthatók.

**Három címes elképzelés (5.25. ábra):**

1	Műv.kód	0	cél	forrás1	forrás2	Műv.kód
2	Műv.kód	1	cél	forrás1	eltolás	
3	Műv.kód	eltolás				

- típus: aritmetikai utasítások.
- típus: közvetlen adat megadás, index módú **LOAD** és **STORE** utasítás.
- típus: elágazó, eljárás hívó utasítások, **LOAD** és **STORE**, ezek **R0**-t használnák.

Máté: Architektúrák 10. előadás 8

**Két címes elképzelés (5.26. ábra).**

8	3	5	4	3	5	4
Műv.kód	mód	reg	eltolás	mód	reg	eltolás

Feltételesen: 32 bites direkt operandus vagy eltolás

Feltételesen: 32 bites direkt operandus vagy eltolás

A mód 3 bitje lehetővé teszi a közvetlen operandus, direkt, regiszter, regiszter indirekt, index és verem címzési módokat

Két további mód bevezetésére is lehetőség van.

Máté: Architektúrák 10. előadás 9

**Pentium 4 utasításformái (5.14. ábra)**

Több generáción keresztül kialakult architektúra. Csak egy operandus lehet memória cím.

**Prefix, escape** (bővítésre), **MOD, SIB** (Scale Index Base)

0-5	1-2	0-1	0-1	0-4	0-4	bájt
prefix	műv.kód	MOD	SIB	eltolás	közvetlen	

6 bit utasítás, 1 1 bit, 2 3 3 bit skála index bázis

Melyik operandus a forrás? 2 3 3 bit mód REG R/M

Máté: Architektúrák 11. előadás 10

**SIB (5.28. ábra):** jó, de megéri?

← EBP	2 3 3 bit skála index bázis
Egyéb lokális változók	
← EBP+8	Legyen <i>i</i> az EAX regiszterben
← EBP+12	SIB módú hivatkozás:
← EBP+16	M[4*EAX+EBP+8]

Máté: Architektúrák 11. előadás 11

**Címzési módok (5.27. ábra):** nagyon szabálytalan. **Baj:** nem minden utasításban használható minden mód, nem minden regiszter használható minden módban (nincs **EBP** indirekt, **ESP** relatív címzés). 32 bites címzési módok:

	MÓD			
R/M	00	01	10	11
000	M[EAX]	M[EAX+offset8]	M[EAX+offset32]	EAX v. AL
001	M[ECX]	M[ECX+offset8]	M[ECX+offset32]	ECX v. CL
010	M[EDX]	M[EDX+offset8]	M[EDX+offset32]	EDX v. DL
011	M[EBX]	M[EBX+offset8]	M[EBX+offset32]	EBX v. BL
100	SIB	SIB offset8-cal	SIB offset32-vel	ESP v. AH
101	direkt	M[EBP+offset8]	M[EBP+offset32]	EBP v. CH
110	M[ESI]	M[ESI+offset8]	M[ESI+offset32]	ESI v. DH
111	M[EDI]	M[EDI+offset8]	M[EDI+offset32]	EDI v. BH

Máté: Architektúrák 11. előadás 12

### UltraSPARC utasításformái (5.15. ábra)

32 bites egyszerű utasítások.

Forma 

2	5	6	5	1	8	5
---	---	---	---	---	---	---

1a 

m.k.	cél	m.k.	forrás1	0	FP-m.k.	forrás2
------	-----	------	---------	---	---------	---------

 3 címes

1b 

m.k.	cél	m.k.	forrás1	1	közvetlen konst.
------	-----	------	---------	---	------------------

 2 címes

**Aritmetikai utasítások:**  
 1 cél és 2 forrás regiszter vagy  
 1 cél, 1 forrás regiszter és 1 közvetlen konstans.

**LOAD, STORE** (csak ezek használják a memóriát):  
 a cím két regiszter összege vagy  
 egy regiszter + 13 bites eltolás.

**Processzorokat szinkronizáló utasítás.**

Máté: Architektúrák 11. előadás 13

Forma 

2	5	3	22
---	---	---	----

m.k.	cél	m.k.	közvetlen konstans
------	-----	------	--------------------

**SETHI**

32 bites közvetlen adat megadása: **SETHI** – megad 22 bitet, a következő utasítás a maradék 10 bitet.

Forma 

2	1	4	3	3	22 (19)
---	---	---	---	---	---------

m.k.	A	felt	m.k.	PC relatív cím
------	---	------	------	----------------

**UGRÁS**

Az ugrások **PC**-relatívok, szót (4-gyel osztható bajt címet) címeznek. Jósló utasításokhoz 3 bitet elcsíptek. Az **A** bit az eltolás rést akadályozza meg bizonyos feltételek esetén.

Forma 

2	30
---	----

m.k.	PC relatív cím
------	----------------

**CALL**

Eljárás hívás: 30 bites **PC**-relatív (szó) cím

Máté: Architektúrák 11. előadás 14

### UltraSPARC címzési módjai

Memóriára hivatkozó utasítások:  
 betöltő, tároló, multiprocesszor szinkronizáló  
 bázis-index (1a),  
 index + 13 bit eltolás (1b).

A többi utasítás általában 5 bites regiszter címzést használ

Máté: Architektúrák 11. előadás 15

### A 8051 utasításformátumai

- 1 

Műv.kód
---------

  
Implicit regiszter általában **ACC**, ...pl. **ACC** növelő
- 2 

Műv.kód	R
---------	---

**R 3 bites regisztercím**  
Regiszter és **ACC** tartalmán végzett művelet, mozgatás, ...
- 3 

Műv.kód	Operandus
---------	-----------

  
Operandus: közvetlen, eltolás, bitsorszám
- 4 

Műv.kód	11 bites cím
---------	--------------

  
Ugrás, szubrutin (eljárás) hívás (cím < 2048)
- 5 

Műv.kód	16 bites cím
---------	--------------

  
Ugrás, szubrutin (eljárás) hívás
- 6 

Műv.kód	Operandus1	Operandus2
---------	------------	------------

  
Pl. közvetlen operandus memóriába töltése, ...

Máté: Architektúrák 11. előadás 16

### A 8051 címzési módjai

Implicit: **ACC**  
 Regiszter: akár forrás, akár cél operandus lehet  
 Direkt: 8 bites memóriacím  
 Regiszter-indirekt:  
 8 bites memóriacím,  
 indirekt címzés a 16 bites **DPTR**-rel vagy **PC**-vel  
 Közvetlen operandus:  
 általában 8 bites, de  
 11 ill. 16 bites abszolút cím ugráshoz, eljárás híváshoz

Máté: Architektúrák 11. előadás 17

### Összefoglaló: 5.29. ábra.

Címzési mód	Pentium 4	UltraSPARC III	8051
Akkumulátor			X
Közvetlen	X	X	X
Direkt	X		X
Regiszter	X	X	X
Regiszter indirekt	X		X
Index	X	X	
Bázis-index	X	X	
Verem			

A bonyolult címzési módok tömörebb programokat tesznek lehetővé, de nehezítik a párhuzamosítást. Ha a párosítás nem történhet szabadon, akkor jobb, ha csak egy választási lehetőség van (egyszerűbb hatékony fordítóprogramot írni).

Máté: Architektúrák 11. előadás 18

### Utastástípusok

- Adatmozgató (másoló) utastások.
- Diadikus: +, -, \*, /, **AND**, **OR**, ~~NOT~~, **XOR**, ...
- Monadikus: léptetés, forgatás, **CLR**, **INC**, **NEG**, ...
- Összehasonlítás, feltételes elágazás: **Z**, **O**, **C**, ...
- Eljáráshívás. Visszatérési cím:
  - rögzített helyre (rossz),
  - az eljárás első szavába (jobb),
  - verembe (rekurzív eljárásokhoz is jó).
- Ciklusszervezés (**5.30. ábra**): számláló
- Input/output (**5.31-33. ábra**):
  - programozott **I/O**: tevékeny várakozás, **5.32. ábra**,
  - megszakítás vezérelt **I/O**,
  - **DMA I/O** (**5.33. ábra**): cikluslopás.

Máté: Architektúrák

11. előadás

19

### Ciklusszervezés (5.30. ábra)

<pre> i=1; L1: első utastás . . . utolsó utastás i = i + 1; if(i ≤ n) goto L1;                 </pre> <p><i>Végfeltételes ismétlés</i></p>	<pre> i=1; L1: if(i &gt; n) goto L2;     első utastás . . . utolsó utastás i = i + 1;     goto L1; L2: ...                 </pre> <p><i>Kezdő feltételes ismétlés</i></p>
--	---

Máté: Architektúrák

11. előadás

20

### Feladatok

Hány címes utastások lehetségesek? Adjon mindegyikre példát!  
 Milyen címzési módokat ismer? Részletezze ezeket!  
 Mit jelent a fordított lengyel jelölés?  
 Milyen előnyei vannak a postfix jelölésnek?  
 Írja át postfix alakúra az alábbi formulákat!  
**A+B, A+B+C, A+B\*C, A\*B+C.**  
 Írja át infix alakúra az alábbi postfix formulákat!  
**AB+, AB-C/, AB\*C+, AB\*CD/E+-.**  
 Hogy működik Dijkstra algoritmus?  
 Hogy értékelhetők ki a postfix alakú formulák?

Máté: Architektúrák

10. előadás

21

### Feladatok

Mi az ortogonalitási elv?  
 Milyen utastás formájú 3 címes gépet tervezne?  
 Milyen utastás formájú 2 címes gépet tervezne? Teljesül-e az ortogonalitási elv a **Pentium 4-en**?  
 Milyen utastás formái vannak a **Pentium 4-nek**?  
 Mire szolgál a prefix bájtt a **Pentium 4-en**?  
 Mire szolgál a címzési mód bájtt a **Pentium 4-en**?  
 Mire szolgál a **SIB** bájtt a **Pentium 4-en**?

Máté: Architektúrák

11. előadás

22

### Feladatok

Teljesül-e az ortogonalitási elv az **UltraSPARC III-on**?  
 Milyen utastás formái vannak az **UltraSPARC III-nak**?  
 Mi a különbség az **UltraSPARC III ADD, ADDC, ADDCC** és **ADDCCC** utastásai között?  
 Milyen formátumú **LOAD** utastásai vannak az **UltraSPARC III-nak**?  
 Hogy adható meg 32 bites közvetlen adat az **UltraSPARC III-on**?  
 Milyen formátumú **CALL** utastása van az **UltraSPARC III-nak**?

Máté: Architektúrák

11. előadás

23

### Feladatok

Teljesül-e az ortogonalitási elv a **8051-en**?  
 Milyen utastás formái vannak a **8051-nek**?  
 Milyen formátumú ugró utastásai vannak a **8051-nek**?  
 Hogy érhető el 256-nál magasabb memória cím a **8051-en**?  
 Milyen utastás típusokat ismer?  
 Mondjon diadikus/monadikus utastásokat!  
 Hogy néz ki a vég-/kezdőfeltételes ciklus?

Máté: Architektúrák

11. előadás

24

**Az előadáshoz kapcsolódó**

**Fontosabb tételek**

Veremcímzés. Fordított lengyel (postfix) jelölés.

Dijkstra algoritmus. Postfix alakú formulák kiértékelése.

A Pentium 4, az UltraSPARC III, I-8051 utasítás formátumai, címzési módjai

Utasítás típusok.

Ciklus szervezés.