Microprocessor and Computer Architecture Laboratory

UE19CS256

4th Semester, Academic Year 2020-21

Date: 11/04/2021

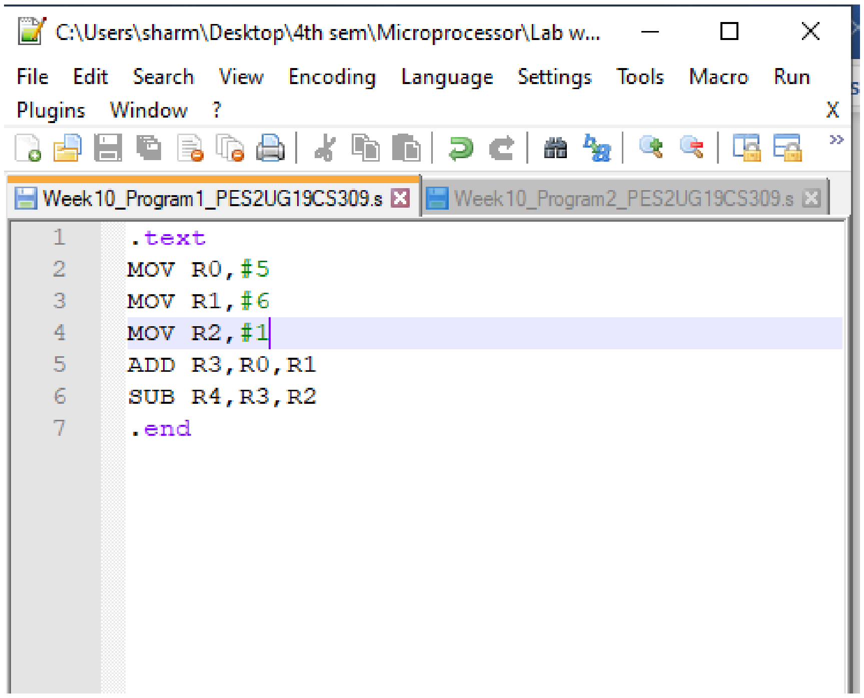
Name: R SHARMILA	SRN: PES2UG19CS309	Section E

Week#____10______ Program Number: ____1__

Given a C- Code convert it in its equivalent ARM Code.
These programs need to be executed on ARMSIM Simulator

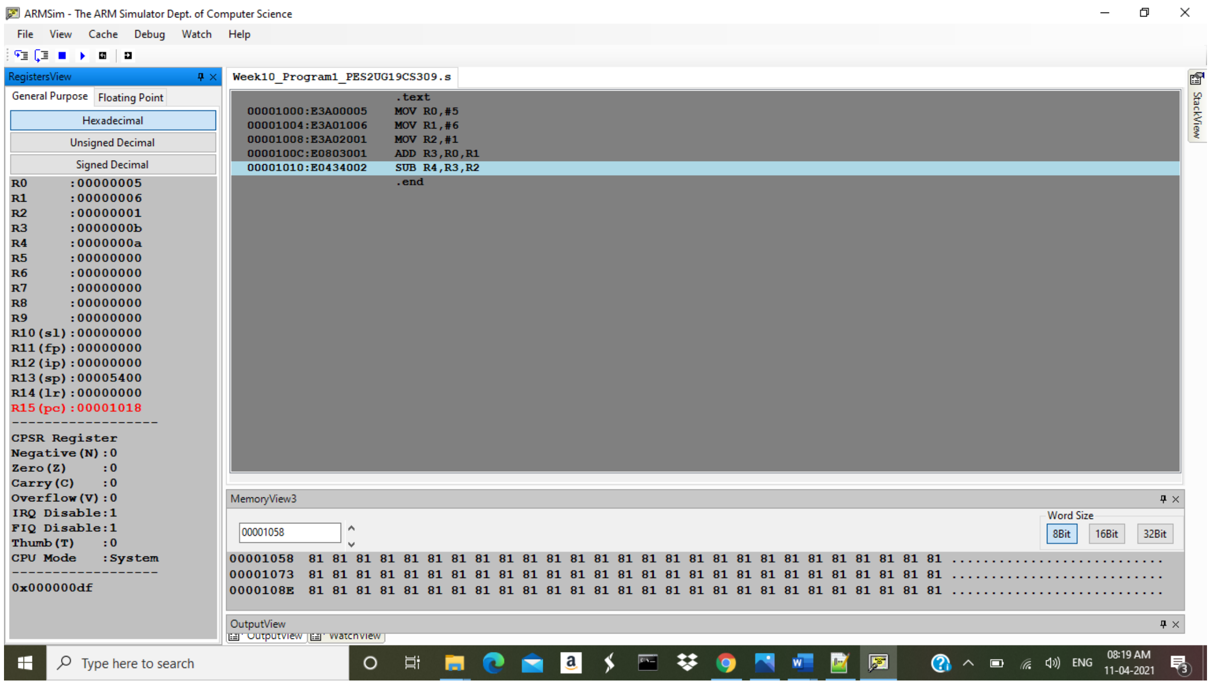
1) x = (a + b) - c;

ARM Assembly Language Code


C:\Users\sharm\Desktop\4th sem\Microprocessor\Lab w...
File Edit Search View Encoding Language Settings Tools Macro Run
Plugins Window ?
Week10_Program1_PES2UG19CS309.s
Week10_Program2_PES2UG19CS309.s

```
.text
MOV R0,#5
MOV R1,#6
MOV R2,#1
ADD R3,R0,R1
SUB R4,R3,R2
.end
```

Screenshot showing the value of x, a, b, c in the register window.



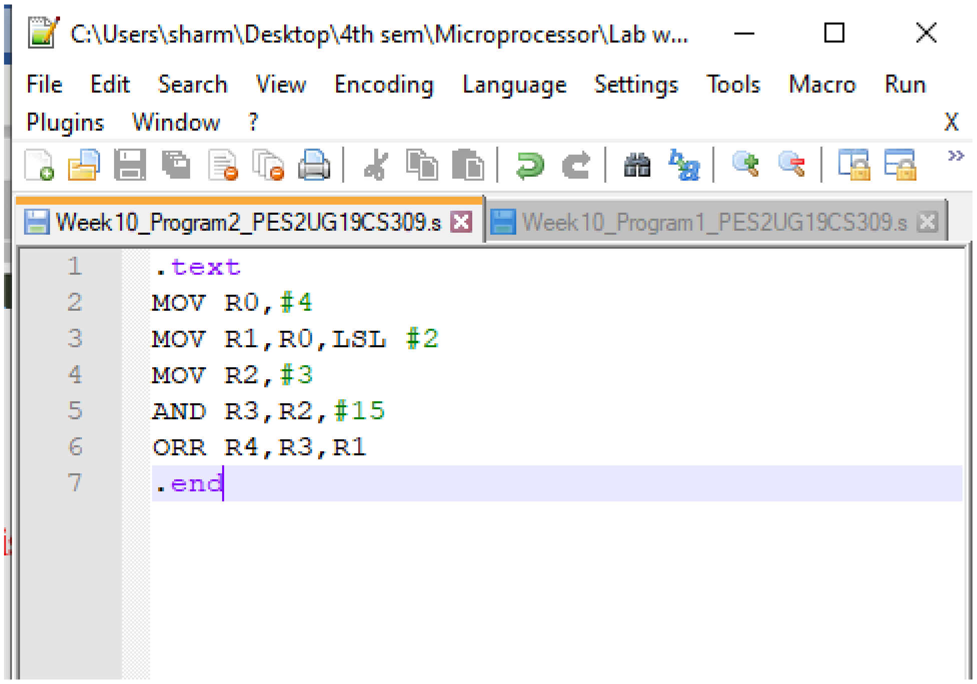
```
Week10_Program1_PES2UG19CS309.s

                        .text
00001000:E3A00005       MOV R0,#5
00001004:E3A01006       MOV R1,#6
00001008:E3A02001       MOV R2,#1
0000100C:E0803001       ADD R3,R0,R1
00001010:E0434002       SUB R4,R3,R2
                        .end
```

```
R0       :00000005
R1       :00000006
R2       :00000001
R3       :0000000b
R4       :0000000a
R5       :00000000
R6       :00000000
R7       :00000000
R8       :00000000
R9       :00000000
R10(sl)  :00000000
R11(fp)  :00000000
R12(ip)  :00000000
R13(sp)  :00005400
R14(lr)  :00000000
R15(pc)  :00001018
------------------
CPSR Register
Negative(N):0
Zero(Z)    :0
Carry(C)   :0
Overflow(V):0
IRQ Disable:1
FIQ Disable:1
Thumb(T)   :0
CPU Mode   :System
------------------
0x000000df
```

```
00001058  81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81  ...........................
00001073  81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81  ...........................
0000108E  81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81  ...........................
```

2) z = (a << 2) | (b & 15);

ARM Assembly Language Code

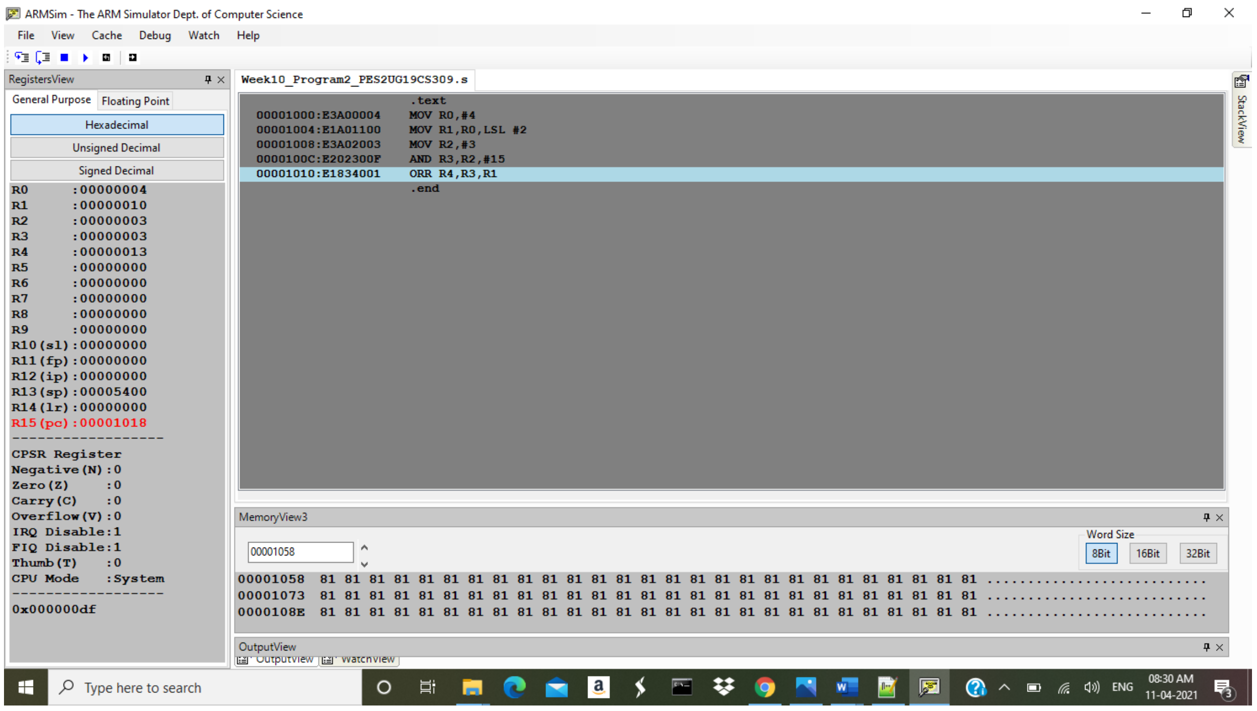


C:\Users\sharm\Desktop\4th sem\Microprocessor\Lab w...

Week10_Program2_PES2UG19CS309.s

```
.text
MOV R0,#4
MOV R1,R0,LSL #2
MOV R2,#3
AND R3,R2,#15
ORR R4,R3,R1
.end
```

Screenshot showing the value of a, b, z in the register window.



```
R0      :00000004
R1      :00000010
R2      :00000003
R3      :00000003
R4      :00000013
R5      :00000000
R6      :00000000
R7      :00000000
R8      :00000000
R9      :00000000
R10(sl) :00000000
R11(fp) :00000000
R12(ip) :00000000
R13(sp) :00005400
R14(lr) :00000000
R15(pc) :00001018
------------------
CPSR Register
Negative(N):0
Zero(Z)    :0
Carry(C)   :0
Overflow(V):0
IRQ Disable:1
FIQ Disable:1
Thumb(T)   :0
CPU Mode   :System
------------------
0x000000df
```

```
                      .text
00001000:E3A00004     MOV R0,#4
00001004:E1A01100     MOV R1,R0,LSL #2
00001008:E3A02003     MOV R2,#3
0000100C:E202300F     AND R3,R2,#15
00001010:E1834001     ORR R4,R3,R1
                      .end
```

```
00001058
00001058  81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81  ...........................
00001073  81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81  ...........................
0000108E  81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81  ...........................
```

Microprocessor and Computer Architecture Laboratory

UE19CS256

4th Semester, Academic Year 2020-21

Date: 11/04/2021

Name: R SHARMILA	SRN: PES2UG19CS309	Section E

Week#____10_______ Program Number: ____2__

1) Consider the following instructions. Execute these instructions using simulator of 5 stage pipeline of MIPS architecture.

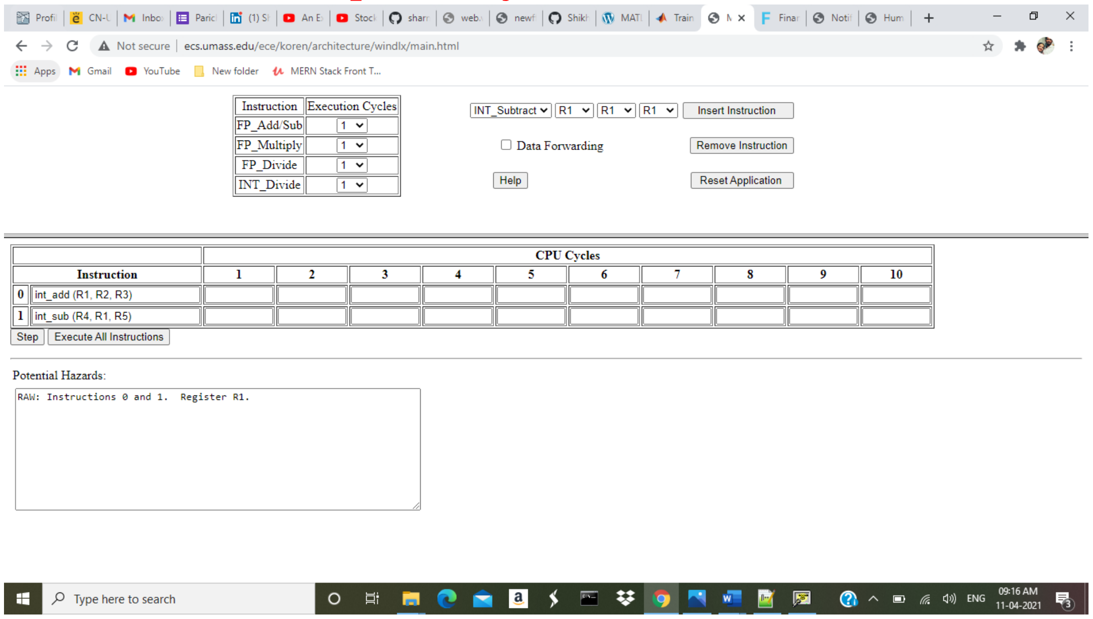
ADD R0, R1, R2

SUB R3, R0, R4.

Observe the following and note down the results.

a) Check whether there is data dependency for the second instruction?

Yes there is raw dependency



ecs.umass.edu/ece/koren/architecture/windlx/main.html

Instruction	Execution Cycles
FP_Add/Sub	1
FP_Multiply	1
FP_Divide	1
INT_Divide	1

INT_Subtract R1 R1 R1 — Insert Instruction

☐ Data Forwarding

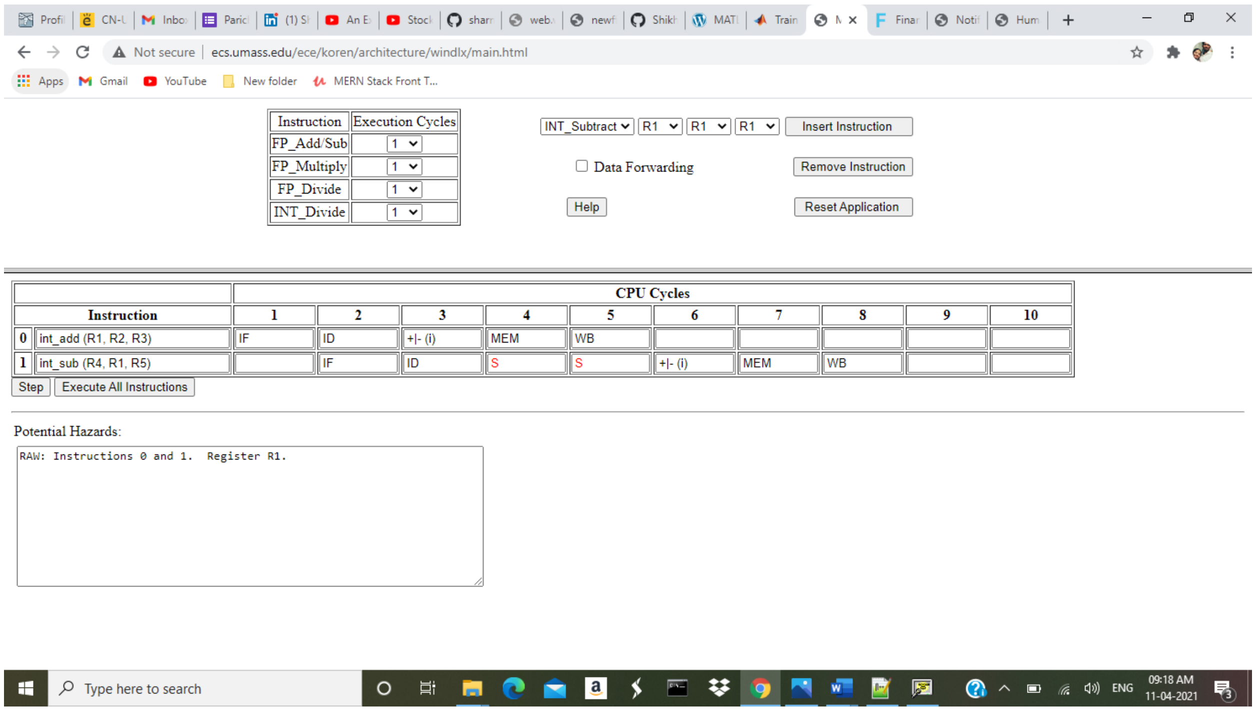
	Instruction	1	2	3	4	5	6	7	8	9	10
0	int_add (R1, R2, R3)										
1	int_sub (R4, R1, R5)										

Potential Hazards:

```
RAW: Instructions 0 and 1.  Register R1.
```

a) If yes, then, how many stall states have been introduced?

2 stalls have been introduced



Instruction	Execution Cycles
FP_Add/Sub	1
FP_Multiply	1
FP_Divide	1
INT_Divide	1

INT_Subtract R1 R1 R1 | Insert Instruction

☐ Data Forwarding | Remove Instruction

Help | Reset Application

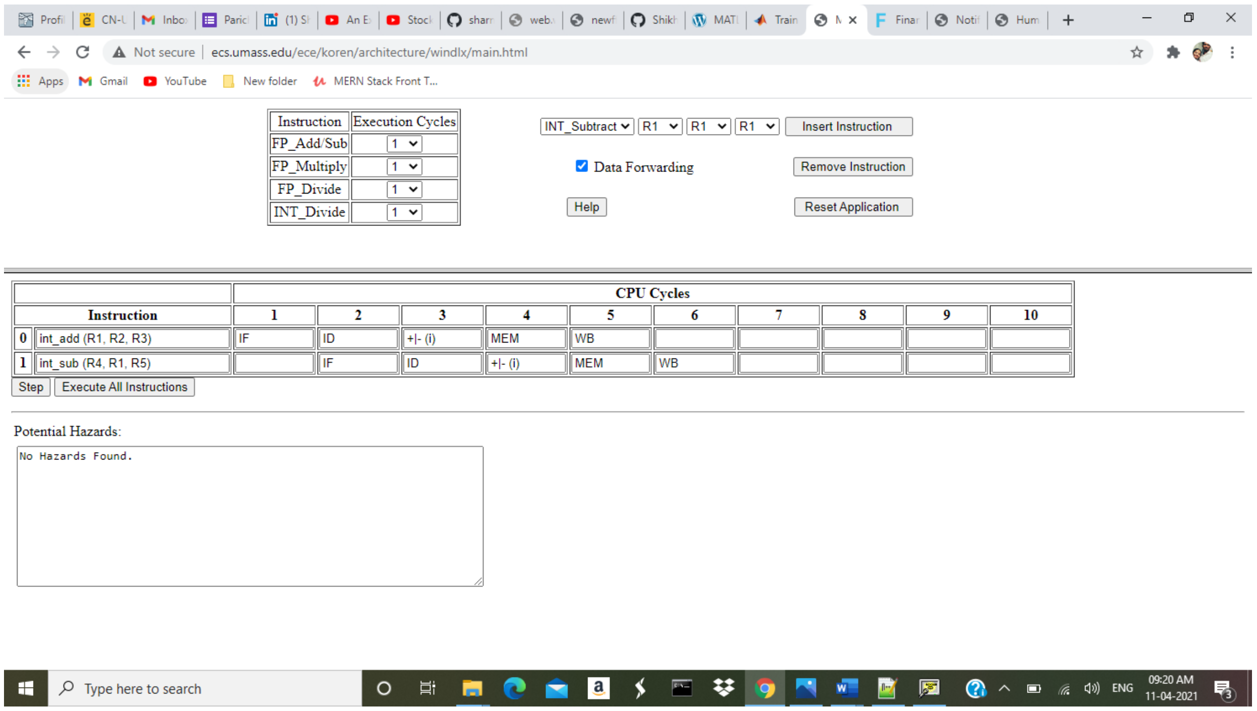
	Instruction	1	2	3	4	5	6	7	8	9	10
0	int_add (R1, R2, R3)	IF	ID	+|- (i)	MEM	WB					
1	int_sub (R4, R1, R5)		IF	ID	S	S	+|- (i)	MEM	WB		

Potential Hazards:

```
RAW: Instructions 0 and 1.  Register R1.
```

b) If data forwarding is applied how many stall states have been reduced?

No stalls when data forwarding is applied.



Instruction	Execution Cycles
FP_Add/Sub	1
FP_Multiply	1
FP_Divide	1
INT_Divide	1

INT_Subtract R1 R1 R1 | Insert Instruction

☑ Data Forwarding | Remove Instruction

Help | Reset Application

	Instruction	1	2	3	4	5	6	7	8	9	10
0	int_add (R1, R2, R3)	IF	ID	+|- (i)	MEM	WB					
1	int_sub (R4, R1, R5)		IF	ID	+|- (i)	MEM	WB				

Potential Hazards:

```
No Hazards Found.
```

Microprocessor and Computer Architecture Laboratory

UE19CS256

4th Semester, Academic Year 2020-21

Date: 11/04/2021

Name: R SHARMILA	SRN: PES2UG19CS309	Section E

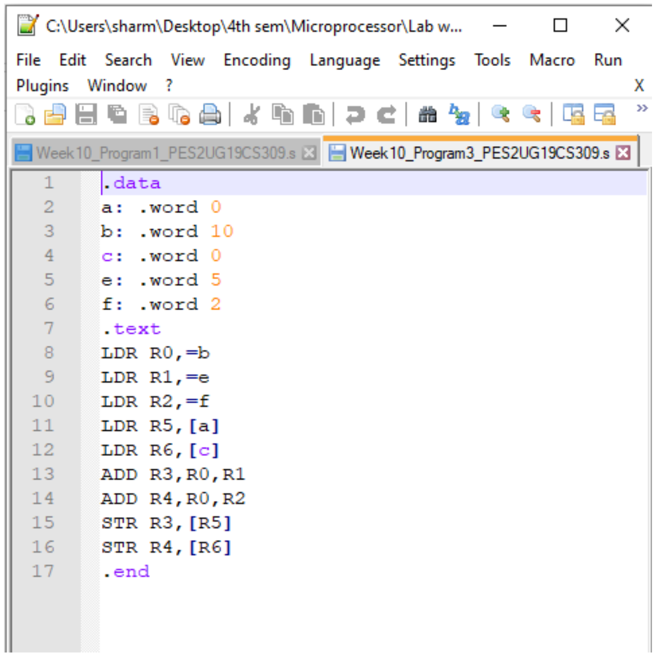
Week#____10_______ Program Number: ____3__

Consider the following code segment in C.

A = B + E;
C = B + F;

a) Write the code using MIPS 5 STAGE pipeline architecture.

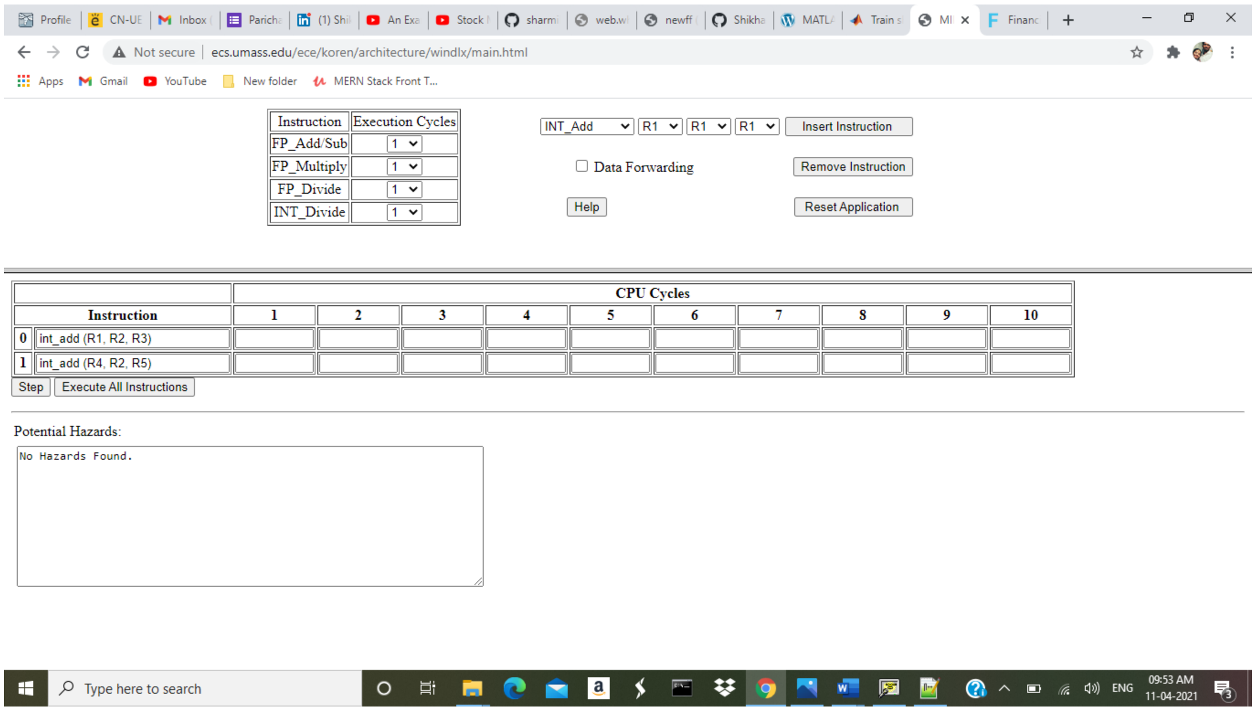
ARM Assembly Language Code


C:\Users\sharm\Desktop\4th sem\Microprocessor\Lab w...
File Edit Search View Encoding Language Settings Tools Macro Run
Plugins Window ?
Week10_Program1_PES2UG19CS309.s
Week10_Program3_PES2UG19CS309.s

```
.data
a:  .word 0
b:  .word 10
c:  .word 0
e:  .word 5
f:  .word 2
.text
LDR R0,=b
LDR R1,=e
LDR R2,=f
LDR R5,[a]
LDR R6,[c]
ADD R3,R0,R1
ADD R4,R0,R2
STR R3,[R5]
STR R4,[R6]
.end
```

b) Find the hazards;

No hazard



Not secure | ecs.umass.edu/ece/koren/architecture/windlx/main.html

Instruction	Execution Cycles
FP_Add/Sub	1
FP_Multiply	1
FP_Divide	1
INT_Divide	1

INT_Add R1 R1 R1 — Insert Instruction

☐ Data Forwarding — Remove Instruction

Help — Reset Application

		CPU Cycles									
	Instruction	1	2	3	4	5	6	7	8	9	10
0	int_add (R1, R2, R3)										
1	int_add (R4, R2, R5)										

Step — Execute All Instructions

Potential Hazards:

```
No Hazards Found.
```

c) Reorder the instructions to avoid pipeline stalls.

No reordering is required.

Microprocessor and Computer Architecture Laboratory

UE19CS256

4th Semester, Academic Year 2020-21

Date: 11/04/2021

Name: R SHARMILA	SRN: PES2UG19CS309	Section E

Week#____10_______ Program Number: ___4__

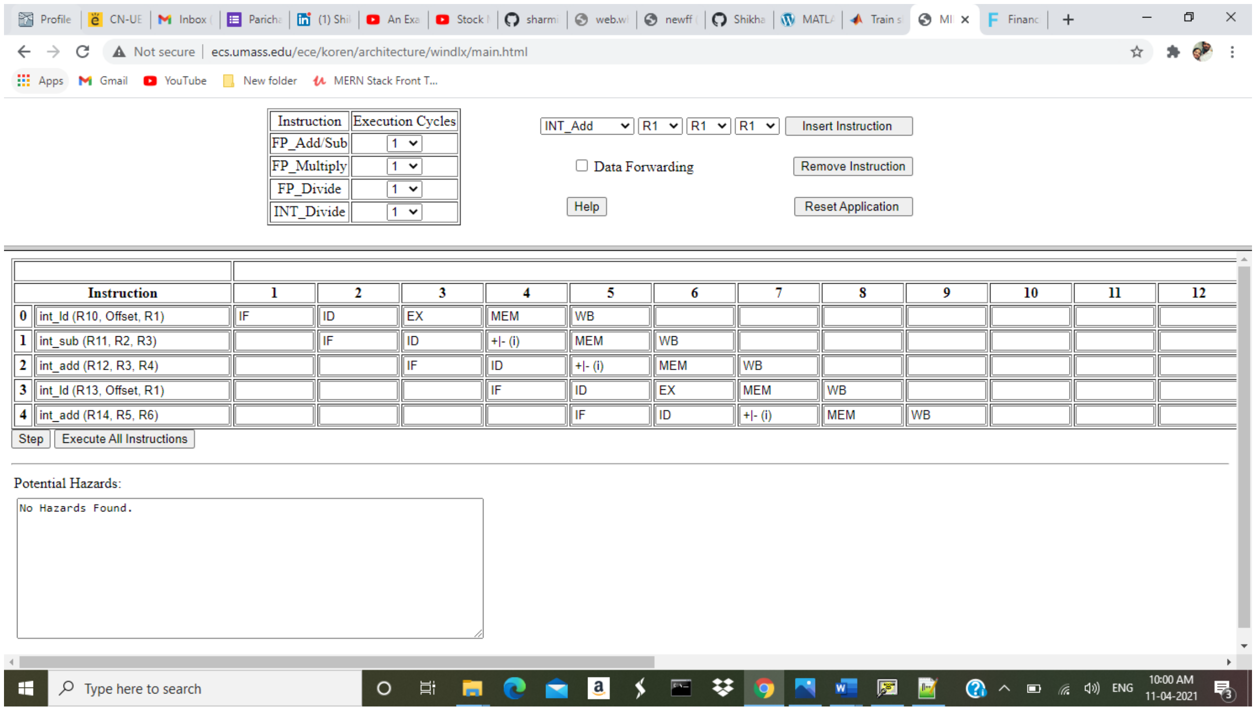
Using MIPS 5 stage pipeline architecture, execute the following instructions and avoid stall states if any.

```
LW    $10, 20($1)
SUB   $11, $2, $3
ADD   $12, $3, $4
LW    $13, 24($1)
ADD   $14, $5, $6
```

a) Related Screenshot with stalls

No stalls

b) Related Screenshot without stalls



Instruction	Execution Cycles
FP_Add/Sub	1
FP_Multiply	1
FP_Divide	1
INT_Divide	1

☐ Data Forwarding

	Instruction	1	2	3	4	5	6	7	8	9	10	11	12
0	int_ld (R10, Offset, R1)	IF	ID	EX	MEM	WB							
1	int_sub (R11, R2, R3)		IF	ID	+|- (i)	MEM	WB						
2	int_add (R12, R3, R4)			IF	ID	+|- (i)	MEM	WB					
3	int_ld (R13, Offset, R1)				IF	ID	EX	MEM	WB				
4	int_add (R14, R5, R6)					IF	ID	+|- (i)	MEM	WB			

Potential Hazards:

```
No Hazards Found.
```

Microprocessor and Computer Architecture Laboratory

UE19CS256

4th Semester, Academic Year 2020-21

Date: 11/04/2021

Name: R SHARMILA	SRN: PES2UG19CS309	Section E

Week#____10_______ Program Number: ___5__

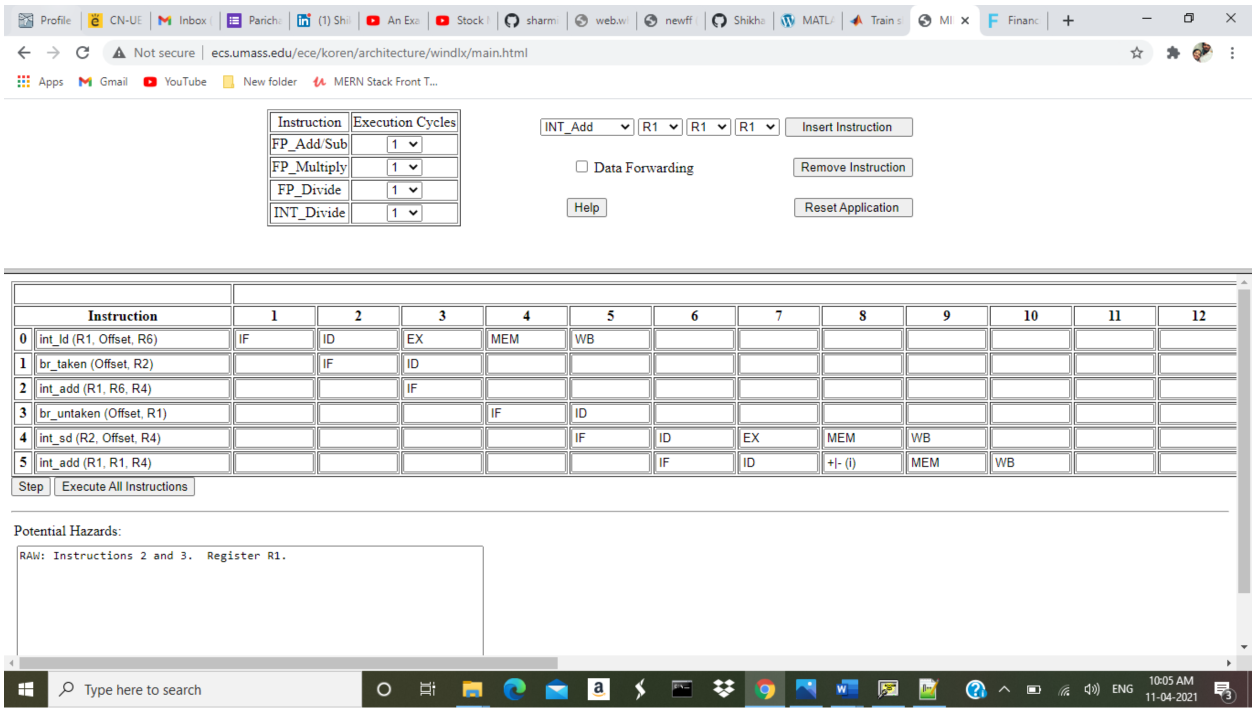
This exercise is to understand the relationship between delay slots, control hazards and branch execution in a 5 stage MIPS pipelined processor.

```
Label 1:   LW    $1, 40($6)
           BEQ   $2, $3, Label2    : branch taken
           ADD   $1, $6, $4
Label2:    BEQ   $1, $2, Label1    : branch not taken
            SW    $2, 20($4)
            ADD   $1, $1, $4
```

Assume full data forwarding and predict- taken branch prediction.

Note the observations.

Related Screenshot



Instruction	Execution Cycles
FP_Add/Sub	1
FP_Multiply	1
FP_Divide	1
INT_Divide	1

INT_Add R1 R1 R1

☐ Data Forwarding

	Instruction	1	2	3	4	5	6	7	8	9	10	11	12
0	int_ld (R1, Offset, R6)	IF	ID	EX	MEM	WB							
1	br_taken (Offset, R2)		IF	ID									
2	int_add (R1, R6, R4)			IF									
3	br_untaken (Offset, R1)				IF	ID							
4	int_sd (R2, Offset, R4)					IF	ID	EX	MEM	WB			
5	int_add (R1, R1, R4)						IF	ID	+|- (i)	MEM	WB		

Potential Hazards:

```
RAW: Instructions 2 and 3.  Register R1.
```

Disclaimer:

- The programs and output submitted is duly written, verified and executed by me.
- I have not copied from any of my peers nor from the external resource such as internet.
- If found plagiarized, I will abide with the disciplinary action of the University.

Signature: R SHARMILA
Name: R SHARMILA
SRN: PES2UG19CS309
Section: E
Date: 11/04/2021