

# Tomasulov algoritam



# Drugi dinamički algoritam: Tomasulov algoritam

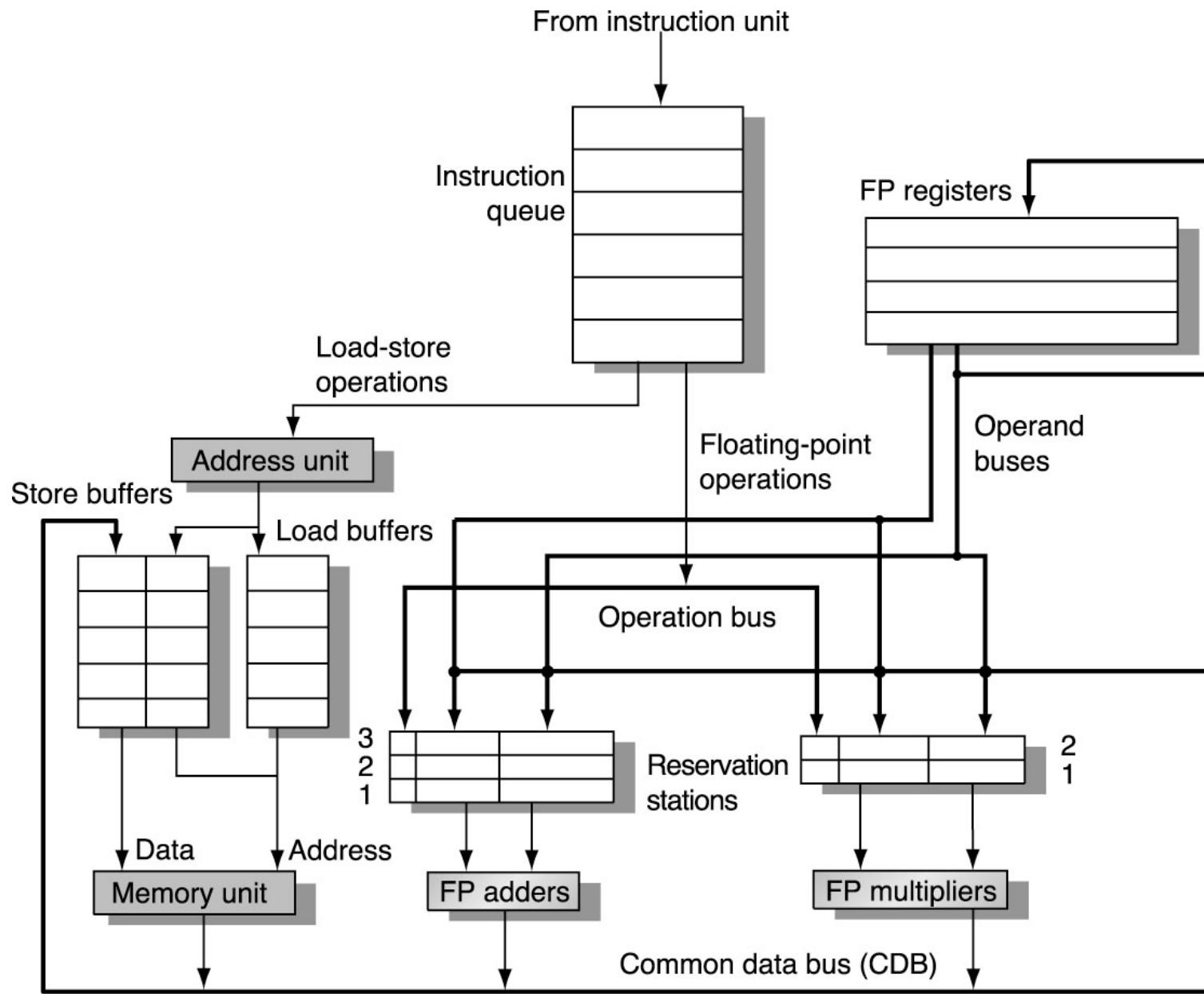
- \* Razvijen u IBM i prvi put implementiran na IBM 360/91 1966, 3 godine nakon scoreboard u CDC 6600. (autor Robert Tomasulo)
- \* Cilj:
  - postizanje visokih performansi bez specijalnih kompjajlera
- \* Kombinuje ključne elemente Sc. šeme i tehnike preimenovanja registara radi eliminacije WAR i WAW hazarda.
  - jedna varijanta je da preimenovanje obavi kompjajler, ali to zahteva veći broj registara opšte namene
  - IBM 360/91 je imao samo 4 FP regista (8 kod CDC 6600)
- \* Današnje CPU arhitekture koje se mogu smatrati naslednicima IBM 360/91 koje implementiraju i koriste varijante Tomasulo Algoritma:

RISC CPUs: Alpha 21264, HP 8600, MIPS R12000, PowerPC G4, AMD Athlon, Pentium III, 4, Xeon,

# IBM 360/91



# Arhitektura staze podataka



# Arhitektura staze podataka

- \* Instrukcije se smeštaju u FIFO bafer, odakle se vrši izdavanje instrukcija
- \* Svakoj funkcionalnoj jedinici (FU) je pridružena "rezervaciona stanica" koja upravlja radom FU i izvršenjem instrukcije.
- \* rezervaciona stanica pamti instrukciju koja je izdata i čeka na izvršenje u FU, operande (vrednosti, a ne imena registara) ako su dostupni, ili imena FU koje će generisati rezultat.
- \* load i store baferi pamte podatke ili adrese (u zavisnosti dokle se stiglo sa izvršenjem instrukcije) load i store baferi se ponašaju slično kao rezervacione stanice
  - load baferi
    - pamte komponente efektivne adrese dok se ona ne izračuna
    - pamte status aktivnih load instrukcija koje čekaju na pristup memoriji
    - pamte rezultat load instrukcije koja čeka na CDB
  - Store buferi
    - pamte komponente efektivne adrese dok se ona ne izračuna
    - pamti adresu i podatak koji treba da se upiše u memoriju

# Arhitektura staze podataka-nast.

- \* Svi rezultati, bilo da dolaze iz FP funkcionalnih jedinica ili load bafera, se smeštaju na CDB (Common data bus), koji dolazi do svih FP registara, rezervacionih stanica i store bafera.
- \* Svakoj FU i load baferu je dodeljen jedinstveni 4-bitni tag (oznaka)
  - 1, 2, 3, 4, 5, 6 za load bufere
  - 8, 9 za jedinice koje ovavljuju množenje/deljenje
  - 10, 11, 12 za jedinice koje obavljaju sabiranje/oduzimanje
  - Tag 0 ukazuje na prisustvo validnog podataka u registru
- \* CDB omogućava da sve FU koje čekaju na isti operand mogu jednovremeno da ga pribave, za razliku od Sc. trhničke gde se rezultat prvo upisuje u registre, kojima zatim sukcesivno pristupaju FU.
- \* FP registri su povezani parom magistrala sa rezervacionim stanicama i jednom magistralom sa store baferima

# Faze u izvršenju instrukcija

## \* Issue – instrukcija se pribavlja iz reda čekanja.

- Ako je u pitanju FP operacija, instrukcija se izdaje ako postoji slobodna rezervaciona stanica i šalju se operandi, ako su u registrima.
- Ako je u pitanju load ili store instrukcija, ona se izdaje ako postoji slobodan load ili store bafer
- Ako nema slobodne rezervacione stanice ili load/store bafera onda postoji strukturni hazard i instrukcija se zaustavlja
- u ovom koraku se vrši i proces preimenovanja registara

## \* Execution (izvršenje) – Ako neki operand nije dostupan, nadgleda se CDB.

- Kada operand postane dostupan smešta se u odgovarajuću RS
- Kada su oba operanda dostupna, izvršava se FP operacija
- Vrši se provera RAW hazarda (čekanjem da operandi postanu dostupni razrešavaju se RAW hazardi)
- Load i store zahtevaju dvostepeno izvršenje:
  - u prvom koraku se izračunava efektivna adresa. a zatim se pamti u load/store bafer
  - Pribavljanje u load bafer se obavlja čim je memorijска единица расположена
  - upisi u store bafere чекају на податак пре него што се обави упис у memoriju

# Faze u izvršenju instrukcija

## \* Write result (upis rezultata) –

- ako je CDB slobodan, rezultat se upisuje na CDB, a odatle u registre i bilo koju FU koja čeka taj rezultat
- Normal data bus: data + destination ("go to" bus)
- Common data bus: data + source ("come from" bus)
  - 64 bits of data + 4 bits of Functional Unit source address
- označiti da je FU slobodna

# Tomasulo i Scoreboard - razlike

- \* Upravljanje *distribuirano* po Rezervacionim stanicama naspram centralizovanog upravljanja kod Scoreboard.
- FP registri opšte namene u instrukcijama su zamenjeni vrednostima (ako su dostupne) ili oznakama rezervacionih stanica (RS) koje će generisati rezultat:
  - Ovaj proces se zove *preimenovanje registara*
  - Izbegavaju se WAR, WAW hazardi.
  - Omogućava hardversko odmotavanje petlji.
  - Ima više RS nego registara opšte namene, što omogućava optimizacije koje kompjuler ne može postići, pa se broj registara opšte namene ne javlja kao usko grlo.
- Rezultati instrukcija se prosledjuju FU iz RS preko *Common Data Bus (CDB)*, (*a ne preko registara*),
- Load i Store se tretiraju kao FU sa svojim RS.

# Upravljanje

## \* Svaka rezervaciona stanica ima 6 polja:

- Busy – označava da li je RS i odgovarajuća FU slobodna
- Op – operacija koja treba da se izvrši (npr. + ili -)
- Vj, Vk – vrednosti izvornih operanada S1 i S2
  - Store baferi imaju smo jedno V polje u kome se pamti rezultat koji treba da se upiše u memoriju
- Qj, Qk – rezervacione stanice koje generišu izvorne operative
  - nema ready flegova kao kod Sc; Qj,Qk=0 => ready
  - store baferi imaju samo jedno Q polje za oznaku RS koja proizvodi rezultat

## \* Za load i store

- A – adresna informacija za load ili store. U startu sadrži neposredni operand, zatim efektivnu adresu kada se izračuna

## \* Register result status (status registra rezultata):

- Qi ukazuje koja funkcionalna jedinica će izvršiti upis (ako postoji)
  - blanko ili 0 ako nema aktivne instrukcije koja će upisati rezultat (tj. podatak je prisutan u registru)

# Tomasulo primer – clk 0

### *Instruction status:*

## *Reservation Stations:*

<i>on Stations:</i>			<i>S1</i>	<i>S2</i>	<i>RS</i>	<i>RS</i>	
<i>Time</i>	<i>Name</i>	<i>Busy</i>	<i>Op</i>	<i>Vj</i>	<i>Vk</i>	<i>Qj</i>	<i>Qk</i>
	Add1	No					
	Add2	No					
	Add3	No					
	Mult1	No					
	Mult2	No					

### *Register result status:*

# Tomasulo primer – clk 1

## Instruction status:

Instruction	j	k	Issue	Exec	Write
				Comp	Result
LD	F6	34+	R2	1	
LD	F2	45+	R3		
MULTD	F0	F2	F4		
SUBD	F8	F6	F2		
DIVD	F10	F0	F6		
ADDD	F6	F8	F2		

	Busy	Address
Load1	Yes	34+R2
Load2	No	
Load3	No	

## Reservation Stations:

Time	Name	Busy	Op	S1	S2	RS	RS
				Vj	Vk	Qj	Qk
	Add1	No					
	Add2	No					
	Add3	No					
	Mult1	No					
	Mult2	No					

## Register result status:

Clock	F0	F2	F4	F6	F8	F10	F12	...	F30
1	FU				Load1				

\* Prva load izdata

# Tomasulo primer Cycle 2

*Instruction status:*

Instruction	<i>j</i>	<i>k</i>	Exec Write		
			<i>Issue</i>	<i>Comp</i>	<i>Result</i>
LD	F6	34+	R2	1	2--
LD	F2	45+	R3	2	
MULTD	F0	F2	F4		
SUBD	F8	F6	F2		
DIVD	F10	F0	F6		
ADDD	F6	F8	F2		

	Busy	Address
Load1	Yes	34+R2
Load2	Yes	45+R3
Load3	No	

*Reservation Stations:*

Time	Name	Busy	<i>S1</i>	<i>S2</i>	<i>RS</i>	<i>RS</i>
			<i>Op</i>	<i>Vj</i>	<i>Vk</i>	<i>Qj</i>
	Add1	No				
	Add2	No				
	Add3	No				
	Mult1	No				
	Mult2	No				

*Register result status:*

Clock	<i>F0</i>	<i>F2</i>	<i>F4</i>	<i>F6</i>	<i>F8</i>	<i>F10</i>	<i>F12</i>	...	<i>F30</i>
2	<i>FU</i>	Load2			Load1				

I druga load izdata (Za razliku od CDC 6600, više load može biti izdato)

# Tomasulo primer Cycle 3

## Instruction status:

Instruction	<i>j</i>	<i>k</i>	<i>Issue</i>	<i>Exec</i>	<i>Write</i>	<i>Busy</i>	<i>Address</i>
				<i>Comp</i>	<i>Result</i>		
LD	F6	34+	R2	1	2--3	Load1	Yes 34+R2
LD	F2	45+	R3	2	3--	Load2	Yes 45+R3
MULTD	F0	F2	F4	3		Load3	No
SUBD	F8	F6	F2				
DIVD	F10	F0	F6				
ADDD	F6	F8	F2				

## Reservation Stations:

Time	Name	<i>Busy</i>	<i>S1</i>	<i>S2</i>	<i>RS</i>	<i>RS</i>
			<i>Op</i>	<i>Vj</i>	<i>Vk</i>	<i>Qj</i>
	Add1	No				
	Add2	No				
	Add3	No				
	Mult1	Yes	MULTD		R(F4)	Load2
	Mult2	No				

## Register result status:

Clock	<i>F0</i>	<i>F2</i>	<i>F4</i>	<i>F6</i>	<i>F8</i>	<i>F10</i>	<i>F12</i>	...	<i>F30</i>
3	<i>FU</i>	Mult1	Load2		Load1				

- **MULT izdata za razliku od scoreboard**
  - Napomena: imena registara su uklonjena ("preimenovana") u rezervacionim stanicama;
  - Load1 okončana; ko čeka na rezultat Load1?

# Tomasulo primer Cycle 4

## Instruction status:

Instruction	<i>j</i>	<i>k</i>	<i>Issue</i>	<i>Exec</i>	<i>Write</i>	<i>Busy</i>	<i>Address</i>	
				<i>Comp</i>	<i>Result</i>			
LD	F6	34+	R2	1	2--3	4	Load1	No
LD	F2	45+	R3	2	3--4		Load2	Yes
MULTD	F0	F2	F4	3			Load3	No
SUBD	F8	F6	F2	4				
DIVD	F10	F0	F6					
ADDD	F6	F8	F2					

## Reservation Stations:

Time	Name	Busy	<i>S1</i>	<i>S2</i>	<i>RS</i>	<i>RS</i>
			<i>Op</i>	<i>Vj</i>	<i>Vk</i>	<i>Qj</i>
	Add1	Yes	SUBD	M(A1)		Load2
	Add2	No				
	Add3	No				
	Mult1	Yes	MULTD		R(F4) Load2	
	Mult2	No				

## Register result status:

Clock		<i>F0</i>	<i>F2</i>	<i>F4</i>	<i>F6</i>	<i>F8</i>	<i>F10</i>	<i>F12</i>	...	<i>F30</i>
4	<i>FU</i>	Mult1	Load2		M(A1)	Add1				

- SUBD izdata
- Load2 okončana; ko čeka na rezultat Load2?

# Tomasulo primer Cycle 5

*Instruction status:*

Instruction	<i>j</i>	<i>k</i>	<i>Issue</i>	<i>Exec</i>	<i>Write</i>	<i>Busy</i>	<i>Address</i>	
				<i>Comp</i>	<i>Result</i>			
LD	F6	34+	R2	1	2--3	4	Load1	No
LD	F2	45+	R3	2	3--4	5	Load2	No
MULTD	F0	F2	F4	3			Load3	No
SUBD	F8	F6	F2	4				
DIVD	F10	F0	F6	5				
ADDD	F6	F8	F2					

*Reservation Stations:*

<i>Time</i>	<i>Name</i>	<i>Busy</i>	<i>S1</i>	<i>S2</i>	<i>RS</i>	<i>RS</i>	
			<i>Op</i>	<i>Vj</i>	<i>Vk</i>	<i>Qj</i>	<i>Qk</i>
2	Add1	Yes	SUBD	M(A1)	M(A2)		
	Add2	No					
	Add3	No					
10	Mult1	Yes	MULTD	M(A2)	R(F4)		
	Mult2	Yes	DIVD		M(A1)	Mult1	

*Register result status:*

Clock	<i>F0</i>	<i>F2</i>	<i>F4</i>	<i>F6</i>	<i>F8</i>	<i>F10</i>	<i>F12</i>	...	<i>F30</i>
<i>FU</i>	Mult1	M(A2)		M(A1)	Add1	Mult2			
5									

- DIVD izdata, MULTD i SUBD dobijaju operande

# Tomasulo primer Cycle 6

## Instruction status:

Instruction	<i>j</i>	<i>k</i>	<i>Issue</i>	<i>Exec</i>	<i>Write</i>	<i>Busy</i>	<i>Address</i>	
				<i>Comp</i>	<i>Result</i>			
LD	F6	34+	R2	1	2--3	4	Load1	No
LD	F2	45+	R3	2	3--4	5	Load2	No
MULTD	F0	F2	F4	3	6--		Load3	No
SUBD	F8	F6	F2	4	6--			
DIVD	F10	F0	F6	5				
ADDD	F6	F8	F2	6				

## Reservation Stations:

			<i>S1</i>	<i>S2</i>	<i>RS</i>	<i>RS</i>	
<i>Time</i>	<i>Name</i>	<i>Busy</i>	<i>Op</i>	<i>Vj</i>	<i>Vk</i>	<i>Qj</i>	<i>Qk</i>
1	Add1	Yes	SUBD	M(A1)	M(A2)		
	Add2	Yes	ADDD		M(A2)	Add1	
	Add3	No					
9	Mult1	Yes	MULTD	M(A2)	R(F4)		
	Mult2	Yes	DIVD		M(A1)	Mult1	

## Register result status:

Clock		<i>F0</i>	<i>F2</i>	<i>F4</i>	<i>F6</i>	<i>F8</i>	<i>F10</i>	<i>F12</i>	...	<i>F30</i>
<i>6</i>	<i>FU</i>	Mult1	M(A2)		Add2	Add1	Mult2			

- ADDD je izdata za razliku od scoreboard - WAR hazard eliminisan preimenovanjem
- MULTD i SUBD kreću sa izvršenjem

# Tomasulo primer Cycle 7

## Instruction status:

Instruction	<i>j</i>	<i>k</i>	<i>Issue</i>	<i>Exec</i>	<i>Write</i>	<i>Busy</i>	<i>Address</i>	
				<i>Comp</i>	<i>Result</i>			
LD	F6	34+	R2	1	2--3	4	Load1	No
LD	F2	45+	R3	2	3--4	5	Load2	No
MULTD	F0	F2	F4	3	6--		Load3	No
SUBD	F8	F6	F2	4	6--7			
DIVD	F10	F0	F6	5				
ADDD	F6	F8	F2	6				

## Reservation Stations:

			<i>S1</i>	<i>S2</i>	<i>RS</i>	<i>RS</i>	
<i>Time</i>	<i>Name</i>	<i>Busy</i>	<i>Op</i>	<i>Vj</i>	<i>Vk</i>	<i>Qj</i>	<i>Qk</i>
0	Add1	Yes	SUBD	M(A1)	M(A2)		
	Add2	Yes	ADDD		M(A2)	Add1	
	Add3	No					
8	Mult1	Yes	MULTD	M(A2)	R(F4)		
	Mult2	Yes	DIVD		M(A1)	Mult1	

## Register result status:

Clock	<i>F0</i>	<i>F2</i>	<i>F4</i>	<i>F6</i>	<i>F8</i>	<i>F10</i>	<i>F12</i>	...	<i>F30</i>
<i>FU</i>	Mult1	M(A2)		Add2	Add1	Mult2			
7									

- SUBD (Add1) okončana; ko čeka na rezultat?

# Tomasulo primer Cycle 8

## Instruction status:

Instruction	<i>j</i>	<i>k</i>	<i>Issue</i>	<i>Exec</i>	<i>Write</i>	<i>Busy</i>	<i>Address</i>	
				<i>Comp</i>	<i>Result</i>			
LD	F6	34+	R2	1	2--3	4	Load1	No
LD	F2	45+	R3	2	3--4	5	Load2	No
MULTD	F0	F2	F4	3	6--		Load3	No
SUBD	F8	F6	F2	4	6--7	8		
DIVD	F10	F0	F6	5				
ADDD	F6	F8	F2	6				

## Reservation Stations:

Time	Name	<i>Busy</i>	<i>S1</i>	<i>S2</i>	<i>RS</i>	<i>RS</i>	
			<i>Op</i>	<i>Vj</i>	<i>Vk</i>	<i>Qj</i>	<i>Qk</i>
	Add1	No					
2	Add2	Yes	ADDD	(M-M)	M(A2)		
	Add3	No					
7	Mult1	Yes	MULTD	M(A2)	R(F4)		
	Mult2	Yes	DIVD		M(A1)	Mult1	

## Register result status:

Clock	<i>F0</i>	<i>F2</i>	<i>F4</i>	<i>F6</i>	<i>F8</i>	<i>F10</i>	<i>F12</i>	...	<i>F30</i>
8	<i>FU</i>	Mult1	M(A2)		Add2	(M-M)	Mult2		

rezultat add1 je na CDB

ADDD može da krene u izvršenje; MULTD se još uvek izvršava

# Tomasulo primer Cycle 9

*Instruction status:*

Instruction	j	k	Issue	Exec	Write	Busy	Address	
LD	F6	34+	R2	1	2--3	4	Load1	No
LD	F2	45+	R3	2	3--4	5	Load2	No
MULTD	F0	F2	F4	3	6--		Load3	No
SUBD	F8	F6	F2	4	6--7	8		
DIVD	F10	F0	F6	5				
ADDD	F6	F8	F2	6	9--			

*Reservation Stations:*

Time	Name	Busy	Op	S1	S2	RS	RS
				Vj	Vk	Qj	Qk
	Add1	No					
1	Add2	Yes	ADDD	(M-M)	M(A2)		
	Add3	No					
6	Mult1	Yes	MULTD	M(A2)	R(F4)		
	Mult2	Yes	DIVD		M(A1)	Mult1	

*Register result status:*

Clock		F0	F2	F4	F6	F8	F10	F12	...	F30
9	FU	Mult1	M(A2)		Add2	(M-M)	Mult2			

MULTD i ADDD se još uvek izvršavaju

# Tomasulo primer Cycle 10

*Instruction status:*

Instruction	<i>j</i>	<i>k</i>	<i>Issue</i>	<i>Exec</i>	<i>Write</i>	<i>Busy</i>	<i>Address</i>	
				<i>Comp</i>	<i>Result</i>			
LD	F6	34+	R2	1	2--3	4	Load1	No
LD	F2	45+	R3	2	3--4	5	Load2	No
MULTD	F0	F2	F4	3	6--		Load3	No
SUBD	F8	F6	F2	4	6--7	8		
DIVD	F10	F0	F6	5				
ADDD	F6	F8	F2	6	9--10			

*Reservation Stations:*

<i>Time</i>	<i>Name</i>	<i>Busy</i>	<i>S1</i>	<i>S2</i>	<i>RS</i>	<i>RS</i>	
			<i>Op</i>	<i>Vj</i>	<i>Vk</i>	<i>Qj</i>	<i>Qk</i>
	Add1	No					
0	Add2	Yes	ADDD	(M-M)	M(A2)		
	Add3	No					
5	Mult1	Yes	MULTD	M(A2)	R(F4)		
	Mult2	Yes	DIVD		M(A1)	Mult1	

*Register result status:*

Clock	<i>F0</i>	<i>F2</i>	<i>F4</i>	<i>F6</i>	<i>F8</i>	<i>F10</i>	<i>F12</i>	...	<i>F30</i>
10	<i>FU</i>	Mult1	M(A2)		Add2	(M-M)	Mult2		

- Add2 okončana;
- MULTD se još uvek izvršava

# Tomasulo primer Cycle 11

## Instruction status:

Instruction	<i>j</i>	<i>k</i>	<i>Issue</i>	<i>Exec</i>	<i>Write</i>	<i>Busy</i>	<i>Address</i>	
				<i>Comp</i>	<i>Result</i>			
LD	F6	34+	R2	1	2--3	4	Load1	No
LD	F2	45+	R3	2	3--4	5	Load2	No
MULTD	F0	F2	F4	3	6--		Load3	No
SUBD	F8	F6	F2	4	6--7	8		
DIVD	F10	F0	F6	5				
ADDD	F6	F8	F2	6	9--10	11		

## Reservation Stations:

Time	Name	Busy	<i>S1</i>	<i>S2</i>	<i>RS</i>	<i>RS</i>
			<i>Op</i>	<i>Vj</i>	<i>Vk</i>	<i>Qj</i>
	Add1	No				
	Add2	No				
	Add3	No				
4	Mult1	Yes	MULTD	M(A2)	R(F4)	
	Mult2	Yes	DIVD		M(A1)	Mult1

## Register result status:

Clock		<i>F0</i>	<i>F2</i>	<i>F4</i>	<i>F6</i>	<i>F8</i>	<i>F10</i>	<i>F12</i>	...	<i>F30</i>
<i>11</i>	<i>FU</i>	Mult1	M(A2)		(M-M+N)	(M-M)	Mult2			

- upis rezultata ADDD za razliku od scoreboard kod koga je postojao WAR hazard

# Tomasulo primer Cycle 12

## Instruction status:

Instruction	<i>j</i>	<i>k</i>	<i>Issue</i>	<i>Exec</i>	<i>Write</i>	<i>Busy</i>	<i>Address</i>	
				<i>Comp</i>	<i>Result</i>			
LD	F6	34+	R2	1	2--3	4	Load1	No
LD	F2	45+	R3	2	3--4	5	Load2	No
MULTD	F0	F2	F4	3	6--		Load3	No
SUBD	F8	F6	F2	4	6--7	8		
DIVD	F10	F0	F6	5				
ADDD	F6	F8	F2	6	9--10	11		

## Reservation Stations:

Time	Name	<i>Busy</i>	<i>S1</i>	<i>S2</i>	<i>RS</i>	<i>RS</i>	
			<i>Op</i>	<i>Vj</i>	<i>Vk</i>	<i>Qj</i>	<i>Qk</i>
	Add1	No					
	Add2	No					
	Add3	No					
3	Mult1	Yes	MULTD	M(A2)	R(F4)		
	Mult2	Yes	DIVD		M(A1)	Mult1	

## Register result status:

Clock	<i>F0</i>	<i>F2</i>	<i>F4</i>	<i>F6</i>	<i>F8</i>	<i>F10</i>	<i>F12</i>	...	<i>F30</i>
12	FU	Mult1	M(A2)		(M-M+N(M-M))	Mult2			

Još uvek traje izvršenje MULTD; DIVD čeka na rezultat

# Tomasulo primer Cycle 13

## Instruction status:

Instruction	<i>j</i>	<i>k</i>	<i>Issue</i>	<i>Exec</i>	<i>Write</i>	<i>Busy</i>	<i>Address</i>	
				<i>Comp</i>	<i>Result</i>			
LD	F6	34+	R2	1	2--3	4	Load1	No
LD	F2	45+	R3	2	3--4	5	Load2	No
MULTD	F0	F2	F4	3	6--		Load3	No
SUBD	F8	F6	F2	4	6--7	8		
DIVD	F10	F0	F6	5				
ADDD	F6	F8	F2	6	9--10	11		

## Reservation Stations:

Time	Name	<i>Busy</i>	<i>S1</i>	<i>S2</i>	<i>RS</i>	<i>RS</i>	
			<i>Op</i>	<i>Vj</i>	<i>Vk</i>	<i>Qj</i>	<i>Qk</i>
	Add1	No					
	Add2	No					
	Add3	No					
2	Mult1	Yes	MULTD	M(A2)	R(F4)		
	Mult2	Yes	DIVD		M(A1)	Mult1	

## Register result status:

Clock	<i>F0</i>	<i>F2</i>	<i>F4</i>	<i>F6</i>	<i>F8</i>	<i>F10</i>	<i>F12</i>	...	<i>F30</i>
13	FU	Mult1	M(A2)		(M-M+N)	(M-M)	Mult2		

Još uvek traje izvršenje MULTD; DIVD čeka na rezultat

# Tomasulo primer Cycle 14

*Instruction status:*

Instruction	<i>j</i>	<i>k</i>	<i>Issue</i>	<i>Exec</i>	<i>Write</i>	<i>Busy</i>	<i>Address</i>	
				<i>Comp</i>	<i>Result</i>			
LD	F6	34+	R2	1	2--3	4	Load1	No
LD	F2	45+	R3	2	3--4	5	Load2	No
MULTD	F0	F2	F4	3	6--		Load3	No
SUBD	F8	F6	F2	4	6--7	8		
DIVD	F10	F0	F6	5				
ADDD	F6	F8	F2	6	9--10	11		

*Reservation Stations:*

Time	Name	<i>Busy</i>	<i>S1</i>	<i>S2</i>	<i>RS</i>	<i>RS</i>	
			<i>Op</i>	<i>Vj</i>	<i>Vk</i>	<i>Qj</i>	<i>Qk</i>
	Add1	No					
	Add2	No					
	Add3	No					
1	Mult1	Yes	MULTD	M(A2)	R(F4)		
	Mult2	Yes	DIVD		M(A1)	Mult1	

*Register result status:*

Clock	<i>F0</i>	<i>F2</i>	<i>F4</i>	<i>F6</i>	<i>F8</i>	<i>F10</i>	<i>F12</i>	...	<i>F30</i>
14	<i>FU</i>	Mult1	M(A2)		(M-M+M)	(M-M)	Mult2		

Još uvek traje izvršenje MULTD; DIVD čeka na rezultat

# Tomasulo primer Cycle 15

## Instruction status:

Instruction	<i>j</i>	<i>k</i>	<i>Issue</i>	<i>Exec</i>	<i>Write</i>	<i>Busy</i>	<i>Address</i>	
				<i>Comp</i>	<i>Result</i>			
LD	F6	34+	R2	1	2--3	4	Load1	No
LD	F2	45+	R3	2	3--4	5	Load2	No
MULTD	F0	F2	F4	3	6--15		Load3	No
SUBD	F8	F6	F2	4	6--7	8		
DIVD	F10	F0	F6	5				
ADDD	F6	F8	F2	6	10-Sep	11		

## Reservation Stations:

Time	Name	Busy	<i>S1</i>	<i>S2</i>	<i>RS</i>	<i>RS</i>	
			<i>Op</i>	<i>Vj</i>	<i>Vk</i>	<i>Qj</i>	<i>Qk</i>
	Add1	No					
	Add2	No					
	Add3	No					
0	Mult1	Yes	MULTD	M(A2)	R(F4)		
	Mult2	Yes	DIVD		M(A1)	Mult1	

## Register result status:

Clock	<i>F0</i>	<i>F2</i>	<i>F4</i>	<i>F6</i>	<i>F8</i>	<i>F10</i>	<i>F12</i>	...	<i>F30</i>
15	FU	Mult1	M(A2)		(M-M+N)	(M-M)	Mult2		

MULTD okončava izvršenje, rezultat dostupan na CDB

# Tomasulo primer Cycle 16

*Instruction status:*

Instruction	<i>j</i>	<i>k</i>	<i>Issue</i>	<i>Exec</i>	<i>Write</i>	<i>Busy</i>	<i>Address</i>	
				<i>Comp</i>	<i>Result</i>			
LD	F6	34+	R2	1	2--3	4	Load1	No
LD	F2	45+	R3	2	3--4	5	Load2	No
MULTD	F0	F2	F4	3	6--15	16	Load3	No
SUBD	F8	F6	F2	4	6--7	8		
DIVD	F10	F0	F6	5				
ADDD	F6	F8	F2	6	9--10	11		

*Reservation Stations:*

Time	Name	<i>Busy</i>	<i>S1</i>	<i>S2</i>	<i>RS</i>	<i>RS</i>	
			<i>Op</i>	<i>Vj</i>	<i>Vk</i>	<i>Qj</i>	<i>Qk</i>
	Add1	No					
	Add2	No					
	Add3	No					
	Mult1	No					
40	Mult2	Yes	DIVD	M*F4	M(A1)		

*Register result status:*

Clock	<i>F0</i>	<i>F2</i>	<i>F4</i>	<i>F6</i>	<i>F8</i>	<i>F10</i>	<i>F12</i>	...	<i>F30</i>
16	<i>FU</i>	M*F4	M(A2)		(M-M+N)(M-M)	Mult2			

Ostala je samo DIVD instrukcija (kreće u izvršenje)

preskočimo nekoliko clk ciklusa!

# Tomasulo primer Cycle 55

*Instruction status:*

Instruction	<i>j</i>	<i>k</i>	<i>Issue</i>	<i>Exec</i>	<i>Write</i>	<i>Busy</i>	<i>Address</i>	
				<i>Comp</i>	<i>Result</i>			
LD	F6	34+	R2	1	2--3	4	Load1	No
LD	F2	45+	R3	2	3--4	5	Load2	No
MULTD	F0	F2	F4	3	6--15	16	Load3	No
SUBD	F8	F6	F2	4	6--7	8		
DIVD	F10	F0	F6	5	17--			
ADDD	F6	F8	F2	6	9--10	11		

*Reservation Stations:*

<i>Time</i>	<i>Name</i>	<i>Busy</i>	<i>Op</i>	<i>S1</i>	<i>S2</i>	<i>RS</i>	<i>RS</i>
	Add1	No					
	Add2	No					
	Add3	No					
	Mult1	No					
1	Mult2	Yes	DIVD	M*F4	M(A1)		

*Register result status:*

Clock	<i>F0</i>	<i>F2</i>	<i>F4</i>	<i>F6</i>	<i>F8</i>	<i>F10</i>	<i>F12</i>	...	<i>F30</i>
55	<i>FU</i>	M*F4	M(A2)		(M-M+N)(M-M)	Mult2			

- DIVD (Mult2) se još izvršava

# Tomasulo primer Cycle 56

## Instruction status:

Instruction	<i>j</i>	<i>k</i>	Issue	Exec	Write	Busy	Address	
				Comp	Result			
LD	F6	34+	R2	1	2--3	4	Load1	No
LD	F2	45+	R3	2	3--4	5	Load2	No
MULTD	F0	F2	F4	3	6--15	16	Load3	No
SUBD	F8	F6	F2	4	6--7	8		
DIVD	F10	F0	F6	5	17--56			
ADDD	F6	F8	F2	6	9--10	11		

## Reservation Stations:

Time	Name	Busy	Op	S1	S2	RS	RS
				Vj	Vk	Qj	Qk
	Add1	No					
	Add2	No					
	Add3	No					
	Mult1	No					
0	Mult2	Yes	DIVD	M*F4	M(A1)		

## Register result status:

Clock	F0	F2	F4	F6	F8	F10	F12	...	F30
56	FU	M*F4	M(A2)	(M-M+N(M-M)	Mult2				

- Mult2 je okončala izvršenje

# Tomasulo primer Cycle 57

## Instruction status:

Instruction	<i>j</i>	<i>k</i>	Issue	Exec	Write	Busy	Address
				Comp	Result		
LD	F6	34+	R2	1	3	4	Load1
LD	F2	45+	R3	2	4	5	Load2
MULTD	F0	F2	F4	3	15	16	Load3
SUBD	F8	F6	F2	4	7	8	
DIVD	F10	F0	F6	5	56	57	
ADDD	F6	F8	F2	6	10	11	

## Reservation Stations:

Time	Name	Busy	Op	<i>V<sub>j</sub></i>	<i>V<sub>k</sub></i>	<i>Q<sub>j</sub></i>	<i>Q<sub>k</sub></i>
	Add1	No					
	Add2	No					
	Add3	No					
	Mult1	No					
0	Mult2	Yes	DIVD	M*F4	M(A1)		

## Register result status:

Clock	<i>F0</i>	<i>F2</i>	<i>F4</i>	<i>F6</i>	<i>F8</i>	<i>F10</i>	<i>F12</i>	...	<i>F30</i>
56	FU	M*F4	M(A2)		(M-M+N(M-M))	Mult2			

- ponovo imamo: In-order issue, out-of-order izvršenje i okončanje.

# Scoreboard je okončao izvršenje u 62. ciklusu

Scoreboard

*Instruction status:*

Instruction	j	k	Issue	Read	Exec	Write
				Oper	Comp	Result
LD	F6	34+	R2	1	2	3
LD	F2	45+	R3	5	6	7
MULTD	F0	F2	F4	6	9	19
SUBD	F8	F6	F2	7	9	11
DIVD	F10	F0	F6	8	21	61
ADDD	F6	F8	F2	13	14	16

Tomasulo

Issue	Exec	Write
	Comp	Result
1	3	4
2	4	5
3	15	16
4	7	8
5	56	57
6	10	11

- Zašto izvršenje sa scoreboard traje duže?  
**Strukturni hazardi**  
**WAW i WAR hazardi**  
**nema forwarding-a**

- \* Protočne Funkcionalne jedinice (6 load, 3 store, 3 +, 2 x/÷)
- \* veličina prozora:  $\leq 14$  instrukcija
- \* Nema izdavanja u slučaju struktturnih hazara
- \* WAW: izbegavaju se preimenovanjem
- \* WAR: izbegavaju se preimenovanjem
- \* Emisija rezultata iz FU preko CDB (Implementira forwarding)
- \* Upravljanje: rezervacione stanice (distribuirano)
- \* Više FUs (nisu protočne) (1 load/store, 1 +, 2 x, 1 ÷)
- \*  $\leq 5$  instrukcija
- \* Nema izdavanja u slučaju struktturnih hazara
- \* zaustavlja se izdavanje
- \* zaustavlja se okončanje
- \* Write/read registara (Forwarding *nije* podržan)
- \* centralizovano u scoreboard

# Upravljanje kod Tomasulovog algoritma

u tabeli registra rezultata označiti da će FU generisati rezultat za registar D

Instruction status	Wait until	Action or bookkeeping
Issue	Station or buffer empty	<pre> if (Register['S1'].Qi ≠ 0)     {RS[r].Qj←1; Register['S1'].Qi} else {RS[r].Vj← S1; RS[r].Qj← 0}; if (Register['S2'].Qi≠0)     {RS[r].Qk← Register[S2].Qi}; else {RS[r].Vj← S2; RS[r].Qk← 0} RS[r].Busy← yes; Register['D'].Qi← RS[r].Vj; </pre> <p>čeka se da operandi postanu dostupni</p>
Execute	(RS[r].Qj=0) and (RS[r].Qk=0)	<p>None—operands are available</p> <p>upisati rezultat u registar i obrisati odgovarajuće polje u tabeli registra rezultata</p>
Write result	Execution completed at r and CDB available	<pre> ∀x(if (Register[x].Qi=0) (RS[x].Vj← result;     RS[x].Qi← 0)); ∀x(if (RS[x].Qi=r) (RS[x].Vj← result;     RS[x].Qj← 0)); ∀x(if (RS[x].Qk=r) (RS[x].Vj← result;     RS[x].Qk← 0)); ∀x(if (Store[x].Qi=0) (RS[x].Vj← result;     Store[x].Qi← 0)); </pre> <p>proveriti da li neka FU čeka na rezult;</p> <p>upisati rezultat u RS</p> <p>osloboditi FU</p>

# Hw odmotavanje petlje

\* Prava snaga eliminisanja WAW i WAR hazarda kroz dinamičko preimenovanje registara najbolje se može videti na pimeru izvršenja petlje

➤ loop:	LD	F0, 0(R1)
➤	MULD	F4, F0, F2
➤	SD	F4, 0(R1)
➤	SUBI	R1, R1, #8
➤	BNEZ	R1, Loop

\* Ako je predvidajanje da će se grananje obaviti, korišćenje RS će omogućiti da se više iteracija petlje izvršava jednovremeno

\* ovo se postiže bez promene koda

- u suštini petlja se dinamički odmotava uz pomoć hw korišćenjem RS koje deluju kao dodatni registri

# Usvajamo sledeće

- \* množenje traje 4 clk.
- \* prva load traje 8 clocks (recimo zbog keš promašaja)
- \* druga load traje 4 clocks (pogodak).
- \* inicijalno  $R_1 = 80$ .
- \* branch je predvidjen kao taken (obavlja se).
- \* posmatramo izvršenje prve dve iteracije

# Loop primer

### *Instruction status:*

Instruction status:					Exec		Write		
ITER	Instruction	j	k	Issue	CompResult		Busy	Addr	Fu
1	LD	F0	0	R1			Load1	No	
1	MULTD	F4	F0	F2			Load2	No	
1	SD	F4	0	R1			Load3	No	
2	LD	F0	0	R1			Store1	No	
2	MULTD	F4	F0	F2			Store2	No	
2	SD	F4	0	R1			Store3	No	

## *Reservation Stations:*

Observation Stations:			<i>S1</i>	<i>S2</i>	<i>RS</i>	Code:				
<i>Time</i>	<i>Name</i>	<i>Busy</i>	<i>Op</i>	<i>Vj</i>	<i>Vk</i>	<i>Qj</i>	<i>Qk</i>			
	Add1	No					LD	F0	0	R1
	Add2	No					MULTD	F4	F0	F2
	Add3	No					SD	F4	0	R1
	Mult1	No					SUBI	R1	R1	#8
	Mult2	No					BNEZ	R1	Loop	

## *Register result status*

# Loop primer Cycle 1

*Instruction status:*

ITER	Instruction	j	k	Exec Write		Busy	Addr	Fu
				Issue	CompResult			
1	LD	F0	0	R1	1	Load1	Yes	80
1	MULTD	F4	F0	F2		Load2	No	
1	SD	F4	0	R1		Load3	No	
2	LD	F0	0	R1		Store1	No	
2	MULTD	F4	F0	F2		Store2	No	
2	SD	F4	0	R1		Store3	No	

*Reservation Stations:*

Time	Name	Busy	Op	RS			Code:
				S1	S2	RS	
	Add1	No					LD F0 0 R1
	Add2	No					MULTD F4 F0 F2
	Add3	No					SD F4 0 R1
	Mult1	No					SUBI R1 R1 #8
	Mult2	No					BNEZ R1 Loop

*Register result status*

Clock	R1	F0	F2	F4	F6	F8	F10	F12	...	F30
1	80	Fu	Load1							

Prva load izdata

# Loop primer Cycle 2

*Instruction status:*

ITER	Instruction	j	k	Exec Write		Busy	Addr	Fu
				Issue	CompResult			
1	LD	F0	0	R1	1	Load1	Yes	80
1	MULTD	F4	F0	F2	2	Load2	No	
1	SD	F4	0	R1		Load3	No	
2	LD	F0	0	R1		Store1	No	
2	MULTD	F4	F0	F2		Store2	No	
2	SD	F4	0	R1		Store3	No	

*Reservation Stations:*

Time	Name	Busy	Op	S1	S2	RS	Code:
				Vj	Vk	Qj	
	Add1	No					LD F0 0 R1
	Add2	No					MULTD F4 F0 F2
	Add3	No					SD F4 0 R1
	Mult1	Yes	Multd		R(F2)	Load1	SUBI R1 R1 #8
	Mult2	No					BNEZ R1 Loop

*Register result status*

Clock	R1	F0	F2	F4	F6	F8	F10	F12	...	F30
2	80	Fu	Load1	Mult1						

MULTD izdata

# Loop primer Cycle 3

*Instruction status:*

ITER	Instruction	j	k	Exec Write		Busy	Addr	Fu
				Issue	CompResult			
1	LD	F0	0	R1	1	Load1	Yes	80
1	MULTD	F4	F0	F2	2	Load2	No	
1	SD	F4	0	R1	3	Load3	No	
2	LD	F0	0	R1		Store1	Yes	80
2	MULTD	F4	F0	F2		Store2	No	
2	SD	F4	0	R1		Store3	No	

*Reservation Stations:*

Time	Name	Busy	Op	S1	S2	RS	Code:	
				Vj	Vk	Qj		
	Add1	No					LD	F0 0 R1
	Add2	No					MULTD	F4 F0 F2
	Add3	No					SD	F4 0 R1
	Mult1	Yes	Multd				SUBI	R1 R1 #8
	Mult2	No					BNEZ	R1 Loop

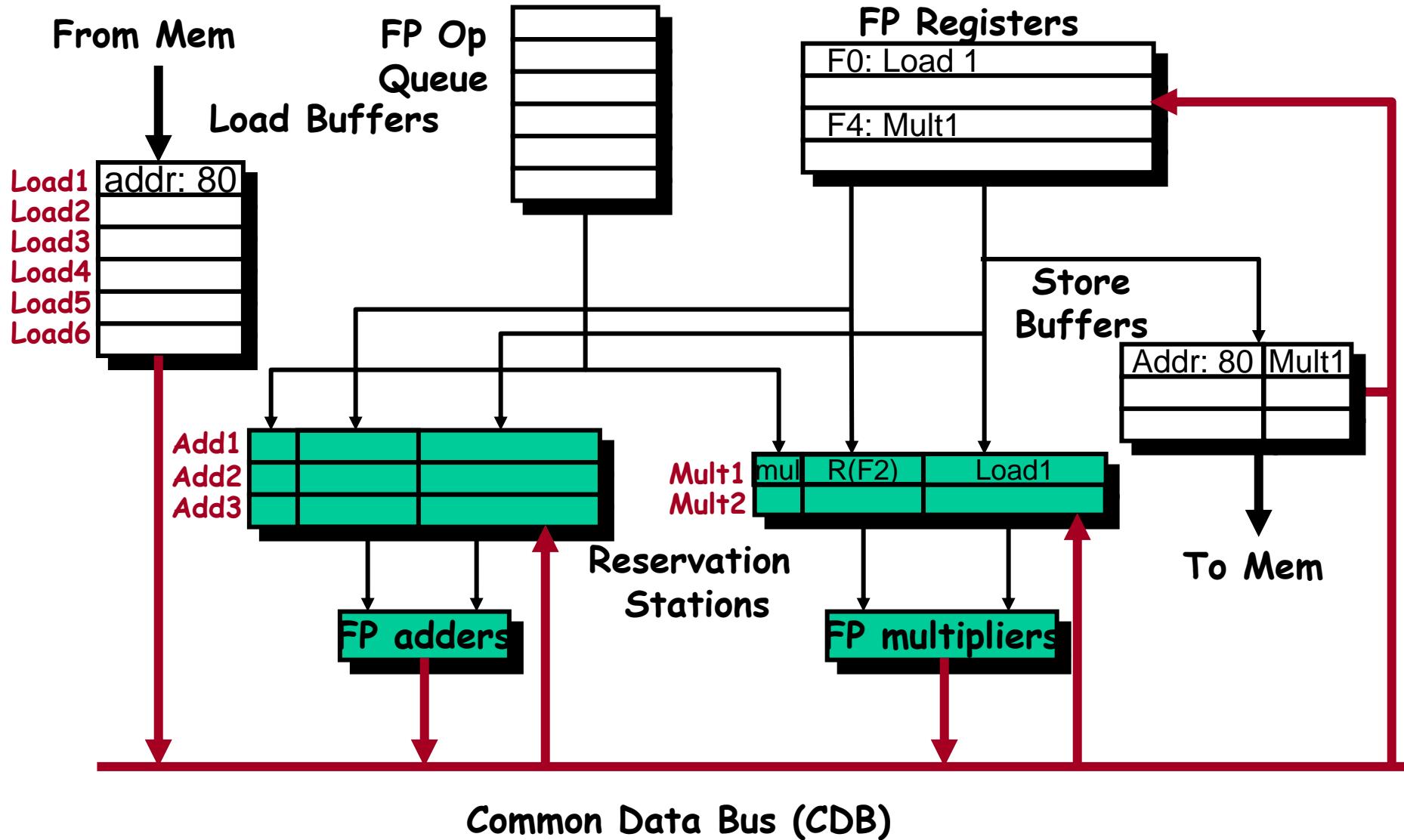
*Register result status*

Clock	R1	F0	F2	F4	F6	F8	F10	F12	...	F30
3	80	Fu	Load1		Mult1					

\* Izdata je SD

\* implicitno preimenovanje: u MULT1 ne figurišu imena registara

# Šta to fizički značí?



# Loop primer Cycle 4

*Instruction status:*

ITER	Instruction	j	k	Exec Write		Busy	Addr	Fu
				Issue	CompResult			
1	LD	F0	0	R1	1	Load1	Yes	80
1	MULTD	F4	F0	F2	2	Load2	No	
1	SD	F4	0	R1	3	Load3	No	
2	LD	F0	0	R1		Store1	Yes	80
2	MULTD	F4	F0	F2		Store2	No	Mult1
2	SD	F4	0	R1		Store3	No	

*Reservation Stations:*

Time	Name	Busy	Op	S1	S2	RS	Code:
				Vj	Vk	Qj	
	Add1	No					LD F0 0 R1
	Add2	No					MULTD F4 F0 F2
	Add3	No					SD F4 0 R1
	Mult1	Yes	Multd		R(F2)	Load1	SUBI R1 R1 #8
	Mult2	No					BNEZ R1 Loop

*Register result status*

Clock	R1	F0	F2	F4	F6	F8	F10	F12	...	F30
4	80	Fu	Load1		Mult1					

\* izdavanje SUBI Instrukcije

# Loop primer Cycle 5

*Instruction status:*

ITER	Instruction	j	k	Exec Write		Busy	Addr	Fu
				Issue	CompResult			
1	LD	F0	0	R1	1	Load1	Yes	80
1	MULTD	F4	F0	F2	2	Load2	No	
1	SD	F4	0	R1	3	Load3	No	
2	LD	F0	0	R1		Store1	Yes	80
2	MULTD	F4	F0	F2		Store2	No	Mult1
2	SD	F4	0	R1		Store3	No	

*Reservation Stations:*

Time	Name	Busy	Op	RS			Code:
				S1	S2	RS	
	Add1	No					LD F0 0 R1
	Add2	No					MULTD F4 F0 F2
	Add3	No					SD F4 0 R1
	Mult1	Yes	Multd		R(F2)	Load1	SUBI R1 R1 #8
	Mult2	No					BNEZ R1 Loop

*Register result status*

Clock	R1	F0	F2	F4	F6	F8	F10	F12	...	F30
5	72	Fu	Load1		Mult1					

\* i, BNEZ instrukcije

# Loop primer Cycle 6

*Instruction status:*

ITER	Instruction	j	k	Exec Write		Busy	Addr	Fu
				Issue	CompResult			
1	LD	F0	0	R1	1	Load1	Yes	80
1	MULTD	F4	F0	F2	2	Load2	Yes	72
1	SD	F4	0	R1	3	Load3	No	
2	LD	F0	0	R1	6	Store1	Yes	80
2	MULTD	F4	F0	F2		Store2	No	Mult1
2	SD	F4	0	R1		Store3	No	

*Reservation Stations:*

Time	Name	Busy	Op	S1	S2	RS	Code:
				Vj	Vk	Qj	
	Add1	No					LD F0 0 R1 ←
	Add2	No					MULTD F4 F0 F2
	Add3	No					SD F4 0 R1
	Mult1	Yes	Multd		R(F2)	Load1	SUBI R1 R1 #8
	Mult2	No					BNEZ R1 Loop

*Register result status*

Clock	R1	F0	F2	F4	F6	F8	F10	F12	...	F30
6	72	Fu	Load2		Mult1					

- \* Izdaje se sledeća LD; u F0 se nikad ne upiše rezultat Load1 sa lokacije 80

# Loop Example Cycle 7

*Instruction status:*

ITER	Instruction	j	k	Exec Write		Busy	Addr	Fu
				Issue	CompResult			
1	LD	F0	0	R1	1	Load1	Yes	80
1	MULTD	F4	F0	F2	2	Load2	Yes	72
1	SD	F4	0	R1	3	Load3	No	
2	LD	F0	0	R1	6	Store1	Yes	80
2	MULTD	F4	F0	F2	7	Store2	No	Mult1
2	SD	F4	0	R1		Store3	No	

*Reservation Stations:*

Time	Name	Busy	Op	RS			Code:
				S1	S2	RS	
	Add1	No					LD F0 0 R1
	Add2	No					MULTD F4 F0 F2
	Add3	No					SD F4 0 R1
	Mult1	Yes	Multd		R(F2)	Load1	SUBI R1 R1 #8
	Mult2	Yes	Multd		R(F2)	Load2	BNEZ R1 Loop

*Register result status*

Clock	R1	F0	F2	F4	F6	F8	F10	F12	...	F30
7	72	Fu	Load2		Mult2					

- Izadje se i druga MULTD; Registarski fajl potpuno izolovan od iteracije 1 (u F0 se nikad ne upiše rezultat Load1, u F4 se nikad ne upiše rezultat MUL1)

# Loop Example Cycle 8

*Instruction status:*

ITER	Instruction	j	k	Exec Write		Busy	Addr	Fu
				Issue	CompResult			
1	LD	F0	0	R1	1	Load1	Yes	80
1	MULTD	F4	F0	F2	2	Load2	Yes	72
1	SD	F4	0	R1	3	Load3	No	
2	LD	F0	0	R1	6	Store1	Yes	80
2	MULTD	F4	F0	F2	7	Store2	Yes	72
2	SD	F4	0	R1	8	Store3	No	

*Reservation Stations:*

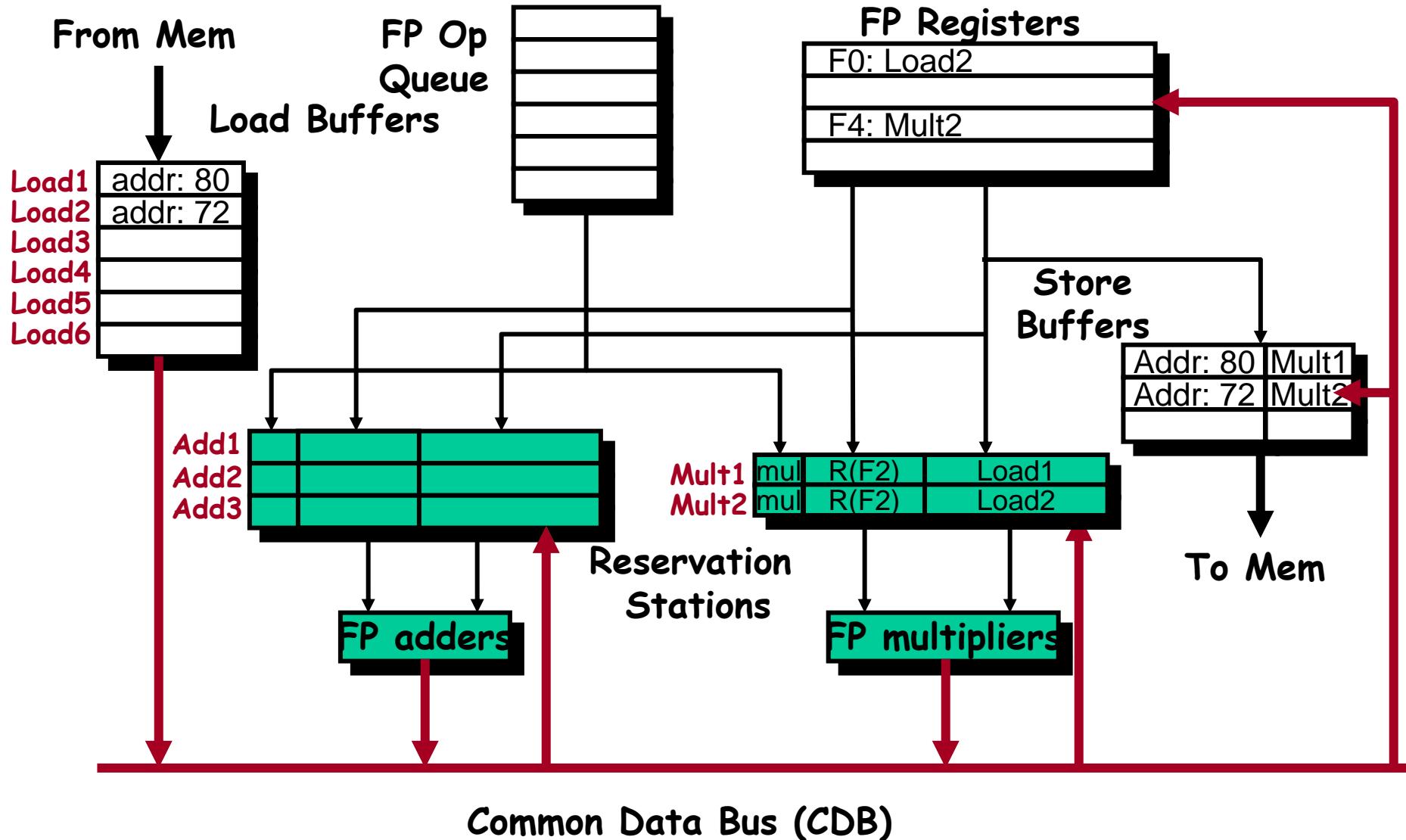
Time	Name	Busy	Op	RS			Code:
				S1	S2	RS	
	Add1	No					LD F0 0 R1
	Add2	No					MULTD F4 F0 F2
	Add3	No					SD F4 0 R1
	Mult1	Yes	Multd		R(F2)	Load1	SUBI R1 R1 #8
	Mult2	Yes	Multd		R(F2)	Load2	BNEZ R1 Loop

*Register result status*

Clock	R1	F0	F2	F4	F6	F8	F10	F12	...	F30
8	72	Fu	Load2		Mult2					

\* Izdata je i druga SD; prva i druga iteracija se potpuno preklapaju

# Šta to fizički znači?



# Loop primer Cycle 9

*Instruction status:*

ITER	Instruction	j	k	Exec Write		Busy	Addr	Fu
				Issue	CompResult			
1	LD	F0	0	R1	1	9	Load1	Yes 80
1	MULTD	F4	F0	F2	2		Load2	Yes 72
1	SD	F4	0	R1	3		Load3	No
2	LD	F0	0	R1	6		Store1	Yes 80 Mult1
2	MULTD	F4	F0	F2	7		Store2	Yes 72 Mult2
2	SD	F4	0	R1	8		Store3	No

*Reservation Stations:*

Time	Name	Busy	Op	S1 S2 RS			Code:
				Vj	Vk	Qj	
	Add1	No					LD F0 0 R1
	Add2	No					MULTD F4 F0 F2
	Add3	No					SD F4 0 R1
	Mult1	Yes	Multd		R(F2)	Load1	SUBI R1 R1 #8
	Mult2	Yes	Multd		R(F2)	Load2	BNEZ R1 Loop

*Register result status*

Clock	R1	F0	F2	F4	F6	F8	F10	F12	...	F30
9	72	Fu	Load2		Mult2					

- \* Load1 okončana: ko čeka na rezultat?
- \* izdavanje SUBI

# Loop primer Cycle 10

*Instruction status:*

ITER	Instruction	j	k	Exec Write			Busy	Addr	Fu
				Issue	Comp	Result			
1	LD	F0	0	R1	1	9	10	Load1	No
1	MULTD	F4	F0	F2	2			Load2	Yes 72
1	SD	F4	0	R1	3			Load3	No
2	LD	F0	0	R1	6	10		Store1	Yes 80 Mult1
2	MULTD	F4	F0	F2	7			Store2	Yes 72 Mult2
2	SD	F4	0	R1	8			Store3	No

*Reservation Stations:*

Time	Name	Busy	Op	S1 S2 RS			Code:
				Vj	Vk	Qj	
	Add1	No					LD F0 0 R1
	Add2	No					MULTD F4 F0 F2
	Add3	No					SD F4 0 R1
4	Mult1	Yes	Multd M[80]	R(F2)			SUBI R1 R1 #8
	Mult2	Yes	Multd		R(F2)	Load2	BNEZ R1 Loop

*Register result status*

Clock	R1	F0	F2	F4	F6	F8	F10	F12	...	F30
10	64	Fu	Load2		Mult2					

\* Load2 okončana: ko čeka rezultat?

\* izdavanje BNEZ

# Loop primer Cycle 11

*Instruction status:*

ITER	Instruction	j	k	Exec Write			Busy	Addr	Fu
				Issue	Comp	Result			
1	LD	F0	0	R1	1	9	10	Load1	No
1	MULTD	F4	F0	F2	2			Load2	No
1	SD	F4	0	R1	3			Load3	Yes 64
2	LD	F0	0	R1	6	10	11	Store1	Yes 80 Mult1
2	MULTD	F4	F0	F2	7			Store2	Yes 72 Mult2
2	SD	F4	0	R1	8			Store3	No

*Reservation Stations:*

Time	Name	Busy	Op	RS			Code:
				S1	S2	RS	
	Add1	No					LD F0 0 R1
	Add2	No					MULTD F4 F0 F2
	Add3	No					SD F4 0 R1
3	Mult1	Yes	Multd M[80] R(F2)				SUBI R1 R1 #8
4	Mult2	Yes	Multd M[72] R(F2)				BNEZ R1 Loop

*Register result status*

Clock	R1	F0	F2	F4	F6	F8	F10	F12	...	F30
11	64	Fu	Load3		Mult2					

\* sledeća load u sekvenci (iz treće iteracije)

# Loop Example Cycle 12

*Instruction status:*

ITER	Instruction	j	k	Exec Write			Busy	Addr	Fu
				Issue	Comp	Result			
1	LD	F0	0	R1	1	9	10	Load1	No
1	MULTD	F4	F0	F2	2			Load2	No
1	SD	F4	0	R1	3			Load3	Yes 64
2	LD	F0	0	R1	6	10	11	Store1	Yes 80 Mult1
2	MULTD	F4	F0	F2	7			Store2	Yes 72 Mult2
2	SD	F4	0	R1	8			Store3	No

*Reservation Stations:*

Time	Name	Busy	Op	RS			Code:
				S1	S2	RS	
	Add1	No					LD F0 0 R1
	Add2	No					MULTD F4 F0 F2
	Add3	No					SD F4 0 R1
2	Mult1	Yes	Multd M[80] R(F2)				SUBI R1 R1 #8
3	Mult2	Yes	Multd M[72] R(F2)				BNEZ R1 Loop

*Register result status*

Clock	R1	F0	F2	F4	F6	F8	F10	F12	...	F30
12	64	Fu	Load3		Mult2					

\* zašto se ne izdaje treća MULD?

# Loop Example Cycle 13

*Instruction status:*

ITER	Instruction	j	k	Exec Write			Busy	Addr	Fu
				Issue	Comp	Result			
1	LD	F0	0	R1	1	9	10	Load1	No
1	MULTD	F4	F0	F2	2			Load2	No
1	SD	F4	0	R1	3			Load3	Yes 64
2	LD	F0	0	R1	6	10	11	Store1	Yes 80 Mult1
2	MULTD	F4	F0	F2	7			Store2	Yes 72 Mult2
2	SD	F4	0	R1	8			Store3	No

*Reservation Stations:*

Time	Name	Busy	Op	RS			Code:
				S1	S2	RS	
	Add1	No					LD F0 0 R1
	Add2	No					MULTD F4 F0 F2 ←
	Add3	No					SD F4 0 R1
1	Mult1	Yes	Multd M[80] R(F2)				SUBI R1 R1 #8
2	Mult2	Yes	Multd M[72] R(F2)				BNEZ R1 Loop

*Register result status*

Clock	R1	F0	F2	F4	F6	F8	F10	F12	...	F30
13	64	Fu	Load3		Mult2					

Mult1 i Mult2 su još aktivne. Zašto se ne izdaje treća SD?

# Loop primer Cycle 14

*Instruction status:*

ITER	Instruction	j	k	Exec Write			Busy	Addr	Fu
				Issue	Comp	Result			
1	LD	F0	0	R1	1	9	10	Load1	No
1	MULTD	F4	F0	F2	2	14		Load2	No
1	SD	F4	0	R1	3			Load3	Yes 64
2	LD	F0	0	R1	6	10	11	Store1	Yes 80 Mult1
2	MULTD	F4	F0	F2	7			Store2	Yes 72 Mult2
2	SD	F4	0	R1	8			Store3	No

*Reservation Stations:*

Time	Name	Busy	Op	RS			Code:
				S1	S2	RS	
	Add1	No					LD F0 0 R1
	Add2	No					MULTD F4 F0 F2
	Add3	No					SD F4 0 R1
0	Mult1	Yes	Multd M[80] R(F2)				SUBI R1 R1 #8
1	Mult2	Yes	Multd M[72] R(F2)				BNEZ R1 Loop

*Register result status*

Clock	R1	F0	F2	F4	F6	F8	F10	F12	...	F30
14	64	Fu	Load3		Mult2					

\* Mult1 okončana. ko čeka?

# Loop primer Cycle 15

*Instruction status:*

ITER	Instruction	j	k	Exec Write			Busy	Addr	Fu
				Issue	Comp	Result			
1	LD	F0	0	R1	1	9	10	Load1	No
1	MULTD	F4	F0	F2	2	14	15	Load2	No
1	SD	F4	0	R1	3			Load3	Yes 64
2	LD	F0	0	R1	6	10	11	Store1	Yes 80 [80]*R2
2	MULTD	F4	F0	F2	7	15		Store2	Yes 72 Mult2
2	SD	F4	0	R1	8			Store3	No

*Reservation Stations:*

Time	Name	Busy	Op	RS			Code:
				S1	S2	RS	
	Add1	No					LD F0 0 R1
	Add2	No					MULTD F4 F0 F2
	Add3	No					SD F4 0 R1
	Mult1	No					SUBI R1 R1 #8
0	Mult2	Yes	Multd	M[72]	R(F2)		BNEZ R1 Loop

*Register result status*

Clock	R1	F0	F2	F4	F6	F8	F10	F12	...	F30
15	64	Fu	Load3		Mult2					

- \* Mult1 slobodna;
- \* Mult2 okončana. ko čeka?

# Loop primer Cycle 16

*Instruction status:*

ITER	Instruction	j	k	Exec Write			Busy	Addr	Fu
				Issue	Comp	Result			
1	LD	F0	0	R1	1	9	10	Load1	No
1	MULTD	F4	F0	F2	2	14	15	Load2	No
1	SD	F4	0	R1	3			Load3	Yes 64
2	LD	F0	0	R1	6	10	11	Store1	Yes 80 [80]*R2
2	MULTD	F4	F0	F2	7	15	16	Store2	Yes 72 [72]*R2
2	SD	F4	0	R1	8			Store3	No

*Reservation Stations:*

Time	Name	Busy	Op	S1	S2	RS	Code:
				Vj	Vk	Qj	
	Add1	No					LD F0 0 R1
	Add2	No					MULTD F4 F0 F2 ←
	Add3	No					SD F4 0 R1
	Mult1	Yes	Multd		R(F2)	Load3	SUBI R1 R1 #8
	Mult2	No					BNEZ R1 Loop

*Register result status*

Clock	R1	F0	F2	F4	F6	F8	F10	F12	...	F30
16	64	Fu	Load3		Mult1					

\*MULTD iz treće iteracije se izdaje (FU Mult1 aktivna)

# Loop primer Cycle 17

*Instruction status:*

ITER	Instruction	j	k	Exec Write			Busy	Addr	Fu
				Issue	Comp	Result			
1	LD	F0	0	R1	1	9	10	Load1	No
1	MULTD	F4	F0	F2	2	14	15	Load2	No
1	SD	F4	0	R1	3			Load3	Yes 64
2	LD	F0	0	R1	6	10	11	Store1	Yes 80 [80]*R2
2	MULTD	F4	F0	F2	7	15	16	Store2	Yes 72 [72]*R2
2	SD	F4	0	R1	8			Store3	Yes 64 Mult1

*Reservation Stations:*

Time	Name	Busy	Op	S1	S2	RS	Code:
				Vj	Vk	Qj	
	Add1	No					LD F0 0 R1
	Add2	No					MULTD F4 F0 F2
	Add3	No					SD F4 0 R1 ←
	Mult1	Yes	Multd		R(F2)	Load3	SUBI R1 R1 #8
	Mult2	No					BNEZ R1 Loop

*Register result status*

Clock	R1	F0	F2	F4	F6	F8	F10	F12	...	F30
17	64	Fu	Load3		Mult1					

Izdaje se treća SD

# Loop primer Cycle 18

*Instruction status:*

ITER	Instruction	j	k	Exec Write			Busy	Addr	Fu
				Issue	Comp	Result			
1	LD	F0	0	R1	1	9	10	Load1	No
1	MULTD	F4	F0	F2	2	14	15	Load2	No
1	SD	F4	0	R1	3	18		Load3	Yes 64
2	LD	F0	0	R1	6	10	11	Store1	Yes 80 [80]*R2
2	MULTD	F4	F0	F2	7	15	16	Store2	Yes 72 [72]*R2
2	SD	F4	0	R1	8			Store3	Yes 64 Mult1

*Reservation Stations:*

Time	Name	Busy	Op	S1	S2	RS	Code:
				Vj	Vk	Qj	
	Add1	No					LD F0 0 R1
	Add2	No					MULTD F4 F0 F2
	Add3	No					SD F4 0 R1
	Mult1	Yes	Multd		R(F2)	Load3	SUBI R1 R1 #8
	Mult2	No					BNEZ R1 Loop

*Register result status*

Clock	R1	F0	F2	F4	F6	F8	F10	F12	...	F30
18	64	Fu	Load3		Mult1					

# Loop primer Cycle 19

*Instruction status:*

ITER	Instruction	j	k	Exec Write			Busy	Addr	Fu
				Issue	Comp	Result			
1	LD	F0	0	R1	1	9	10	Load1	No
1	MULTD	F4	F0	F2	2	14	15	Load2	No
1	SD	F4	0	R1	3	18	19	Load3	Yes 64
2	LD	F0	0	R1	6	10	11	Store1	No
2	MULTD	F4	F0	F2	7	15	16	Store2	Yes 72 [72]*R2
2	SD	F4	0	R1	8	19		Store3	Yes 64 Mult1

*Reservation Stations:*

Time	Name	Busy	Op	S1	S2	RS	Code:
				Vj	Vk	Qj	
	Add1	No					LD F0 0 R1
	Add2	No					MULTD F4 F0 F2
	Add3	No					SD F4 0 R1
	Mult1	Yes	Multd		R(F2)	Load3	SUBI R1 R1 #8
	Mult2	No					BNEZ R1 Loop

*Register result status*

Clock	R1	F0	F2	F4	F6	F8	F10	F12	...	F30
19	64	Fu	Load3		Mult1					

# Loop primer Cycle 20

*Instruction status:*

ITER	Instruction	j	k	Exec Write			Busy	Addr	Fu
				Issue	Comp	Result			
1	LD	F0	0	R1	1	9	10	Load1	No
1	MULTD	F4	F0	F2	2	14	15	Load2	No
1	SD	F4	0	R1	3	18	19	Load3	Yes 64
2	LD	F0	0	R1	6	10	11	Store1	No
2	MULTD	F4	F0	F2	7	15	16	Store2	No
2	SD	F4	0	R1	8	19	20	Store3	Yes 64 Mult1

*Reservation Stations:*

Time	Name	Busy	Op	S1	S2	RS	Code:
				Vj	Vk	Qj	
	Add1	No					LD F0 0 R1
	Add2	No					MULTD F4 F0 F2
	Add3	No					SD F4 0 R1
	Mult1	Yes	Multd		R(F2)	Load3	SUBI R1 R1 #8
	Mult2	No					BNEZ R1 Loop

*Register result status*

Clock	R1	F0	F2	F4	F6	F8	F10	F12	...	F30
20	64	Fu	Load3		Mult1					

Ponovo imamo izdavanje po redosledu, izvršenje i okončanje van redosleda pribavljanja

# Zašto Tomasulo može da preklapa iteracije petlje?

## \* Preimenovanje registara

- Različite iteracije koriste različite fizičke destinacije za registre (dinamičko odmotavanje petlje)
- imena registara zamjenjena imenima RS koje generišu rezultat
- efektivno povećava veličinu registarskog fajla