



HÁLÓZATI RENDSZEREK
ÉS SZOLGÁLTATÁSOK
TANSZÉK



Budapest,
2023.05.16.

SZÁMÍTÓGÉP ARCHITEKTÚRÁK

A soron kívüli utasításvégrehajtás implementációja

Horváth Gábor, Belső Zoltán

BME Hálózati Rendszerek és Szolgáltatások Tanszék
ghorvath@hit.bme.hu, belso@hit.bme.hu

- Alapelemek:
 - Utasítástároló
 - Dinamikus ütemező
 - Regiszterátnevezés
 - Sorrendvisszaállító buffer
- Nem minden megvalósítás használ minden elemet:
 - Scoreboard (1964): se regiszterátnevezés, se ROB
 - Tomasulo (IBM, 1967): van regiszterátnevezés, ROB nincs
 - Intel P6 óta (Core i7 is): mindegyiket használja
 - Mobil eszközökben is: ARM Cortex A9 óta



Scoreboard

- Utasítások végrehajtása a CDC 6600-ban 5 fázisból áll:

- IF: lehívás
- DS: dekódolás & műveleti egységhez rendelés
- IS: forrásoperandusok kiolvasása
- EX: végrehajtás
- WB: regiszter visszaírás

- Egymásrahatások kezelése:

- WAW: nem engedjük belépni a DS-be

D1 ← D2 / D3; ... ; D1 ← MEM [R0]

- RAW: nem engedjük belépni az IS-be

D1 ← D2 / D3; ... ; D5 ← D1 + D0

- WAR: nem engedjük belépni a WB-be

D5 ← D1 / D3; ... ; D1 ← MEM [R0]

- A CDC 6600 Scoreboard-ot használ
 - Az utasítások végrehajtási állapotának
 - A műveleti egységek foglaltsági állapotának
...nyilvántartására.
- Táblázatok:
 - **Utasítás állapot tábla:** az utasítás melyik fázisban tart
 - **Regiszter állapot tábla:** a regisztert melyik utasítás szándékozik megváltoztatni
 - **Műveleti egység tábla:** minden műveleti egységre tárolja:
 - Foglalt?
 - Mit kell csinálni?
 - Mi a két forrás operandus értéke?
 - Ha valamelyik még nincs kész, melyik művelet eredménye lesz?
 - Hova kell az eredményt írni?

- Példa:
 - 5 műveleti egység:
 - 1 ALU
 - 1 Load
 - 1 Store
 - 2 Lebegőpontos aritmetikai
 - Késleltetés: 3 ciklus, iterációs idő: 3 ciklus

Ciklus:

Utasítás állapot tábla					
Utasítás	DS	IS	EX	WB	
i1: D0 $\leftarrow$ MEM [R1]					
i2: D4 $\leftarrow$ D0 * D2					
i3: MEM [R1+16] $\leftarrow$ D4					
i4: R1 $\leftarrow$ R1 + 8					
i5: D0 $\leftarrow$ MEM [R1]					
i6: D4 $\leftarrow$ D0 * D2					
i7: MEM [R1+16] $\leftarrow$ D4					

Regiszter állapot tábla	
Regiszter	Műveleti egys.
D0	
D2	
D4	
R1	

Műveleti egység állapot tábla							
Egység	Foglalt	Műv.	R	RA	RB	EA	EB
Egész ALU	Nem						
Load	Nem						
Store	Nem						
FP ALU 1	Nem						
FP ALU 2	Nem						

Ciklus: 1

Utasítás állapot tábla					
Utasítás		DS	IS	EX	WB
i1:	D0 ← MEM [R1]	c1			
i2:	D4 ← D0 * D2				
i3:	MEM [R1+16] ← D4				
i4:	R1 ← R1 + 8				
i5:	D0 ← MEM [R1]				
i6:	D4 ← D0 * D2				
i7:	MEM [R1+16] ← D4				

Regiszter állapot tábla	
Regiszter	Műveleti egys.
D0	Load
D2	
D4	
R1	

Műveleti egység állapot tábla							
Egység	Foglalt	Műv.	R	RA	RB	EA	EB
Egész ALU	Nem						
Load	Igen	load	D0	R1			
Store	Nem						
FP ALU 1	Nem						
FP ALU 2	Nem						

← **Műveleti egység foglalás**

Ciklus: 2

Utasítás állapot tábla					
	Utasítás	DS	IS	EX	WB
i1:	D0 ← MEM [R1]	c1	c2		
i2:	D4 ← D0 * D2	c2			
i3:	MEM [R1+16] ← D4				
i4:	R1 ← R1 + 8				
i5:	D0 ← MEM [R1]				
i6:	D4 ← D0 * D2				
i7:	MEM [R1+16] ← D4				

Regiszter állapot tábla	
Regiszter	Műveleti egys.
D0	Load
D2	
D4	FP1
R1	

Műveleti egység állapot tábla							
Egység	Foglalt	Műv.	R	RA	RB	EA	EB
Egész ALU	Nem						
Load	Igen	load	D0	R1			
Store	Nem						
FP ALU 1	Igen	*	D4		D2	Load	
FP ALU 2	Nem						

← **Műveleti egység foglálás**

Ciklus: 3

Utasítás állapot tábla					
	Utasítás	DS	IS	EX	WB
i1:	$D0 \leftarrow \text{MEM}[R1]$	c1	c2	c3	
i2:	$D4 \leftarrow D0 * D2$	c2			
i3:	$\text{MEM}[R1+16] \leftarrow D4$	c3			
i4:	$R1 \leftarrow R1 + 8$				
i5:	$D0 \leftarrow \text{MEM}[R1]$				
i6:	$D4 \leftarrow D0 * D2$				
i7:	$\text{MEM}[R1+16] \leftarrow D4$				

Regiszter állapot tábla	
Regiszter	Műveleti egys.
D0	Load
D2	
D4	FP1
R1	

Műveleti egység állapot tábla							
Egység	Foglalt	Műv.	R	RA	RB	EA	EB
Egész ALU	Nem						
Load	Igen	load	D0	R1			
Store	Igen	store		R1			FP1
FP ALU 1	Igen	*	D4		D2	Load	
FP ALU 2	Nem						

← Műveleti egység foglaltság

← RAW miatt áll

Ciklus: 4

Utasítás állapot tábla					
Utasítás	DS	IS	EX	WB	
i1: D0 ← MEM [R1]	c1	c2	c3	c4	
i2: D4 ← D0 * D2	c2	c4			
i3: MEM [R1+16] ← D4	c3				
i4: R1 ← R1 + 8	c4				
i5: D0 ← MEM [R1]					
i6: D4 ← D0 * D2					
i7: MEM [R1+16] ← D4					

Regiszter állapot tábla	
Regiszter	Műveleti egys.
D0	Load
D2	
D4	FP1
R1	Egész ALU

← i1 vége

Műveleti egység állapot tábla							
Egység	Foglalt	Műv.	R	RA	RB	EA	EB
Egész ALU	Igen	+	R1	R1			
Load	Nem						
Store	Igen	store		R1			FP1
FP ALU 1	Igen	*	D4	D0	D2	Load	
FP ALU 2	Nem						

← Műveleti egység foglalás

← Bejegyzés törlése

← D0 betöltve

Ciklus: 5

Utasítás állapot tábla					
Utasítás	DS	IS	EX	WB	
i1: D0 ← MEM [R1]	c1	c2	c3	c4	
i2: D4 ← D0 * D2	c2	c4	c5		
i3: MEM [R1+16] ← D4	c3				
i4: R1 ← R1 + 8	c4	c5			
i5: D0 ← MEM [R1]	c5				
i6: D4 ← D0 * D2					
i7: MEM [R1+16] ← D4					

Regiszter állapot tábla	
Regiszter	Műveleti egys.
D0	Load
D2	
D4	FP1
R1	Egész ALU

Műveleti egység állapot tábla							
Egység	Foglalt	Műv.	R	RA	RB	EA	EB
Egész ALU	Igen	+	R1	R1			
Load	Igen	load	D0			ALU	
Store	Igen	store		R1			FP1
FP ALU 1	Igen	*	D4	D0	D2		
FP ALU 2	Nem						

← **Műveleti egység foglalás**

Ciklus: 6

Utasítás állapot tábla					
Utasítás	DS	IS	EX	WB	
i1: D0 ← MEM [R1]	c1	c2	c3	c4	
i2: D4 ← D0 * D2	c2	c4	c5+		
i3: MEM [R1+16] ← D4	c3				
i4: R1 ← R1 + 8	c4	c5	c6		
i5: D0 ← MEM [R1]	c5				
i6: D4 ← D0 * D2					
i7: MEM [R1+16] ← D4					

Regiszter állapot tábla	
Regiszter	Műveleti egys.
D0	Load
D2	
D4	FP1
R1	Egész ALU

← WAW egymásrahatás i2-vel, DS szünet

Műveleti egység állapot tábla							
Egység	Foglalt	Műv.	R	RA	RB	EA	EB
Egész ALU	Igen	+	R1	R1			
Load	Igen	load	D0			ALU	
Store	Igen	store		R1			FP1
FP ALU 1	Igen	*	D4	D0	D2		
FP ALU 2	Nem						

Ciklus: 7

Utasítás állapot tábla					
Utasítás	DS	IS	EX	WB	
i1: D0 ← MEM [R1]	c1	c2	c3	c4	
i2: D4 ← D0 * D2	c2	c4	c5+		
i3: MEM [R1+16] ← D4	c3				
i4: R1 ← R1 + 8	c4	c5	c6		
i5: D0 ← MEM [R1]	c5				
i6: D4 ← D0 * D2					
i7: MEM [R1+16] ← D4					

Regiszter állapot tábla	
Regiszter	Műveleti egys.
D0	Load
D2	
D4	FP1
R1	Egész ALU

WAR egymásrahatás i3-mal, WB szünet

WAW egymásrahatás i2-vel, DS szünet

Műveleti egység állapot tábla							
Egység	Foglalt	Műv.	R	RA	RB	EA	EB
Egész ALU	Igen	+	R1	R1			
Load	Igen	load	D0			ALU	
Store	Igen	store		R1			FP1
FP ALU 1	Igen	*	D4	D0	D2		
FP ALU 2	Nem						

Ciklus: 8

Utasítás állapot tábla					
Utasítás	DS	IS	EX	WB	
i1: D0 ← MEM [R1]	c1	c2	c3	c4	
i2: D4 ← D0 * D2	c2	c4	c5+	c8	
i3: MEM [R1+16] ← D4	c3	c8			
i4: R1 ← R1 + 8	c4	c5	c6		
i5: D0 ← MEM [R1]	c5				
i6: D4 ← D0 * D2	c8				
i7: MEM [R1+16] ← D4					

Regiszter állapot tábla	
Regiszter	Műveleti egys.
D0	Load
D2	
D4	FP1 FP2
R1	Egész ALU

i2 vége

WAR egymásrahatás i3-mal, WB szünet

Műveleti egység állapot tábla							
Egység	Foglalt	Műv.	R	RA	RB	EA	EB
Egész ALU	Igen	+	R1	R1			
Load	Igen	load	D0			ALU	
Store	Igen	store		R1	D4		FP1
FP ALU 1	Nem						
FP ALU 2	Igen	*	D4		D2	Load	

D4 kész

Bejegyzés törlése

Műveleti egység foglалás

Ciklus: 9

Utasítás állapot tábla					
Utasítás	DS	IS	EX	WB	
i1: D0 ← MEM [R1]	c1	c2	c3	c4	
i2: D4 ← D0 * D2	c2	c4	c5+	c8	
i3: MEM [R1+16] ← D4	c3	c8	c9		
i4: R1 ← R1 + 8	c4	c5	c6	c9	
i5: D0 ← MEM [R1]	c5	c9			
i6: D4 ← D0 * D2	c8				
i7: MEM [R1+16] ← D4					

Regiszter állapot tábla	
Regiszter	Műveleti egys.
D0	Load
D2	
D4	FP2
R1	Egész ALU

i4 vége

Feldolgozási egymáshatás a Store egységre, DS szünet

Műveleti egység állapot tábla							
Egység	Foglalt	Műv.	R	RA	RB	EA	EB
Egész ALU	Nem						
Load	Igen	load	D0	R1		ALU	
Store	Igen	store		R1	D4		
FP ALU 1	Nem						
FP ALU 2	Igen	*	D4		D2	Load	

Bejegyzés törlése

R1 kész

Ciklus: 10

Utasítás állapot tábla					
Utasítás	DS	IS	EX	WB	
i1: D0 ← MEM [R1]	c1	c2	c3	c4	
i2: D4 ← D0 * D2	c2	c4	c5+	c8	
i3: MEM [R1+16] ← D4	c3	c8	c9	c10	
i4: R1 ← R1 + 8	c4	c5	c6	c9	
i5: D0 ← MEM [R1]	c5	c9	c10		
i6: D4 ← D0 * D2	c8				
i7: MEM [R1+16] ← D4	c10				

Regiszter állapot tábla	
Regiszter	Műveleti egys.
D0	Load
D2	
D4	FP2
R1	

← **WB és a tőle függő DS egyszerre mehet**

Műveleti egység állapot tábla							
Egység	Foglalt	Műv.	R	RA	RB	EA	EB
Egész ALU	Nem						
Load	Igen	load	D0	R1			
Store	Igen	store		R1	D4		FP2
FP ALU 1	Nem						
FP ALU 2	Igen	*	D4		D2	Load	

← **Bejegyzés törlése, új foglalás**

- Egyszerű
- Viszonylag hatékony
 - Magas szintű programok (FORTRAN): 1.7x gyorsabbak
 - Alacsony szintű, optimalizált programok: 2.5x gyorsabbak
- Egymásrahatások kezelése:
 - WAW: nem engedjük belépni a DS-be
 - RAW: nem engedjük belépni az IS-be
 - WAR: nem engedjük belépni a WB-be



A Tomasulo algoritmus

- 1967: IBM 360/91-es processzor
- Out-of-order algoritmust Robert Tomasulo tervezte
- A P6 architektúra ma is ennek egy változatát használja
- (pl. Intel Core i7)
- Miben jobb?
 - Egymásrahatások kezelésében
 - WAW, WAR feloldása: **regiszter átnevezés**sel
 - Feldolgozási egymásrahatások kiküszöbölése várakozó sorral:
utasítás tároló
 - Könnyen kiegészíthető, hogy a külvilág sorrendi végrehajtást lásson: **sorrendvisszaállító buffer**rel

- **Reorder Buffer** mezők:
 - V: utasítás eredménye (érték)
 - R: eredményregiszter neve, ide kell majd írni az eredményt
 - Addr: Store esetén erre a címre kell majd írni az eredményt
 - „Ready” bit: kész-e az utasítás végrehajtása
- **Reservation Station** mezők:
 - Egység: az utasítás által igényelt egység azonosítója
 - Művelet kódja (+, −, *, /, Load, Store, stb.)
 - T: ebbe a ROB bejegyzésbe kell majd írni az eredményt
 - RA, RB: a 2 forrásoperandus
 - EA, EB: ha nincs meg minden forrásoperandus, ebből a ROB bejegyzésből kell majd venni, ha kész lesz

- **Regiszter állapot tábla:**
 - Minden architekturális regiszterhez egy bejegyzés
 - Vagy a regiszter értéke
 - Vagy a ROB bejegyzés száma, ahol
 - Vagy már meg is van az értéke
 - Vagy még nincs meg az értéke, de majd ott jelenik meg
- **CDB: Common data bus**
 - Üzenetszórásos busz
 - Ha egy utasítás kész, ezen keresztül kürtöli világgá
 - Aki pont rá várt, így tudja meg, hogy már nem kell várnia

- Példa:
 - 5 műveleti egység:
 - 1 ALU
 - 1 Load
 - 1 Store
 - 2 Lebegőpontos aritmetikai
 - Késleltetés: 3 ciklus, iterációs idő: 3 ciklus
 - Centralizált, 3 bejegyzéses utasítástároló

Ciklus:

Sorrendvisszaállító buffer (ROB)							
	Utasítás	V	R	Addr	IS	EX	CM
i1:	D0 $\leftarrow$ MEM [R1]						
i2:	D4 $\leftarrow$ D0 * D2						
i3:	MEM [R1+16] $\leftarrow$ D4						
i4:	R1 $\leftarrow$ R1 + 8						
i5:	D0 $\leftarrow$ MEM [R1]						
i6:	D4 $\leftarrow$ D0 * D2						
i7:	MEM [R1+16] $\leftarrow$ D4						

Regiszter állapot tábla	
Regiszter	Érték/ROB bej.
D0	
D2	
D4	
R1	

CDB	
Érték (V)	ROB bej.

Utasítástároló (RS)							
Foglalt	Egység	Műv.	T	RA	RB	EA	EB
Nem							
Nem							
Nem							

Ciklus: 1

Sorrendvisszaállító buffer (ROB)							
	Utasítás	V	R	Addr	IS	EX	CM
i1:	D0 ← MEM [R1]		D0				
i2:	D4 ← D0 * D2						
i3:	MEM [R1+16] ← D4						
i4:	R1 ← R1 + 8						
i5:	D0 ← MEM [R1]						
i6:	D4 ← D0 * D2						
i7:	MEM [R1+16] ← D4						

Regiszter állapot tábla	
Regiszter	Érték/ROB bej.
D0	ROB#1
D2	
D4	
R1	

CDB	
Érték (V)	ROB bej.

A későbbi utasítások
D0-t ROB#1-ben keressék

Utasítástároló (RS)							
Foglalt	Egység	Műv.	T	RA	RB	EA	EB
Igen (i1)	Load	load	ROB#1	[R1]			
Nem							
Nem							

ROB és RS bejegyzések létrehozása

Ciklus: 2

Sorrendvisszaállító buffer (ROB)							
	Utasítás	V	R	Addr	IS	EX	CM
i1:	D0 ← MEM [R1]		D0		c2		
i2:	D4 ← D0 * D2		D4				
i3:	MEM [R1+16] ← D4						
i4:	R1 ← R1 + 8						
i5:	D0 ← MEM [R1]						
i6:	D4 ← D0 * D2						
i7:	MEM [R1+16] ← D4						

Regiszter állapot tábla	
Regiszter	Érték/ROB bej.
D0	ROB#1
D2	
D4	ROB#2
R1	

CDB	
Érték (V)	ROB bej.

A későbbi utasítások
D4-et ROB#2-ben keressék

ROB és RS bejegyzések létrehozása

Utasítástároló (RS)							
Foglalt	Egység	Műv.	T	RA	RB	EA	EB
Igen (i1)	Load	load	ROB#1	[R1]			
Igen (i2)	FP	*	ROB#2		[D2]	ROB#1	
Nem							

Első forrásoperandust a ROB#1 fogja adni

Ciklus: 3

Sorrendvisszaállító buffer (ROB)							
	Utasítás	V	R	Addr	IS	EX	CM
→ i1:	D0 ← MEM [R1]		D0		c2	c3	
i2:	D4 ← D0 * D2		D4				
→ i3:	MEM [R1+16] ← D4						
i4:	R1 ← R1 + 8						
i5:	D0 ← MEM [R1]						
i6:	D4 ← D0 * D2						
i7:	MEM [R1+16] ← D4						

Regiszter állapot tábla	
Regiszter	Érték/ROB bej.
D0	ROB#1
D2	
D4	ROB#2
R1	

CDB	
Érték (V)	ROB bej.

Utasítástároló (RS)							
Foglalt	Egység	Műv.	T	RA	RB	EA	EB
Nem							
Igen (i2)	FP	*	ROB#2		[D2]	ROB#1	
Igen (i3)	Store	store	ROB#3	[R1]			ROB#2

i1 EX fázisba lép, RS-ből törölve

ROB és RS bejegyzések létrehozása

TOMASULO ALGORITMUS A P6-BAN

Ciklus: 4

Sorrendvisszaállító buffer (ROB)							
	Utasítás	V	R	Addr	IS	EX	CM
i1:	D0 ← MEM [R1]	[D0]	D0		c2	c3	c4
i2:	D4 ← D0 * D2		D4		c4		
i3:	MEM [R1+16] ← D4						
i4:	R1 ← R1 + 8		R1				
i5:	D0 ← MEM [R1]						
i6:	D4 ← D0 * D2						
i7:	MEM [R1+16] ← D4						

Regiszter állapot tábla	
Regiszter	Érték/ROB bej.
D0	ROB#1
D2	
D4	ROB#2
R1	ROB#4

CDB	
Érték (V)	ROB bej.
[D0]	ROB#1

i1 kész, eredményt kihirdeti

i1 eredménye a ROB-ba kerül

Utasítástároló (RS)							
Foglalt	Egység	Műv.	T	RA	RB	EA	EB
Igen (i4)	Egész ALU	+	ROB#4	[R1]			
Igen (i2)	FP	*	ROB#2	CDB.V	[D2]	ROB#1	
Igen (i3)	Store	store	ROB#3	[R1]			ROB#2

i2 ROB#1-re várt, most megjött.
Ezzel végrehajthatóvá vált, IS-be lép.

Ciklus: 5

Sorrendvisszaállító buffer (ROB)							
	Utasítás	V	R	Addr	IS	EX	CM
→ i1:	D0 ← MEM [R1]	[D0]	D0		c2	c3	c4
i2:	D4 ← D0 * D2		D4		c4	c5	
i3:	MEM [R1+16] ← D4						
i4:	R1 ← R1 + 8		R1		c5		
→ i5:	D0 ← MEM [R1]		D0				
i6:	D4 ← D0 * D2						
i7:	MEM [R1+16] ← D4						

Regiszter állapot tábla	
Regiszter	Érték/ROB bej.
D0	ROB#5
D2	
D4	ROB#2
R1	ROB#4

CDB	
Érték (V)	ROB bej.

i1 kilépett, ROB bejegyzése érvényre juttatható.
DE: D0-t azóta más használja (ROB#5), ezért az eredményt nem kell beírni a regiszter tárolóba

Utasítástároló (RS)							
Foglalt	Egység	Műv.	T	RA	RB	EA	EB
Igen (i4)	Egész ALU	+	ROB#4	[R1]			
Igen (i5)	Load	load	ROB#5			ROB#4	
Igen (i3)	Store	store	ROB#3	[R1]			ROB#2

i2 EX fázisba lép, RS-ből törölve,
helyére jön az i5 bejegyzése

Ciklus: 6

Sorrendvisszaállító buffer (ROB)							
	Utasítás	V	R	Addr	IS	EX	CM
→	i1: D0 ← MEM [R1]	[D0]	D0		c2	c3	c4
	i2: D4 ← D0 * D2		D4		c4	c5+	
	i3: MEM [R1+16] ← D4						
	i4: R1 ← R1 + 8		R1		c5	c6	
	i5: D0 ← MEM [R1]		D0				
→	i6: D4 ← D0 * D2		D4				
	i7: MEM [R1+16] ← D4						

Regiszter állapot tábla	
Regiszter	Érték/ROB bej.
D0	ROB#5
D2	
D4	ROB#6
R1	ROB#4

CDB	
Érték (V)	ROB bej.

Utasítástároló (RS)							
Foglalt	Egység	Műv.	T	RA	RB	EA	EB
Igen (i6)	FP	*	ROB#6		[D2]	ROB#5	
Igen (i5)	Load	load	ROB#5			ROB#4	
Igen (i3)	Store	store	ROB#3	[R1]			ROB#2

i4 EX fázisba lép, RS-ből törölve,
helyére jön az i6 bejegyzése

Ciklus: 7

Sorrendvisszaállító buffer (ROB)							
	Utasítás	V	R	Addr	IS	EX	CM
i1:	D0 ← MEM [R1]	[D0]	D0		c2	c3	c4
i2:	D4 ← D0 * D2		D4		c4	c5+	
i3:	MEM [R1+16] ← D4						
i4:	R1 ← R1 + 8	[R1]	R1		c5	c6	c7
i5:	D0 ← MEM [R1]		D0		c7		
i6:	D4 ← D0 * D2		D4				
i7:	MEM [R1+16] ← D4						

Regiszter állapot tábla	
Regiszter	Érték/ROB bej.
D0	ROB#5
D2	
D4	ROB#6
R1	ROB#4

CDB	
Érték (V)	ROB bej.
[R1]	ROB#4

i4 kész, eredményt kihirdeti

i4 eredménye a ROB-ba kerül

Feldolgozási egymásrahatás:
nincs hely az RS-ben, nem hozható be

Utasítástároló (RS)							
Foglalt	Egység	Műv.	T	RA	RB	EA	EB
Igen (i6)	FP	*	ROB#6		[D2]	ROB#5	
Igen (i5)	Load	load	ROB#5	CDB.V		ROB#4	
Igen (i3)	Store	store	ROB#3	[R1]			ROB#2

i5 ROB#4-re várt, most megjött.
Ezzel végrehajthatóvá vált, IS-be lép.

Ciklus: 8

Sorrendvisszaállító buffer (ROB)							
	Utasítás	V	R	Addr	IS	EX	CM
i1:	D0 ← MEM [R1]	[D0]	D0		c2	c3	c4
i2:	D4 ← D0 * D2	[D4]	D4		c4	c5+	c8
i3:	MEM [R1+16] ← D4				c8		
i4:	R1 ← R1 + 8	[R1]	R1		c5	c6	c7
i5:	D0 ← MEM [R1]		D0		c7	c8	
i6:	D4 ← D0 * D2		D4				
i7:	MEM [R1+16] ← D4						

Regiszter állapot tábla	
Regiszter	Érték/ROB bej.
D0	ROB#5
D2	
D4	ROB#6
R1	ROB#4

CDB	
Érték (V)	ROB bej.
[D4]	ROB#2

Utasítástároló (RS)							
Foglalt	Egység	Műv.	T	RA	RB	EA	EB
Igen (i6)	FP	*	ROB#6		[D2]	ROB#5	
Igen (i7)	Store	store	ROB#7	ROB#4.V			ROB#6
Igen (i3)	Store	store	ROB#3	[R1]	CDB.V		ROB#2

i5 EX fázisba lép, RS-ből törölve,
helyére jön az i7 bejegyzése

R1 értéke kész, de a ROB#4-ből kell venni,
mert még nincs a reg. tárolóba írva

Ciklus: 9

Sorrendvisszaállító buffer (ROB)							
	Utasítás	V	R	Addr	IS	EX	CM
i1:	D0 ← MEM [R1]	[D0]	D0		c2	c3	c4
i2:	D4 ← D0 * D2	[D4]	D4		c4	c5+	c8
i3:	MEM [R1+16] ← D4				c8	c9	
i4:	R1 ← R1 + 8	[R1]	R1		c5	c6	c7
i5:	D0 ← MEM [R1]	[D0]	D0		c7	c8	c9
i6:	D4 ← D0 * D2		D4		c9		
i7:	MEM [R1+16] ← D4						

Regiszter állapot tábla	
Regiszter	Érték/ROB bej.
D0	ROB#5
D2	
D4	ROB#6
R1	ROB#4

CDB	
Érték (V)	ROB bej.
[D0]	ROB#5

i2 kilépett, ROB bejegyzése érvényre juttatható.
DE: D4-t azóta más használja (ROB#6), ezért az eredményt nem kell beírni a regiszter tárolóba

Utasítástároló (RS)							
Foglalt	Egység	Műv.	T	RA	RB	EA	EB
Igen (i6)	FP	*	ROB#6	CDB.V	[D2]	ROB#5	
Igen (i7)	Store	store	ROB#7	ROB#4.V			ROB#6
Nem							

i6 ROB#5-re várt, most megjött.
Ezzel végrehajthatóvá vált, IS-be lép.

i3 EX fázisba lép, RS-ből törölve

Ciklus: 10

Sorrendvisszaállító buffer (ROB)							
	Utasítás	V	R	Addr	IS	EX	CM
i1:	D0 ← MEM [R1]	[D0]	D0		c2	c3	c4
i2:	D4 ← D0 * D2	[D4]	D4		c4	c5+	c8
i3:	MEM [R1+16] ← D4	[D4]		&MEM[R1+16]	c8	c9	c10
i4:	R1 ← R1 + 8	[R1]	R1		c5	c6	c7
i5:	D0 ← MEM [R1]	[D0]	D0		c7	c8	c9
i6:	D4 ← D0 * D2		D4		c9	c10	
i7:	MEM [R1+16] ← D4						

Regiszter állapot tábla	
Regiszter	Érték/ROB bej.
D0	ROB#5
D2	
D4	ROB#6
R1	ROB#4

CDB	
Érték (V)	ROB bej.

Rendelkezésre áll a Store ut. memóriacíme és az írandó adat. Ha sorra kerül, érvényre jut

Utasítástároló (RS)							
Foglalt	Egység	Műv.	T	RA	RB	EA	EB
Nem							
Igen (i7)	Store	store	ROB#7	ROB#4.V			ROB#6
Nem							

Ciklus: 11

Sorrendvisszaállító buffer (ROB)							
	Utasítás	V	R	Addr	IS	EX	CM
i1:	D0 ← MEM [R1]	[D0]	D0		c2	c3	c4
i2:	D4 ← D0 * D2	[D4]	D4		c4	c5+	c8
i3:	MEM [R1+16] ← D4	[D4]		&MEM[R1+16]	c8	c9	c10
i4:	R1 ← R1 + 8	[R1]	R1		c5	c6	c7
i5:	D0 ← MEM [R1]	[D0]	D0		c7	c8	c9
i6:	D4 ← D0 * D2		D4		c9	c10+	
i7:	MEM [R1+16] ← D4						

Regiszter állapot tábla	
Regiszter	Érték/ROB bej.
D0	ROB#5
D2	
D4	ROB#6
R1	ROB#4

CDB	
Érték (V)	ROB bej.

i3 kilépett, ROB bejegyzése érvényre juttatható.
(A tényleges memóriairás itt történik)

Utasítástároló (RS)							
Foglalt	Egység	Műv.	T	RA	RB	EA	EB
Nem							
Igen (i7)	Store	store	ROB#7	ROB#4.V			ROB#6
Nem							

- Egyszerű, de sokkal hatékonyabb a Scoreboard-nál
- Egy pillanat: hol volt itt a regiszterátnevezés?
 - Bújtatott formában
 - Fizikai regiszterek: ROB bejegyzések
 - Ide kerülnek az eredmények
 - Az ettől függő utasítások innen veszik a forrásoperandust
 - Regiszter leképző tábla: regiszter állapot tábla
 - Itt az összerendelés:
 - Melyik architektúrális regiszterhez melyik fizikai (ROB bejegyzés) tartozik



HÁLÓZATI RENDSZEREK
ÉS SZOLGÁLTATÁSOK
TANSZÉK

