Computer Organization Assignment-4

Name: Aayush Patel

Roll: EE21B003

PART – A

Modulo Operation

Computing : a0 % a1

Program (for **Flushing Enabled**) :

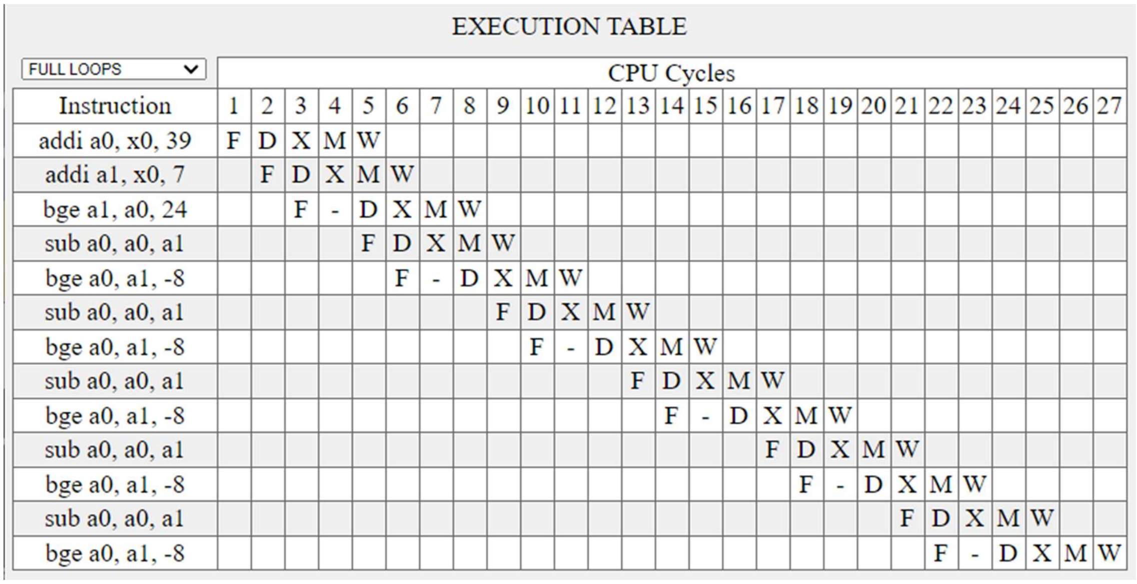
```
addi a0, x0, 39
addi a1, x0, 7


bge a1, a0, skip


loop:
        sub a0, a0, a1
        bge a0, a1, loop
skip:
```

Data Hazard: Read After Write (RAW)

The 3rd instruction reads value of a1, after writing in 2nd instruction. So the pipeline needs to stall.

Branch Hazard : For both the branching instructions. Resolved by flushing.

Forwarding Enabled (with Flushing)



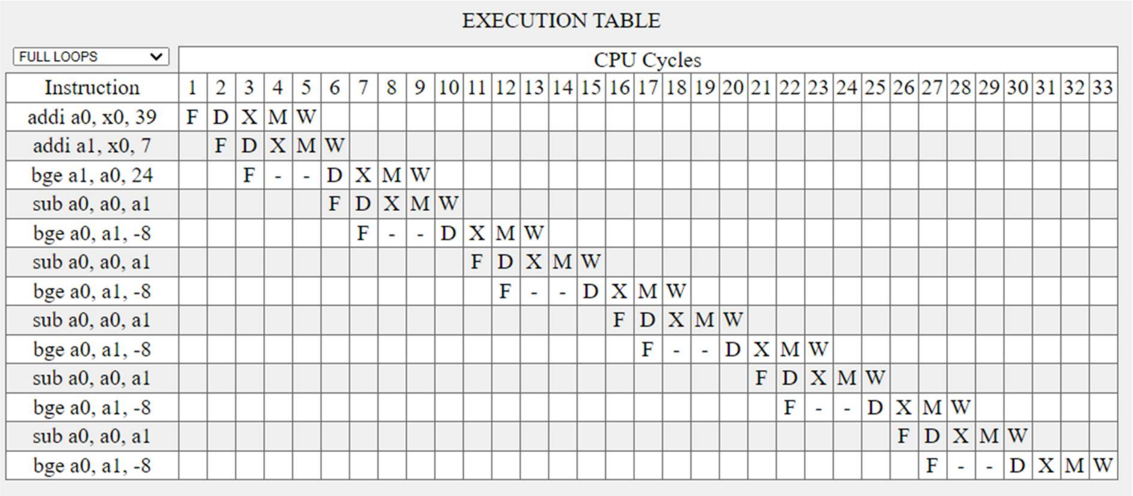
EXECUTION TABLE

	CPU Cycles																										
Instruction	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
addi a0, x0, 39	F	D	X	M	W																						
addi a1, x0, 7		F	D	X	M	W																					
bge a1, a0, 24			F	-	D	X	M	W																			
sub a0, a0, a1					F	D	X	M	W																		
bge a0, a1, -8						F	-	D	X	M	W																
sub a0, a0, a1									F	D	X	M	W														
bge a0, a1, -8										F	-	D	X	M	W												
sub a0, a0, a1													F	D	X	M	W										
bge a0, a1, -8														F	-	D	X	M	W								
sub a0, a0, a1																	F	D	X	M	W						
bge a0, a1, -8																		F	-	D	X	M	W				
sub a0, a0, a1																					F	D	X	M	W		
bge a0, a1, -8																						F	-	D	X	M	W

Throughput = 13/27 = 0.48148

(27 Clock Cycles)

Forwarding Disabled (with Flushing)



EXECUTION TABLE

	CPU Cycles																																
Instruction	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33
addi a0, x0, 39	F	D	X	M	W																												
addi a1, x0, 7		F	D	X	M	W																											
bge a1, a0, 24			F	-	-	D	X	M	W																								
sub a0, a0, a1						F	D	X	M	W																							
bge a0, a1, -8							F	-	-	D	X	M	W																				
sub a0, a0, a1											F	D	X	M	W																		
bge a0, a1, -8												F	-	-	D	X	M	W															
sub a0, a0, a1																F	D	X	M	W													
bge a0, a1, -8																	F	-	-	D	X	M	W										
sub a0, a0, a1																					F	D	X	M	W								
bge a0, a1, -8																						F	-	-	D	X	M	W					
sub a0, a0, a1																										F	D	X	M	W			
bge a0, a1, -8																											F	-	-	D	X	M	W

Throughput = 13/33 = 0.393939

(33 Clock Cycles)

Program (for **Delay Slot**):

```
addi a0, x0, 39
addi a1, x0, 7

bge a1, a0, skip
nop

loop:
        sub a0, a0, a1
        bge a0, a1, loop
skip:
```

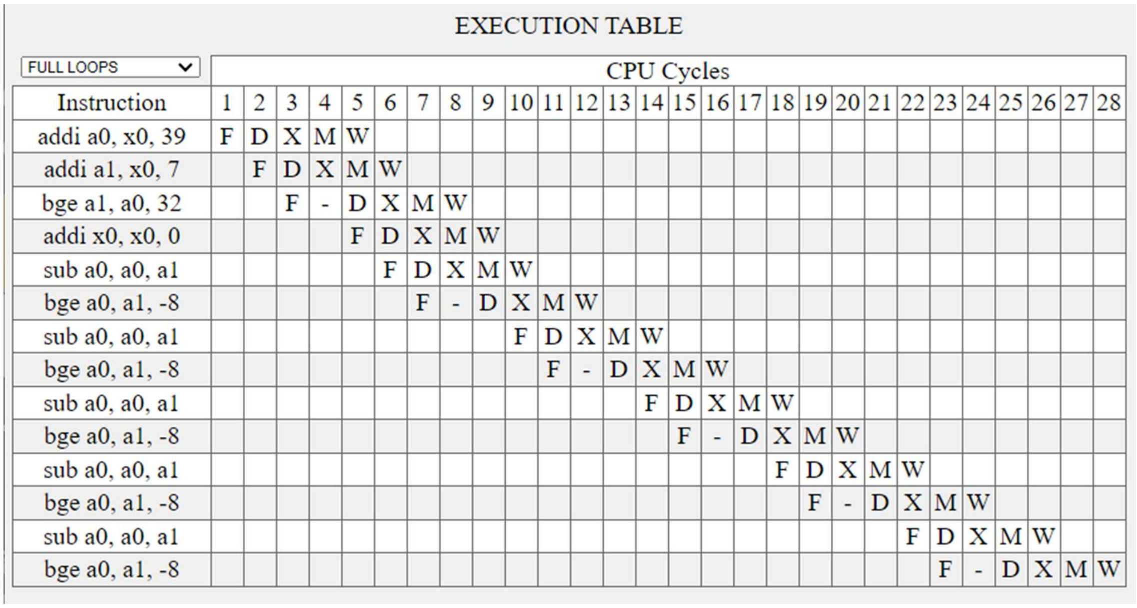
Data Hazard: Read After Write (RAW)

The 3rd instruction reads value of a1, after writing in 2nd instruction. So the pipeline needs to stall.

Branch Hazard : For both the branching instructions, but the second one does not have any instruction after it so it won't initiate it. Resolved by using no operation (nop).

Note: We might get wrong/ unexpected results in case of Delay Slot Branch Hazard Handling as it does not flush the previous instructions which were not meant to take place. So I used nop here to avoid getting wrong results.

Forwarding Enabled (with Delay Slot)



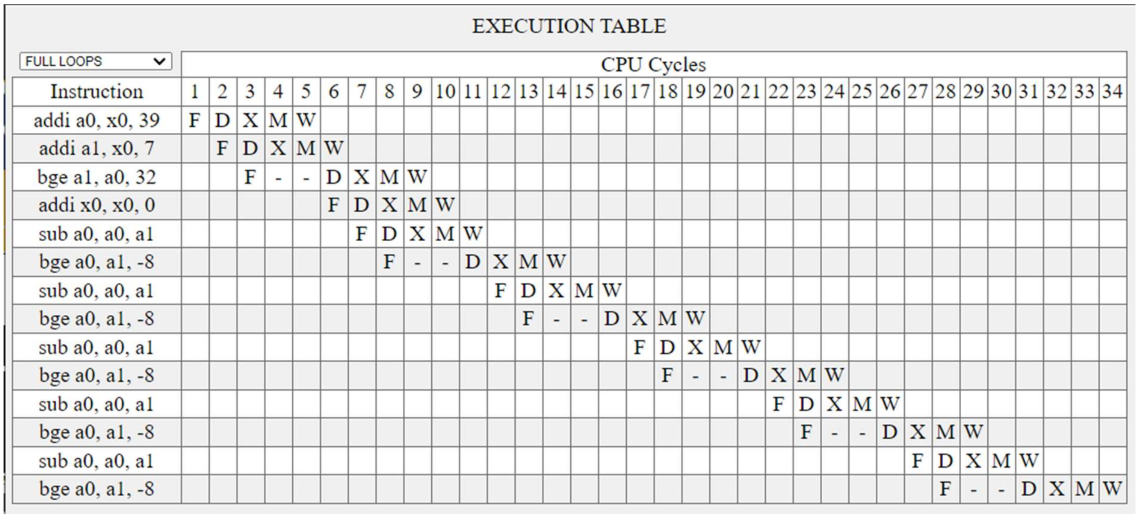
EXECUTION TABLE

FULL LOOPS	CPU Cycles																											
Instruction	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
addi a0, x0, 39	F	D	X	M	W																							
addi a1, x0, 7		F	D	X	M	W																						
bge a1, a0, 32			F	-	D	X	M	W																				
addi x0, x0, 0					F	D	X	M	W																			
sub a0, a0, a1						F	D	X	M	W																		
bge a0, a1, -8							F	-	D	X	M	W																
sub a0, a0, a1										F	D	X	M	W														
bge a0, a1, -8											F	-	D	X	M	W												
sub a0, a0, a1														F	D	X	M	W										
bge a0, a1, -8															F	-	D	X	M	W								
sub a0, a0, a1																		F	D	X	M	W						
bge a0, a1, -8																			F	-	D	X	M	W				
sub a0, a0, a1																						F	D	X	M	W		
bge a0, a1, -8																							F	-	D	X	M	W

Throughput = 13/28 = 0.464285

(28 Clock Cycles)

Forwarding Disabled (with Delay Slot)



EXECUTION TABLE

FULL LOOPS	CPU Cycles																																	
Instruction	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34
addi a0, x0, 39	F	D	X	M	W																													
addi a1, x0, 7		F	D	X	M	W																												
bge a1, a0, 32			F	-	-	D	X	M	W																									
addi x0, x0, 0						F	D	X	M	W																								
sub a0, a0, a1							F	D	X	M	W																							
bge a0, a1, -8								F	-	-	D	X	M	W																				
sub a0, a0, a1												F	D	X	M	W																		
bge a0, a1, -8													F	-	-	D	X	M	W															
sub a0, a0, a1																	F	D	X	M	W													
bge a0, a1, -8																		F	-	-	D	X	M	W										
sub a0, a0, a1																						F	D	X	M	W								
bge a0, a1, -8																							F	-	-	D	X	M	W					
sub a0, a0, a1																											F	D	X	M	W			
bge a0, a1, -8																												F	-	-	D	X	M	W

Throughput = 13/34 = 0.382352

(34 Clock Cycles)

PART – B

After running the given file (helloworld.riscv) on the RISC-V Simulator I got the following Results.

Performance Table:

Performance Parameter	**AT**	**NT**	**BTFNT**	**BPB**
No. of Instructions	152	140	152	152
No. of Cycles	262	228	250	262
Average Cycle/Instruction	1.7237	1.6286	1.6447	1.7237
Accuracy	0.4706	0.5833	0.7059	0.4706
No. of Control Hazard	26	23	22	26
No. of Data Hazard	76	73	77	76
No. of Memory Hazard	1	1	1	1