

Protočno izvršenje instrukcija

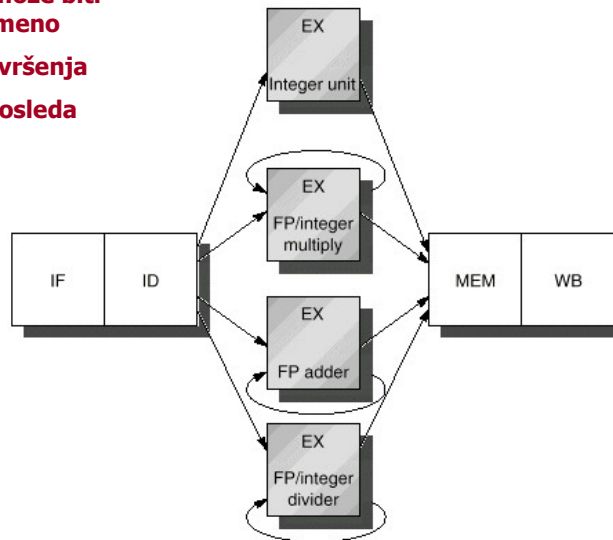
FP operacije

FP operacije

- * Kompletna analiza hazarda sprovedena je pod pretpostavkom da EXE faza traje 1 clk ciklus. (To je tačno za integer ALU)
- * FP operacije imaju različito vreme izvršenja (npr. sabiranje 4, množenje 7, deljenje 25)
- * Izvršenje FP operacija u 1 clk ciklusu je nepraktično iz nekoliko razloga
 - zahteva mnogo duži clk ciklus i/ili
 - složenu logiku
- * Zbog toga se dozvoljava da FP operacije imaju veću latentnost.
- * Izvršenje FP operacija ima iste faze u protočnom sistemu kao i integer instrukcije, sa sledećim razlikama:
 - EX ciklus se može ponavljati više puta (koliko je potrebno)
 - Može postojati više odvojenih funkcionalnih jedinica
 - Do zaustavljanja instrukcije može doći zbog strukturnih hazarda ili hazarda po podacima

Protočni sistem sa FP funkcionalnim jedinicama

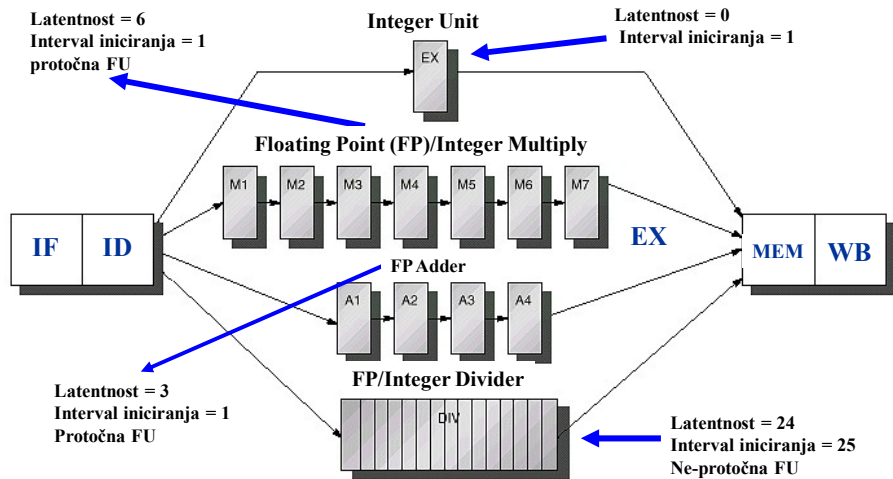
- više instrukcija može biti u EX fazi jednovremeno
- različito vreme izvršenja
- izvršenje van redosleda pribavljanja



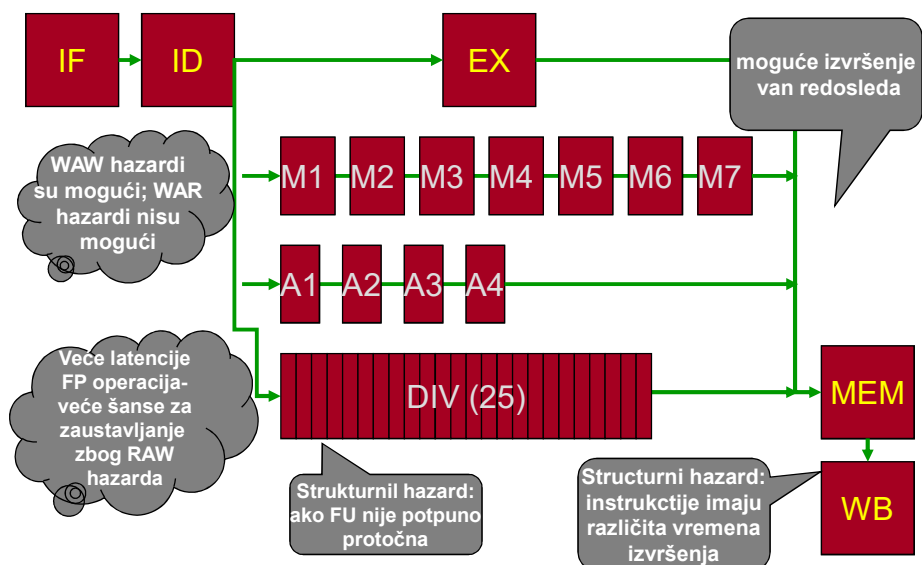
Latentnost i interval iniciranja

- * *Latentnost funkcionalne jedinice* se definiše kao broj klok ciklusa koji mora da protekne izmedju instrukcije koja generiše rezultat i instrukcije koja koristi rezultat (obično je jednaka kašnjenju sa upotrebom pribavljanje u napred)
- * *Interval inicijacije ili ponavljanja* funkcionalne jedinice jednak je broju klok ciklusa koji mora da protekne izmedju izdavanja dve instrukcije istog tipa

Latentnost i vreme iniciranja



FP Operacije – posledice



Primer – izvršenje instrukcija van redosleda

- * Usvojimo petostepeni protočni sistem
- * Treći stepen (EX) ima dve funkcionalne jedinice E1 i E2
 - Instrukcija prolazi ili kroz E1 ili kroz E2
 - E1 i E2 nisu protočno implementirane
 - Kašnjenje kroz stepen of E1 = 2 clk
 - Kašnjenje kroz stepen E2 = 4 clk
- * Kašnjenje ostalih stepena = 1 clk
- * Razmotrimo izvršenje niza od pet instrukcija
 - Instrukcije 1, 3, 5 koriste E1
 - Instrukcije 2, 4 koriste E2

Prostorno-vremenski dijagram izvršenja FP operacija

Delay		1	2	3	4	5	6	7	8	9	10	11	12	13
1	IF	1	2	3	4	5	5	5						
1	ID		1	2	3	4	4	4	5					
2	E1			1	1	3	3			5	5			
4	E2				2	2	2	2	4	4	4	4		
1	MEM					1		3	2			5	4	
1	WB						1		3	2			5	4

- * Izvršenje van redosleda
 - 3 se završava pre 2, i 5 pre 4
- * Instrukcije mogu biti zaustavljene zbog strukturnih hazarda
 - Instrukcije 2 i 4 zahtevaju E2 jedinicu u isto vreme
 - Instrukcija 4 se zaustavlja u ID stepenu
 - Ovo ima za posledicu zaustavljanje instrukcije 5 u IF stepenu

Da li su WAW hazardi zaista mogući?

Primer

DIVD	F0, F2, F4
SUBD	F0, F8, F10

- Izmedju ove dve instrukcije postoji WAW hazard.
- instrukcija SUBD izvršava za kraće vreme od DIVD i upisaće rezultat u F0 pre DIVD i kreirati WAW hazard
- Ovaj hazard nastupa zato što se rezultat DIVD nigde ne koristi.
- Ako bi postojala instrukcija izmedju DIVD i SUBD koja koristi F0, sistem bi se zaustavio zbog RAW zavisnosti i SUBD ne bi otpočela sa izvršenjem dok DIVD ne obavi upis u F0.

• **Zaključak:** WAW hazard nastupa samo ako se izvršava neka instrukcija čiji se rezultat nigde ne koristi (u ovom primeru DIVD)

Da li su WAW hazardi zaista mogući?

- **Ipak, postoji mogućnost!**

Primer:

- Neka kompajler koristi tehniku zakašnjenog grananja
- neka se preuredjenje koda obavlja pod pretpostavkom da se grananje neće obaviti
- Neka se grananje obavi

	BNEZ	R1, ime	
	DIVD	F0, F2, F4	← instrukcija postavljena u delay slot
	.		
	.		
ime:	SUBD	F0, F8, F4	

➤ Kreira se WAW hazard !

Šta preduzeti?

1. Zakasniti izdavanje SUBD dok DIVD ne udje u MEM fazu (jer se svi hazardi mogu detektovati u ID)
2. Kada se detektuje hazard instrukciji DIVD (beskorisnoj) zabraniti upis u RF, a instrukcija SUBD može da se izda bez zakašnjenja

Šta je sa prekidima?

DIVD F0, F2, F4

ADDD F10, F10, F8

SUBD F12, F12, F14

* Instrukcije ADDD i SUBD se mogu izvršiti pre instrukcije DIVD koja je ranije počela sa izvršenjem.

* Zašto je ova situacija opasni i ako nema hazarda?

- Šta se dešava ako u toku izvršenja DIVD nastupi prekid i to kada je ADDD već obavila upis u F10?

- Nakon obrade prekida počinje se sa izvršenjem od instrukcije DIVD, što znači da će se i ADDD ponovo izvršiti
- Doći će do greške jer je jedan operand ADDD izmenjen upisom (F10)!

Kako se rešava problem?

* Nekoliko načina je predloženo:

1. **Baferovanje rezultata operacija dok se ne izvrše sve operacije koje su ranije počele sa izvršenjem (rešenje je moguće ali postaje dosta skupo ako su razlike u vremenu izvršenja FP operacija velika. Potreban je veliki baferski prostor za pamćenje rezultata. Zahteva se i dodatni hw za premošćavanje za rezultate koji se nalaze u redu čekanja a nisu upisani u RF. PowerPC i MIPS R10000 koriste ovo rešenje)**
2. **Čuva se trag originalnih vrednosti odredišnih registara svih instrukcija koje se nalaze u protočnom sistemu, tako da u slučaju prekida ima dovoljno informacija za restartovanje svih instrukcija iza one koja je izazvala prekid (History file - CYBER180/190, VAX)**
3. **A Future file- pamti novije vrednosti registara; kada se ranije izdata instrukcija okonča registarski fajl se ažurira iz future fajla. U slučaju da nastupi prekid, glavni registarski fajl ima originalne vrednosti operanada.**