

- Paralelizam unutar procesora
 - Dinamička preraspodela naredbi
 - Scoreboarding - prati se svaka naredba koja treba da se (uskoro) izvrši i njeno izvršavanje se pokreće samo ako su svi ulazni operandi dostupni (operand je dostupan ako nijedna od naredbi na čekanju ne upisuje u njega), što rešava RAW opasnosti
 - Preimenovanje registara (register renaming)
 - omogućava eliminaciju WAR i WAW opasnosti
 - zahteva postojanje dodatnih internih registara koji će se koristiti samo za ovo (današnji procesori ih mogu imati više desetina ili stotina)

- Paralelizam unutar procesora

- Dinamička preraspodela naredbi

- Preimenovanje registara (register renaming)

- primer

- `fdiv.d f0, f2, f4`

- `fadd.d f6, f0, f8`

- `fsd f6, 0(x1)`

- `fsub.d f8, f10, f14`

- `fmul.d f6, f10, f8`

- `fdiv.d f0, f2, f4`

- `fadd.d S, f0, f8`

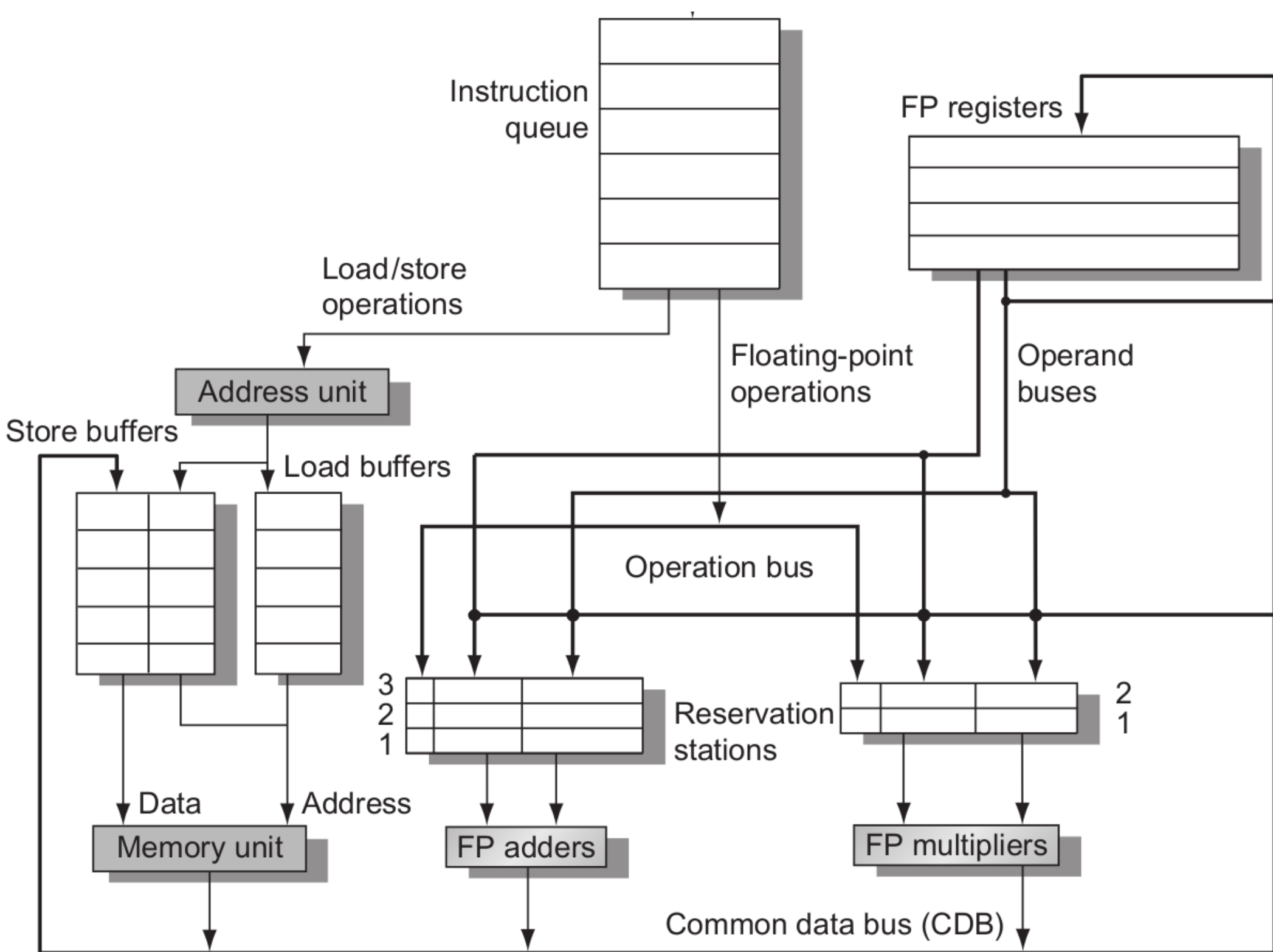
- `fsd S, 0(x1)`

- `fsub.d T, f10, f14`

- `fmul.d f6, f10, T`

- antizavisnost (WAR) između fadd i fsub, i između fsd i fmul
 - izlazna zavisnost između fadd i fmul
 - prave zavisnosti između fdiv i fadd, između fsub i fmul, i između fadd i fsd
 - ako imamo dva dodatna registra , S i T, registri se mogu preimenovati, odnosno zameniti drugim, čime se dobija sekvenca bez zavisnosti (pri čemu se svaka dalja upotreba f8 za čitanje mora zameniti sa T, do prve naredbe koja mu menja vrednost)
 - ovo je nešto što može da odradi kompajler (statički)

- Paralelizam unutar procesora
 - Dinamička preraspodela naredbi
 - Tomasulov algoritam (Robert Tomasulo, IBM 360/91, 1966)
 - algoritam se koristi i u današnjim procesorima, u nešto komplikovanijoj varijanti
 - dinamičko preimenovanje registara koristeći rezervacione stanice (reservation stations), koje baferuju operande naredbi koje treba da se izvrše
 - glavna ideja je da rezervaciona stanica dobavlja i baferuje (čuva) vrednost operanda čim on postane dostupan, čime se eliminiše njegovo preuzimanje iz registara
 - više uzastopnih upisa u registar se razrešava tako da se stvarno obavi samo poslednji
 - rezervacionih stanica može biti više nego registara, i svaka je vezana za jednu ili više funkcionalnih jedinica (jedinice koje obavljaju naredbe)
 - rezultati se iz rezervacionih stanica prosleđuju direktno u funkcionalne jedinice, čime se zaobilaze registri, koristeći zajedničku magistralu podataka (common data bus - CDB)



- Paralelizam unutar procesora
 - Dinamička preraspodela naredbi
 - Tomasulov algoritam
 - delovi FPU procesora:
 - red naredbi (instruction queue) - naredbe koje treba da se izvrše
 - ulazni baferi (load buffers) - podaci koji dolaze iz memorije
 - izlazni baferi (store buffers) - podaci koji odlaze u memoriju
 - registri (FP registers) - floating point registri
 - rezervacione stanice (reservation stations) - ovde se baferuje operacija i operandi; kada operandi postanu dostupni, operacija odlazi u funkcionalnu jedinicu (FP adder za sabiranje/oduzimanje, odnosno FP multiplier za množenje/deljenje); ako nema slobodne funkcionalne jedinice, izvršavanje naredbe mora da čeka
 - sve je povezano sa zajedničkom magistralom (CDB); magistrala za svaki podatak radi objavljivanje (broadcast) i podatak, zajedno sa oznakom stanice koja ga je proizvela, ide istovremeno na sva mesta gde je potreban

- Paralelizam unutar procesora
 - Dinamička preraspodela naredbi
 - Tomasulov algoritam
 - sve naredbe prvo odlaze u red naredbi, u redosledu u kome treba da se izvrše
 - faze izvršavanja naredbe
 - započinjanje (issue): naredba se preuzima se iz reda naredbi i gleda da li ima slobodna rezervaciona stanica za njen tip
 - ako nema, naredba mora da čeka da se neka oslobodi
 - ako ima, ubacuje se u stanicu
 - ako je operand dostupan, upisuje se njegova vrednost, a ako nije, upisuje se rezervaciona stanica koja treba da ga proizvede (ovo je praktično preimenovanje registara i rešava WAW i WAR)
 - izvršavanje (execute): izvršavanje naredbe
 - ako neki od operandi nije dostupan, prati se CDB i čeka da bude objavljen
 - kada svi operandi postanu dostupni naredba se prebacuje u funkcionalnu jedinicu (ako ima slobodna), čime se praktično rešava RAW

- Paralelizam unutar procesora
 - Dinamička preraspodela naredbi
 - Tomasulov algoritam
 - faze izvršavanja naredbe
 - izvršavanje (execute):
 - ako se desi da istovremeno ima više spremnih FP naredbi, bira se jedna od njih
 - load i store se moraju raditi iz dva koraka: računanje adrese (sabiranje konstante i baznog registra) i smeštanje te adrese u bafer
 - load se izvršava čim ima slobodna memorijska jedinica
 - store mora čekati da vrednost postane dostuna, pa tek onda može u memorijsku jedinicu
 - ako je pre neke naredbe naredba grananja, ne može se ulaziti u njenu fazu izvršavanja dok se ne sazna kako će se grananje izvršiti; ovime se čuva redosled izvršavanja izuzetaka
 - ako procesor ima predviđanje grananja, tada se naredba može izvršiti i ranije (što je komplikovanija varijanta), pa će se zaustaviti na fazi upisa rezultata

- Paralelizam unutar procesora
 - Dinamička preraspodela naredbi
 - Tomasulov algoritam
 - faze izvršavanja naredbe
 - upis rezultata (write result): kada se izvršavanje završi, rezultat se šalje na CDB, što za posledicu ima da rezultat odlazi do svih rezervacionih stanica, registara i store jedinica kojima treba
 - kada naredba jednom krene da se izvršava, ako joj nedostaje ulazni operand, čekaće ga od rezervacione stanice čija je oznaka upisana za taj operand
 - ako je operand bio raspoloživ kada je naredba ulazila u rezervacionu stanicu, njegova vrednost se odmah kopira tamo, pa se nadalje za tu operaciju ne koristi registar

- Paralelizam unutar procesora
 - Dinamička preraspodela naredbi
 - Tomasulov algoritam
 - polja rezervacione stanice
 - Op - operacija koja treba da se izvrši
 - Qj,Qk - oznaka (tag) rezervacione stanice koja treba da obezbedi prvi, odnosno drugi ulazni operand; 0 (ili prazno) znači da je operand već dostupan i smešten u Vj, odnosno Vk
 - Vj,Vk - vrednosti ulaznih operanada (za svaki operand je validno ili Q ili V)
 - A - efektivna (izračunata) adresa, za load i store
 - Busy - oznaka da je rezervaciona stanica zauzeta
 - svakom registru se dodeljuje polje Qi, koje je oznaka rezervacione stanice čiji rezultat tu treba da se upiše; 0 (ili prazno) znači da je sadržaj registra validan
 - load/store baferi sadrže polje A koje predstavlja efektivnu adresu

- Paralelizam unutar procesora
 - Dinamička preraspodela naredbi
 - Tomasulov algoritam
 - primer
 1. fld f6, 32(x2)
 2. fld f2, 44(x3)
 3. fmul.d f0, f2, f4
 4. fsub.d f8, f2, f6
 5. fdiv.d f10, f0, f6
 6. fadd.d f6, f8, f2
 - broj uz Add, Mult i Load je redni broj rezervacione stanice u koju se smešta naredba

Instruction status			
Instruction	Issue	Execute	Write result
f1d f6, 32(x2)	X		
f1d f2, 44(x3)			
fmul.d f0, f2, f4			
fsub.d f8, f2, f6			
fdiv.d f10, f0, f6			
fadd.d f6, f8, f2			

32+x2 - adresa lokacije dobijena adresiranjem 32(x2)
M(32+x2) - sadržaj memorije na adresi određenoj sa 32(x2)
rSUB,rADD,rMUL,rDIV - vrednost dobijena izvršavanjem naredbe

Reservation stations							
Tag	Busy	Op	Qj	Vj	Qk	Vk	A
Load1	X	Load					32+x2
Load2							
Add1							
Add2							
Add3							
Mult1							
Mult2							

Register status									
Field	f0	f2	f4	f6	f8	f10	f12	...	f30
Qi				Load1					

Instruction status			
Instruction	Issue	Execute	Write result
f1d f6, 32(x2)	X		
f1d f2, 44(x3)	X		
fmul.d f0, f2, f4			
fsub.d f8, f2, f6			
fdiv.d f10, f0, f6			
fadd.d f6, f8, f2			

32+x2 - adresa lokacije dobijena adresiranjem 32(x2)
M(32+x2) - sadržaj memorije na adresi određenoj sa 32(x2)
rSUB,rADD,rMUL,rDIV - vrednost dobijena izvršavanjem naredbe

Reservation stations							
Tag	Busy	Op	Qj	Vj	Qk	Vk	A
Load1	X	Load					32+x2
Load2	X	Load					44+x3
Add1							
Add2							
Add3							
Mult1							
Mult2							

Register status									
Field	f0	f2	f4	f6	f8	f10	f12	...	f30
Qi		Load2		Load1					

Instruction status			
Instruction	Issue	Execute	Write result
f1d f6, 32(x2)	X		
f1d f2, 44(x3)	X		
fmul.d f0, f2, f4	X		
fsub.d f8, f2, f6			
fdiv.d f10, f0, f6			
fadd.d f6, f8, f2			

32+x2 - adresa lokacije dobijena adresiranjem 32(x2)
M(32+x2) - sadržaj memorije na adresi određenoj sa 32(x2)
rSUB,rADD,rMUL,rDIV - vrednost dobijena izvršavanjem naredbe

Reservation stations							
Tag	Busy	Op	Qj	Vj	Qk	Vk	A
Load1	X	Load					32+x2
Load2	X	Load					44+x3
Add1							
Add2							
Add3							
Mult1	X	MUL	Load2			f4	
Mult2							

Register status									
Field	f0	f2	f4	f6	f8	f10	f12	...	f30
Qi	Mult1	Load2		Load1					

Instruction status			
Instruction	Issue	Execute	Write result
f1d f6, 32(x2)	X		
f1d f2, 44(x3)	X		
fmul.d f0, f2, f4	X		
fsub.d f8, f2, f6	X		
fdiv.d f10, f0, f6			
fadd.d f6, f8, f2			

32+x2 - adresa lokacije dobijena adresiranjem 32(x2)
M(32+x2) - sadržaj memorije na adresi određenoj sa 32(x2)
rSUB,rADD,rMUL,rDIV - vrednost dobijena izvršavanjem naredbe

Reservation stations							
Tag	Busy	Op	Qj	Vj	Qk	Vk	A
Load1	X	Load					32+x2
Load2	X	Load					44+x3
Add1	X	SUB	Load2		Load1		
Add2							
Add3							
Mult1	X	MUL	Load2			f4	
Mult2							

Register status									
Field	f0	f2	f4	f6	f8	f10	f12	...	f30
Qi	Mult1	Load2		Load1	Add1				

Instruction status			
Instruction	Issue	Execute	Write result
f1d f6, 32(x2)	X		
f1d f2, 44(x3)	X		
fmul.d f0, f2, f4	X		
fsub.d f8, f2, f6	X		
fdiv.d f10, f0, f6	X		
fadd.d f6, f8, f2			

32+x2 - adresa lokacije dobijena adresiranjem 32(x2)
M(32+x2) - sadržaj memorije na adresi određenoj sa 32(x2)
rSUB,rADD,rMUL,rDIV - vrednost dobijena izvršavanjem naredbe

Reservation stations							
Tag	Busy	Op	Qj	Vj	Qk	Vk	A
Load1	X	Load					32+x2
Load2	X	Load					44+x3
Add1	X	SUB	Load2		Load1		
Add2							
Add3							
Mult1	X	MUL	Load2			f4	
Mult2	X	DIV	Mult1		Load1		

Register status									
Field	f0	f2	f4	f6	f8	f10	f12	...	f30
Qi	Mult1	Load2		Load1	Add1	Mult2			

Instruction status			
Instruction	Issue	Execute	Write result
f1d f6, 32(x2)	X		
f1d f2, 44(x3)	X		
fmul.d f0, f2, f4	X		
fsub.d f8, f2, f6	X		
fdiv.d f10, f0, f6	X		
fadd.d f6, f8, f2	X		

32+x2 - adresa lokacije dobijena adresiranjem 32(x2)
M(32+x2) - sadržaj memorije na adresi određenoj sa 32(x2)
rSUB,rADD,rMUL,rDIV - vrednost dobijena izvršavanjem naredbe

Reservation stations							
Tag	Busy	Op	Qj	Vj	Qk	Vk	A
Load1	X	Load					32+x2
Load2	X	Load					44+x3
Add1	X	SUB	Load2		Load1		
Add2	X	ADD	Add1		Load2		
Add3							
Mult1	X	MUL	Load2			f4	
Mult2	X	DIV	Mult1		Load1		

Register status									
Field	f0	f2	f4	f6	f8	f10	f12	...	f30
Qi	Mult1	Load2		Add2	Add1	Mult2			

Instruction status			
Instruction	Issue	Execute	Write result
f1d f6, 32(x2)	X	X	
f1d f2, 44(x3)	X		
fmul.d f0, f2, f4	X		
fsub.d f8, f2, f6	X		
fdiv.d f10, f0, f6	X		
fadd.d f6, f8, f2	X		

32+x2 - adresa lokacije dobijena adresiranjem 32(x2)
M(32+x2) - sadržaj memorije na adresi određenoj sa 32(x2)
rSUB,rADD,rMUL,rDIV - vrednost dobijena izvršavanjem naredbe

Reservation stations							
Tag	Busy	Op	Qj	Vj	Qk	Vk	A
Load1	X	Load					32+x2
Load2	X	Load					44+x3
Add1	X	SUB	Load2		Load1		
Add2	X	ADD	Add1		Load2		
Add3							
Mult1	X	MUL	Load2			f4	
Mult2	X	DIV	Mult1		Load1		

Register status									
Field	f0	f2	f4	f6	f8	f10	f12	...	f30
Qi	Mult1	Load2		Add2	Add1	Mult2			

Instruction status			
Instruction	Issue	Execute	Write result
f1d f6, 32(x2)	X	X	X
f1d f2, 44(x3)	X	X	
fmul.d f0, f2, f4	X		
fsub.d f8, f2, f6	X		
fdiv.d f10, f0, f6	X		
fadd.d f6, f8, f2	X		

32+x2 - adresa lokacije dobijena adresiranjem 32(x2)
M(32+x2) - sadržaj memorije na adresi određenoj sa 32(x2)
rSUB,rADD,rMUL,rDIV - vrednost dobijena izvršavanjem naredbe

Reservation stations							
Tag	Busy	Op	Qj	Vj	Qk	Vk	A
Load1							
Load2	X	Load					44+x3
Add1	X	SUB	Load2			M(32+x2)	
Add2	X	ADD	Add1		Load2		
Add3							
Mult1	X	MUL	Load2			f4	
Mult2	X	DIV	Mult1			M(32+x2)	

Register status									
Field	f0	f2	f4	f6	f8	f10	f12	...	f30
Qi	Mult1	Load2		Add2	Add1	Mult2			

Instruction status			
Instruction	Issue	Execute	Write result
f1d f6, 32(x2)	X	X	X
f1d f2, 44(x3)	X	X	X
fmul.d f0, f2, f4	X	X	
fsub.d f8, f2, f6	X		
fdiv.d f10, f0, f6	X		
fadd.d f6, f8, f2	X		

32+x2 - adresa lokacije dobijena adresiranjem 32(x2)
M(32+x2) - sadržaj memorije na adresi određenoj sa 32(x2)
rSUB,rADD,rMUL,rDIV - vrednost dobijena izvršavanjem naredbe

Reservation stations							
Tag	Busy	Op	Qj	Vj	Qk	Vk	A
Load1							
Load2							
Add1	X	SUB		M(44+x3)		M(32+x2)	
Add2	X	ADD	Add1			M(44+x3)	
Add3							
Mult1	X	MUL		M(44+x3)		f4	
Mult2	X	DIV	Mult1			M(32+x2)	

Register status									
Field	f0	f2	f4	f6	f8	f10	f12	...	f30
Qi	Mult1	M(44+x3)		Add2	Add1	Mult2			

Instruction status			
Instruction	Issue	Execute	Write result
f1d f6, 32(x2)	X	X	X
f1d f2, 44(x3)	X	X	X
fmul.d f0, f2, f4	X	X	
fsub.d f8, f2, f6	X	X	
fdiv.d f10, f0, f6	X		
fadd.d f6, f8, f2	X		

32+x2 - adresa lokacije dobijena adresiranjem 32(x2)
M(32+x2) - sadržaj memorije na adresi određenoj sa 32(x2)
rSUB,rADD,rMUL,rDIV - vrednost dobijena izvršavanjem naredbe

Reservation stations							
Tag	Busy	Op	Qj	Vj	Qk	Vk	A
Load1							
Load2							
Add1	X	SUB		M(44+x3)		M(32+x2)	
Add2	X	ADD	Add1			M(44+x3)	
Add3							
Mult1	X	MUL		M(44+x3)		f4	
Mult2	X	DIV	Mult1			M(32+x2)	

Register status									
Field	f0	f2	f4	f6	f8	f10	f12	...	f30
Qi	Mult1	M(44+x3)		Add2	Add1	Mult2			

Instruction status			
Instruction	Issue	Execute	Write result
f1d f6, 32(x2)	X	X	X
f1d f2, 44(x3)	X	X	X
fmul.d f0, f2, f4	X	X	
fsub.d f8, f2, f6	X	X	X
fdiv.d f10, f0, f6	X	X	
fadd.d f6, f8, f2	X		

32+x2 - adresa lokacije dobijena adresiranjem 32(x2)
M(32+x2) - sadržaj memorije na adresi određenoj sa 32(x2)
rSUB,rADD,rMUL,rDIV - vrednost dobijena izvršavanjem naredbe

Reservation stations							
Tag	Busy	Op	Qj	Vj	Qk	Vk	A
Load1							
Load2							
Add1							
Add2	X	ADD		rSUB		M(44+x3)	
Add3							
Mult1	X	MUL		M(44+x3)		f4	
Mult2	X	DIV	Mult1			M(32+x2)	

Register status									
Field	f0	f2	f4	f6	f8	f10	f12	...	f30
Qi	Mult1	M(44+x3)		Add2	rSUB	Mult2			

Instruction status			
Instruction	Issue	Execute	Write result
f1d f6, 32(x2)	X	X	X
f1d f2, 44(x3)	X	X	X
fmul.d f0, f2, f4	X	X	
fsub.d f8, f2, f6	X	X	X
fdiv.d f10, f0, f6	X	X	
fadd.d f6, f8, f2	X	X	

32+x2 - adresa lokacije dobijena adresiranjem 32(x2)
M(32+x2) - sadržaj memorije na adresi određenoj sa 32(x2)
rSUB,rADD,rMUL,rDIV - vrednost dobijena izvršavanjem naredbe

Reservation stations							
Tag	Busy	Op	Qj	Vj	Qk	Vk	A
Load1							
Load2							
Add1							
Add2	X	ADD		rSUB		M(44+x3)	
Add3							
Mult1	X	MUL		M(44+x3)		f4	
Mult2	X	DIV	Mult1			M(32+x2)	

Register status									
Field	f0	f2	f4	f6	f8	f10	f12	...	f30
Qi	Mult1	M(44+x3)		Add2	rSUB	Mult2			

Instruction status			
Instruction	Issue	Execute	Write result
f1d f6, 32(x2)	X	X	X
f1d f2, 44(x3)	X	X	X
fmul.d f0, f2, f4	X	X	
fsub.d f8, f2, f6	X	X	X
fdiv.d f10, f0, f6	X	X	
fadd.d f6, f8, f2	X	X	X

32+x2 - adresa lokacije dobijena adresiranjem 32(x2)
M(32+x2) - sadržaj memorije na adresi određenoj sa 32(x2)
rSUB,rADD,rMUL,rDIV - vrednost dobijena izvršavanjem naredbe

Reservation stations							
Tag	Busy	Op	Qj	Vj	Qk	Vk	A
Load1							
Load2							
Add1							
Add2							
Add3							
Mult1	X	MUL		M(44+x3)		f4	
Mult2	X	DIV	Mult1			M(32+x2)	

Register status									
Field	f0	f2	f4	f6	f8	f10	f12	...	f30
Qi	Mult1	M(44+x3)		rADD	rSUB	Mult2			

Instruction status			
Instruction	Issue	Execute	Write result
f1d f6, 32(x2)	X	X	X
f1d f2, 44(x3)	X	X	X
fmul.d f0, f2, f4	X	X	X
fsub.d f8, f2, f6	X	X	X
fdiv.d f10, f0, f6	X	X	
fadd.d f6, f8, f2	X	X	X

32+x2 - adresa lokacije dobijena adresiranjem 32(x2)
M(32+x2) - sadržaj memorije na adresi određenoj sa 32(x2)
rSUB,rADD,rMUL,rDIV - vrednost dobijena izvršavanjem naredbe

Reservation stations							
Tag	Busy	Op	Qj	Vj	Qk	Vk	A
Load1							
Load2							
Add1							
Add2							
Add3							
Mult1							
Mult2	X	DIV		rMUL		M(32+x2)	

Register status									
Field	f0	f2	f4	f6	f8	f10	f12	...	f30
Qi	rMUL	M(44+x3)		rADD	rSUB	Mult2			

Instruction status			
Instruction	Issue	Execute	Write result
f1d f6, 32(x2)	X	X	X
f1d f2, 44(x3)	X	X	X
fmul.d f0, f2, f4	X	X	X
fsub.d f8, f2, f6	X	X	X
fdiv.d f10, f0, f6	X	X	X
fadd.d f6, f8, f2	X	X	X

32+x2 - adresa lokacije dobijena adresiranjem 32(x2)
M(32+x2) - sadržaj memorije na adresi određenoj sa 32(x2)
rSUB,rADD,rMUL,rDIV - vrednost dobijena izvršavanjem naredbe

Reservation stations							
Tag	Busy	Op	Qj	Vj	Qk	Vk	A
Load1							
Load2							
Add1							
Add2							
Add3							
Mult1							
Mult2							

Register status									
Field	f0	f2	f4	f6	f8	f10	f12	...	f30
Qi	rMUL	M(44+x3)		rADD	rSUB	rDIV			

- Paralelizam unutar procesora
 - Dinamička preraspodela naredbi
 - Tomasulov algoritam
 - osnovne prednosti
 - detekcija opasnosti po podatke se distribuira na više mesta
 - proizilazi iz postojanja više rezervacionih stanica i zajedničke magistrale
 - eliminacija WAW i WAR opasnosti
 - proizilazi iz preimenovanja registara (zamenjuju se oznakama rezervacionih stanica) i smeštanja vrednosti operanada tamo gde treba istog momenta kada postanu dostupni
 - primer: fdiv i fadd imaju WAR oko f6 ako se pokrenu istovremeno:
 - ako se naredba koja obezbeđuje f6 za fdiv nije završila, polje Qk za fdiv se puni sa Load1
 - ako se naredba koja obezbeđuje f6 za fdiv završila, polje Vk za fdiv se odmah puni vrednošću f6
 - u oba slučaja, fdiv postaje nezavisno od fadd

- Paralelizam unutar procesora
 - Dinamička preraspodela naredbi
 - Tomasulov algoritam - detaljnije
 - započinjanje FP naredbe, može samo ako postoji slobodna RS
 - ako je Q_i registra za operand prazno, preuzmi vrednost i stavi u V_j/V_k , a Q_j/Q_k postavi na 0
 - ako Q_i registra za operand nije prazno, Q_j/Q_k postavi na vrednost Q_i
 - postavi Busy RS na tačno, a u Q_i izlaznog registra upiši oznaku RS
 - započinjanje Load/Store naredbe, može samo ako postoji slobodan Load odnosno Store bafer
 - ako je Q_i registra za adresiranje prazno, preuzmi njegovu vrednost u V_j , a Q_j postavi na 0
 - ako Q_i registra za adresiranje nije prazno, Q_j postavi na vrednost Q_i
 - postavi konstantu adresiranja u polje A i Busy za RS na tačno
 - za Load, postavi Q_i registra za podatak na oznaku RS
 - za Store, ako je Q_i registra za podatak prazno postavi V_k na vrednost registra a Q_k na 0, a u suprotnom postavi Q_k na Q_i

- Paralelizam unutar procesora
 - Dinamička preraspodela naredbi
 - Tomasulov algoritam - detaljnije
 - izvršavanje FP naredbe, može samo ako su Q_j i Q_k jednaki 0
 - slanje naredbe sa operandima V_j i V_k na izvršavanje
 - izvršavanje Load/Store naredbe, može samo ako je Q_j jednako 0 i ako je naredba sledeća u load/store baferima
 - u polje A se upisuje izračunata adresa operanda ($V_j + A$)
 - samo za Load: čita se sadržaj lokacije sa adrese A
 - upis rezultata FP ili Load naredbe, može samo ako je izvršavanje RS završeno i ako je CDB slobodan
 - za svaki registar koji ima oznaku RS upisuje se vrednost, a Q_i se postavlja na 0
 - za svaku RS koja u Q_j/Q_k ima oznaku RS koja se završila, upisuje se rezultat u V_j/V_k , a Q_j/Q_k se postavlja na 0
 - Busy se postavlja na netačno
 - upis rezultata za Store, može samo ako je izvršavanje RS završeno i ako je Q_k jednako 0
 - u memoriju na adresu A se upisje V_k
 - Busy se postavlja na netačno

- Paralelizam unutar procesora

- Dinamička preraspodela naredbi

- Tomasulov algoritam

- primer sa petljom: množenje elemenata niza skalarom

```
Loop: fld      f0, 0(x1)
      fmul.d   f4, f0, f2
      fsd      f4, 0(x1)
      addi     x1, x1, -8
      bne      x1, x2, Loop
```

- ako predviđanje skokova odredi da će se skok desiti, tada korišćenjem rezervacionih stanica možemo pokrenuti više tela petlji istovremeno - dinamičko odmotavanje petlje!
 - pretpostavimo da su započete naredbe iz dva ciklusa petlje, ali da se nijedna od naredbi nije završila (napomena: u tabeli nije prikazana addi naredba, ali se smatra da je odrađena između prvog i drugog ciklusa petlje)

Instruction		Instruction status			
		From iteration	Issue	Execute	Write result
f1d	f0,0(x1)	1	✓	✓	
fmul.d	f4,f0,f2	1	✓		
fsd	f4,0(x1)	1	✓		
f1d	f0,0(x1)	2	✓	✓	
fmul.d	f4,f0,f2	2	✓		
fsd	f4,0(x1)	2	✓		

Reservation stations							
Name	Busy	Op	Vj	Vk	Qj	Qk	A
Load1	Yes	Load					Regs[x1] + 0
Load2	Yes	Load					Regs[x1] – 8
Add1	No						
Add2	No						
Add3	No						
Mult1	Yes	MUL		Regs[f2]	Load1		
Mult2	Yes	MUL		Regs[f2]	Load2		
Store1	Yes	Store	Regs[x1]			Mult1	
Store2	Yes	Store	Regs[x1] – 8			Mult2	

Register status									
Field	f0	f2	f4	f6	f8	f10	f12	...	f30
Qi	Load2		Mult2						

- Paralelizam unutar procesora
 - Dinamička preraspodela naredbi
 - Tomasulov algoritam
 - primer sa petljom: množenje elemenata niza skalarom
 - load i store operacije se mogu bezbedno izvršiti van redosleda ako pristupaju različitim adresama
 - ako koriste istu adresu, može se desiti WAR ili RAW
 - dve store operacije na istu adresu mogu rezultovati WAW opasnošću
 - da bi proverilo da li se load može izvršiti, procesor može proveriti da li postoji nezavršena store naredba koja prethodi tom load-u, a koja koristi istu adresu
 - da bi proverio da li se store može izvršiti, procesor može proveriti da li postoji nezavršena load ili store naredba koja prethodi tom store-u, a koja koristi istu adresu
 - za ove provere, sve prethodne efektivne adrese moraju biti sračunate u redosledu pojavljivanja u kodu
 - za load naredbe se proverava da li postoji ista adresa u store baferima; ako postoji, load naredba se ne šalje u load bafer sve dok se store ne završi
 - za store je slično, samo se proveravaju i load i store baferi

- Paralelizam unutar procesora
 - Dinamička preraspodela naredbi
 - Tomasulov algoritam
 - dinamička preraspodela može znatno da popravi performanse, pod uslovom da je predviđanje skokova dobro
 - nezgodno je što je Tomasulov algoritam kompleksan i zahteva značajne hardverske resurse
 - svaka rezervaciona stanica mora imati asocijativne bafere (radi poređenja sa informacijama na CDB) i kompleksnu kontrolnu logiku
 - samo jedna zajednička magistrala može biti usko grlo za veći broj RS
 - može se dodati još paralelnih magistrala, ali to onda dodatno povećava kompleksnost
 - u poslednje vreme su ovi problemi mahom rešeni, jer i mobilni uređaji imaju procesore sa dinamičkim preraspoređivanjem
 - algoritam je ostao mahom neiskorišten posle IBM 360/91, ali je ponovo stekao popularnost od 90-tih i danas je sastavni deo procesora