

A számítógépek felépítése 4.b: A pipeline

Markó Tamás
PTE TTK, 2003

2003.07.28.

Markó Tamás, PTE TTK

1

A rádiótelefonokat kérem KIKAPCSOLNI!

2003.07.28.

Markó Tamás, PTE TTK

2

„Pipelining”

Az utasítás végrehajtásának részfázisokra bontása:

- független részfázisok, önálló erőforrásokkal
- az egyik fázis eredménye a következő induló adata

Dr. Istenes Zoltán, ELTE

4

A pipeline (csővonal, futószalag)

Soros feldolgozás:

UB	UD	OB	UV	ET	UB	UD	OB	UV	ET
----	----	----	----	----	----	----	----	----	----

n. utasítás n+1. utasítás

Pipeline feldolgozás:

n. utasítás	UB	UD	OB	UV	ET			
n+1. utasítás		UB	UD	OB	UV	ET		
n+2. utasítás			UB	UD	OB	UV	ET	

- Ideális esetben az N_{PL} lépcsős pipeline N_{PL} -szeres teljesítménynövekedést jelent.
- Problémák jelentkeznek:
 - ugrásoknál
 - adatfüggőségénél
 - busz-konfliktusoknál

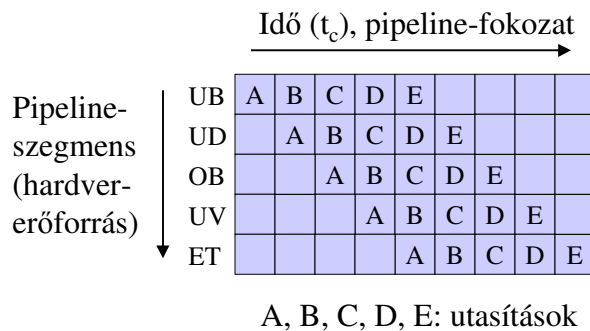
2003.07.28.

Markó Tamás, PTE TTK

5

A terhelési mátrix

- A működés személtetése a terhelési mátrixszal történik
- Az ábrán ötfokozatú, konfliktusmentes pipeline
- Fokozat: a parancs feldolgozásának egy állomása
- Szegmens: a szükséges hardver-erőforrás



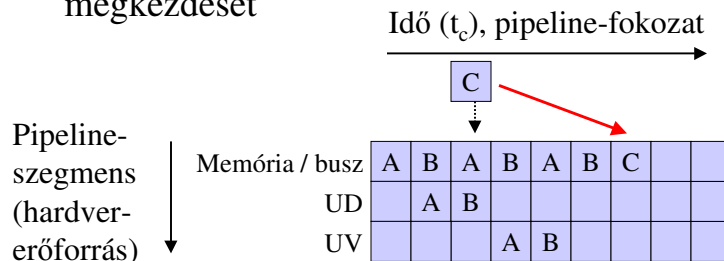
2003.07.28.

Markó Tamás, PTE TTK

6

A terhelési mátrix konfliktus esetén

- Ötlépcsős pipeline azzal a feltételezéssel, hogy az UB, OB és az ET tevékenységnél busz-hozzáférés van
- Az A és a B parancs buszhasználata négy órajelnyit késlelteti a C parancs végrehajtásának megkezdését



2003.07.28.

Markó Tamás, PTE TTK

7

A pipeline tervezési céljai 1.

Gondolatmenet:

- A hatékony pipeline-alkalmazás feltételei:
 - homogén utasításkészlet
 - kis számú, homogén címzési mód
 - ritka memória-hozzáférés
- Következmény:
 - Load/Store architektúra
 - sok regiszter
 - egyszerű parancsok (adott chip-felületnél)

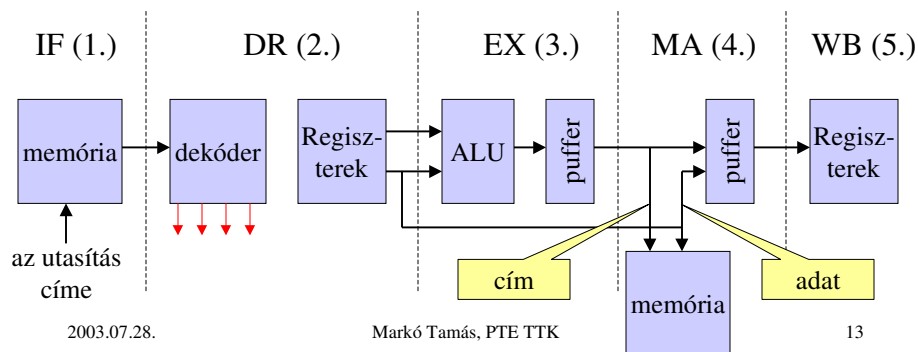
2003.07.28.

Markó Tamás, PTE TTK

8

Példa ötfokozatú pipeline-ra

- IF Instruction Fetch a parancs beolvasása
- DR Decode and Read dekódolás és a regiszter olvasása
- EX Execute végrehajtás (címkiszámításra is való)
- MA Memory Access memória-hozzáférés (elmaradhat)
- WB Write Back a regiszter írása



Az előző példa hardverterhelési mátrixa

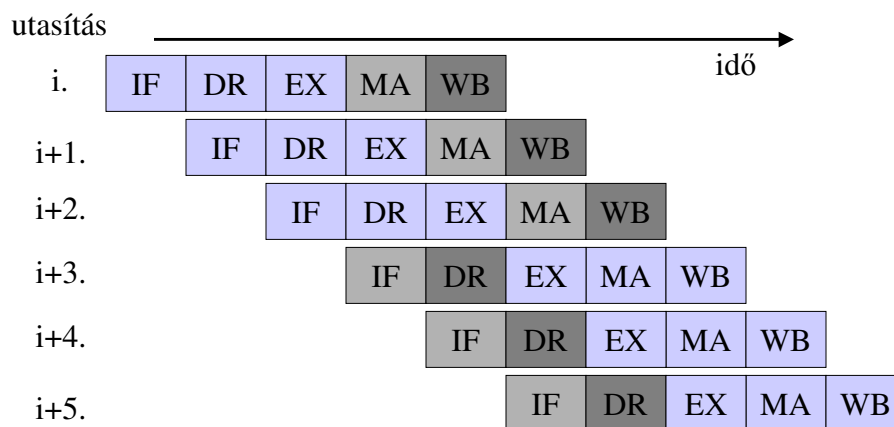
	Pipeline-fokozat				
	1. IF	2. DR	3. EX	4. MA	5. WB
memória	•			•	
dekóder		•			
regiszter		•			•
ALU			•		

2003.07.28.

Markó Tamás, PTE TTK

14

Működési diagramm konfliktusmentes esetben



A szürkével jelölt lépések konfliktusban vannak.

2003.07.28.

Markó Tamás, PTE TTK

15

Konfliktusok feloldása a pipeline-ban 1.

- A példa-pipeline nem lineáris, mivel a regiszterekre a 2. (DR) és az 5. (WB) fokozatban is szükség van.
- Megoldási lehetőség:
Mivel a regiszterekhez való hozzáférés egyszerű, egy félperiódus alatt is megvalósítható:
1. félperiódus írásra / 2. félperiódus olvasásra
- A konfliktus így megszűnik

	1.	2.	3.	4.	5.
	IF	DR	EX	MA	WB
memória	•			•	
dekóder		•			
regiszter		•			•
ALU			•		

2003.07.28.

Markó Tamás, PTE TTK

17

Konfliktusok feloldása a pipeline-ban 2.

- További konfliktus a példa-pipeline-ban:
a memóriára az 1. (IF) és a 4. (MA) fokozatban is szükség van.
- Megoldási lehetőség a **Harvard-architektúra**:
az utasításokat tároló memóriához és az adatokat tároló memóriához külön busz van (mindegyikben címbusz és adatbusz is).

	IF	DR	EX	MA	WB
utasítás-memória	•				
dekóder		•			
regiszter		•			•
ALU			•		
adat-memória				•	

2003.07.28.

Markó Tamás, PTE TTK

18

Adat-konfliktus

Példa:

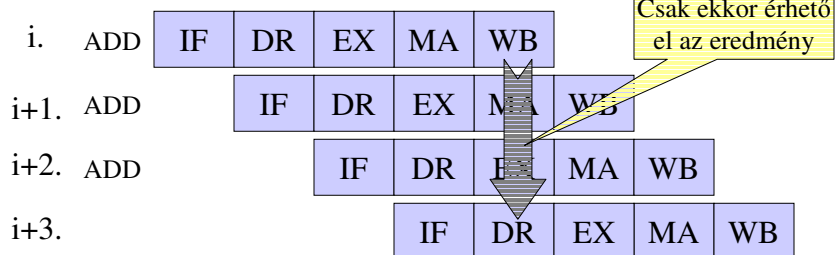
ADD $R_1 + R_2 \rightarrow R_4$

ADD $R_4 + R_3 \rightarrow R_4$

ADD $R_5 + R_0 \rightarrow R_6$

\Rightarrow az R_4 -es operandusz még nem áll rendelkezésre a 2. parancshoz

utasítás



2003.07.28.

Markó Tamás, PTE TTK

21

Megoldás az adat-konfliktusra 1.

a) Szoftverrel:

A fordítóprogram felismeri az adat-konfliktust és feloldja:

- NOP utasítás beszúrásával (mindig működik)

ADD $R_1 + R_2 \rightarrow R_4$

NOP

NOP

ADD $R_4 + R_3 \rightarrow R_4$

ADD $R_5 + R_0 \rightarrow R_6$

- az utasítások sorrendjének megváltoztatásával (jobb)

ADD $R_1 + R_2 \rightarrow R_4$

ADD $R_5 + R_0 \rightarrow R_6$

NOP

ADD $R_4 + R_3 \rightarrow R_4$

2003.07.28.

Markó Tamás, PTE TTK

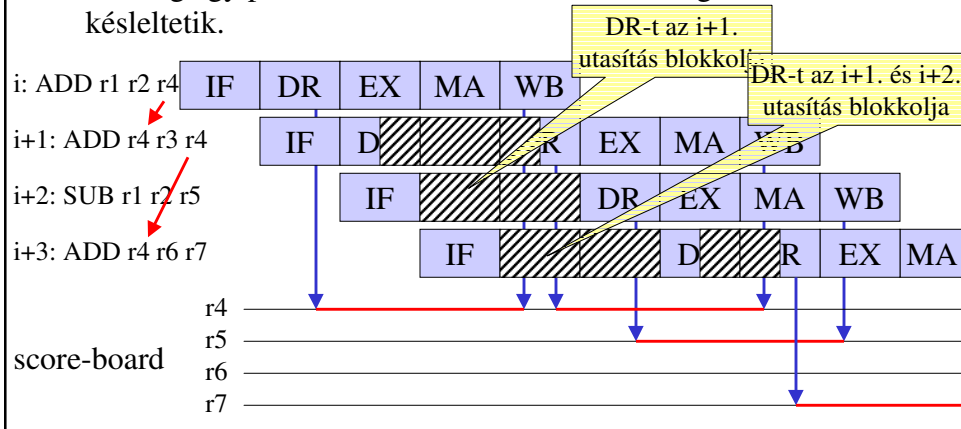
22

Megoldás az adat-konfliktusra 2.

b) Hardverrel: scoreboarding

Minden regiszterre nyilvántartják, hogy az a parancs, ami írni akar bele, a pipeline-ban van-e.

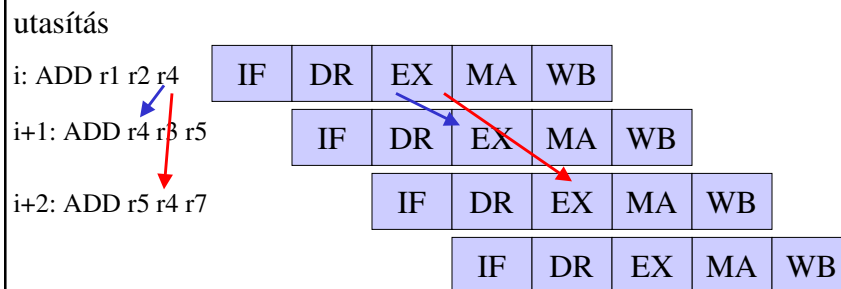
Ha még egy parancs hozzá akar férni ehhez a regiszterhez, azt késleltetik.



Megoldás az adat-konfliktusra 3.

c) Hardverrel: átadás (forwarding, bypass)

Az eredmények valójában nem csak az ötödik fokozat (WB) után állnak rendelkezésre, hanem már a harmadik fokozat (EX) után. \Rightarrow Lehetőség van rá, hogy már a következő periódus EX fokozatának rendelkezésére bocsássák.

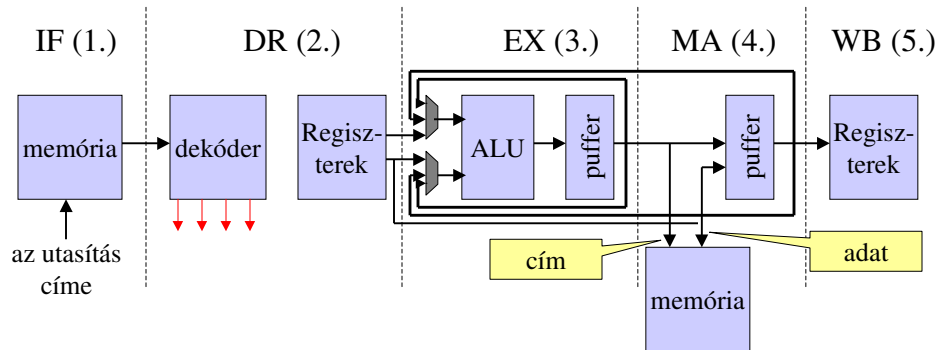


Az átadás (forwarding) ára

Újabb összeköttetések kellenek az alkatrészek között

A példa-pipeline kibővítése az átadáshoz:

- A multiplexerek a scoreboarding vezérli



2003.07.28.

Markó Tamás, PTE TTK

25

A lehetőségek összehasonlítása

- **Szoftveres megoldás:**
 - minden problémát a fordítóprogramra hárít
 - egyszerű fordító \Rightarrow rossz minőségű kód (NOP)
 - jó minőségű kód \Rightarrow lassabb a fordítóprogram
- **Forwarding:**
 - példa: SPARC, MIPS, stb.
 - elegáns megoldás, késleltetés-mentes
 - plusz adatutak + scoreboard \Rightarrow drága
 - nehéz a kiterjesztése a koprocesszorokra
- **Scoreboard:**
 - példa: Motorola MC 88 K, Intel i 860
 - könnyen kiterjeszthető a koprocesszorokra
 - a pipeline inhomogén lesz, plusz periódusok

2003.07.28.

Markó Tamás, PTE TTK

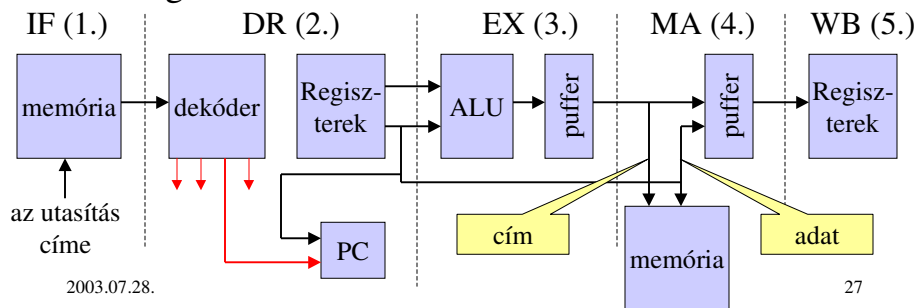
26

Vezérlési konfliktusok

- Felléphetnek:

- ugrásoknál (beleértve a szubrutinhívásokat is)
- megszakításoknál (interrupt)

Eddig nem vettük figyelembe a pipeline szerkezeténél az utasításszámlálót, most azonban szükség van rá



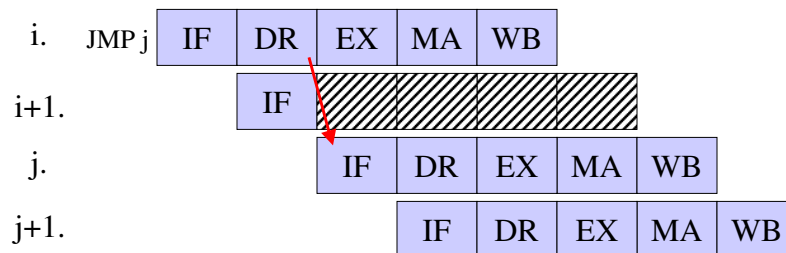
2003.07.28.

27

A vezérlési konfliktus hatása

- A DR szakasz végén derül ki, hogy ugrás következik
- A következő parancsot már beolvastuk, érvényteleníteni kell

utasítás



2003.07.28.

Markó Tamás, PTE TTK

28

A vezérlési konfliktusok szoftveres megoldása

- Az ugró utasítás utáni utasítást mindig végrehajtjuk (*delayed branching*).
- A fordítóprogram értelmes utasításokat szűr be ide (ez a feltételes ugrások kb. 70%-ának esetében sikerül, a feltétlen ugrásoknál mindig).

2003.07.28.

Markó Tamás, PTE TTK

31

Pipeline - összefoglalás

- Egyszerű elv
- Bonyolult, „trükkös” megvalósítás
- Sokat segíthet a jó fordítóprogram

Dr. Istenes Zoltán, ELTE

32