RIO RA
2018
asistentica Hana Haseljić

Vježba 1
Protočna struktura bez prosljeđivanja

U jednociklusnoj mikroarhitekturi čitava instrukcija se izvršava u jednom ciklusu. Ovo znači da je trajanje jednog ciklusa sata određeno trajanjem najsporije instrukcije.
Današnji procesori koriste protočnost. U protočnoj mikroarhitekturi se izvršava više instrukcija simultano čime se poboljšava propusnost.

Zadatak 1
Odredite broj ciklusa potrebnih za izvršenje sekvence instrukcija u protočnoj strukturi MIPS oglednog procesora. Prikazati u kojoj se fazi nalazi svaka od instrukcija u određenom ciklusu.

```
add  r1,  r2,  r3
sub  r4,  r5,  r6
and  r7,  r8,  r9
 or  r9,  r10, r2
```

Put podataka u protočnoj strukturi je razdvojen na pet dijelova, odnosno pet faza izvršenja instrukcije:
1. IF – instruction fatch (pribavljanje instrukcija iz memorije instrukcija IM)
2. ID – instruction decode (dekodiranje i čitanje iz registara Reg)
3. EX – execution (izvršenje instrukcije u aritmetičko-logičkoj jedinici ALU ili računanje adrese)
4. MEM – memory (pristup podatkovnoj memoriji DM)
5. WB – write back (pisanje u registar, iz istog skupa registara iz kojeg se i čita Reg).

U oglednoj protočnoj strukturi dvije instrukcije ne mogu istovremeno biti u istoj fazi. U jednom ciklusu je moguće uraditi čitanje i pisanje u registre jer se čitanje radi u prvoj polovini ciklusa, a faza pisanja u drugoj polovini ciklusa.

Da se radi o jednociklusnom procesoru izvršenje prethodnih instrukcija bi trajalo 5 * 4 ciklusa.

	1	2	3	4	5	6	7	8	9	10	11
add r1, r2, r3	IM	Reg	ALU	DM	Reg						
sub r4, r5, r6						IM	Reg	ALU	DM	Reg	
and r7, r8, r9											IM
or r9, r10, r2											

Broj ciklusa potrebnih za izvršenje sekvence instrukcija je 8.

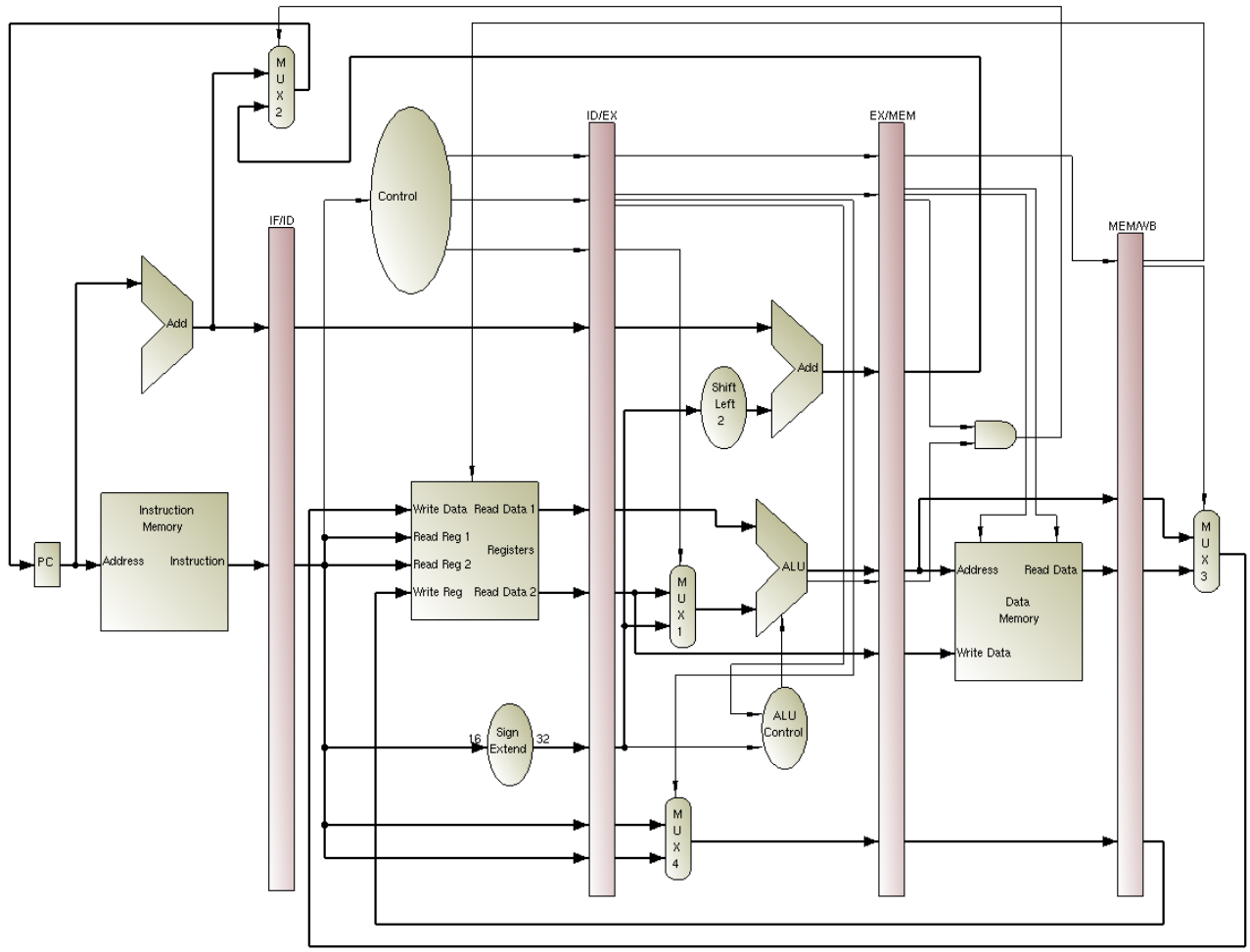
	1	2	3	4	5	6	7	8
add r1, r2, r3	IF	ID	EX	MEM	WB			
sub r4, r5, r6		IF	ID	EX	MEM	WB		
and r7, r8, r9			IF	ID	EX	MEM	WB	
or r9, r10, r2				IF	ID	EX	MEM	WB

				1	2	3	4	5	6	7	8
add	r1,	r2,	r3	IM	Reg	ALU	DM	Reg			
sub	r4,	r5,	r6		IM	Reg	ALU	DM	Reg		
and	r7,	r8,	r9			IM	Reg	ALU	DM	Reg	
or	r9,	r10,	r2				IM	Reg	ALU	DM	Reg

Zadatak 2

Razumjeti komponente puta podataka MIPS protočne strukture koristeći MIPS Datapath na primjeru sekvence instrukcija.

```
lw  r1,   1  (r0)
lw  r2,   5  (r0)
lw  r3,   3  (r0)
nop
add r3,  r1,  r2
```



Eksportovati MIPS Datapath, preimenovati MIPS-Datapath.exe2 u MIPS-Datapath.exe. Iz menija Layout odabrati Pipelined. Po defaultu je postavljeno Simple što predstavlja jednociklusni procesor. Pored putapodataka ovdje je prikazan i kontrolni dio.

Protočna struktura ima pregradne registra, IF/ID, ID/EX, EX/MEM i MEM/WB.

	1	2	3	4	5	6	7	8	9
lw r1, 1 (r0)	IF	ID	EX	MEM	WB				
lw r2, 5 (r0)		IF	ID	EX	MEM	WB			
lw r3, 3 (r0)			IF	ID	EX	MEM	WB		
nop				-	-	-	-	-	
add r3, r1, r2					IF	ID	EX	MEM	WB

Registar PC sadrži adresu instrukcija koja se treba izvršiti. Instrukcija se čita iz instrukcijske memorije. PC je povezan sa instrukcijom memorijom, ulaz *Address*. U 1. ciklusu se pribavlja instrukcija LW.
U 2. ciklusu se pribavlja druga instrukcija. Prva instrukcija se nalazi u fazi dekodiranja. Čita se registar koji sadrži baznu adresu, R0. Odgovarajući biti adrese su povezani sa *Read port 1 RegisterFile*-a. Vrijednost ovog registra se čita u *ReadData1*. Ova instrukcija traži i ofset jer 16-bitna vrijednost može biti ili pozitivna ili negativna pa ju se treba proširiti na 32 bita, *SignExtend*.
U 3. ciklusu se sabiraju bazna adresa i ofset da bi se podatak našao u memoriji. *ALU* prima dva operanda, prvi je bazna adresa, pročitana iz RegFile, a drugi ofset. *ALUControl* određuje operaciju. ALU daje rezultat, što je adresa za lw instrukciju i Zero flag. Rezultat iz ALU se šalje memoriji podataka. U ovom ciklusu druga instrukcija se dekodira, a treća pribavlja.
U 4. ciklusu se podatak čita iz *Data Memory* na *ReadData* sabirnicu. U 5. ciklusu se podatak iz upisuje u destinacijski registar na kraju ciklusa na *WriteData*.
Za petu instrukciju se iz Register File čitaju vrijednosti iz dva registra, *Read Data 1* i *Read Data2*. Kako drugi operand koji prima ALU može biti i registar ili sign extended vrijednost ofseta stavlja se multiplekser MUX 4. Rezultat iz ALU za ovu instrukciju nije adresa što znači da se i ne čita iz memorije pa se rezultat prosljeđuje na MUX 3 kako bi se odlučilo da li se u registar upisuje pročitano iz memorije ili rezultat iz ALU.

Zadatak 3

Odredite broj ciklusa potrebnih za izvršenje sekvence instrukcija u protočnoj strukturi MIPS oglednog procesora. Prikazati u kojoj se fazi nalazi svaka od instrukcija u određenom ciklusu.

```
lw   r1,   16 (r0)
sub  r4,   r5, r6
add  r4,   r8, r9
sw   r5,   16 (r5)
```

Zadatak 4

U sekvencu instrukcija iz zadatka 3 učitati u svaki registar koji se koristi vrijednost i pokrenuti u MIPS Datapath. Mijenjati sekvencu je moguće unutar simulatora odabirom taba Editor.