Microprocessor and Computer Architecture Laboratory

UE19CS256

4th Semester, Academic Year 2020-21

Date:10/04/2021

Name: PRIYA MOHATA	SRN:PES2UG19CS301	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

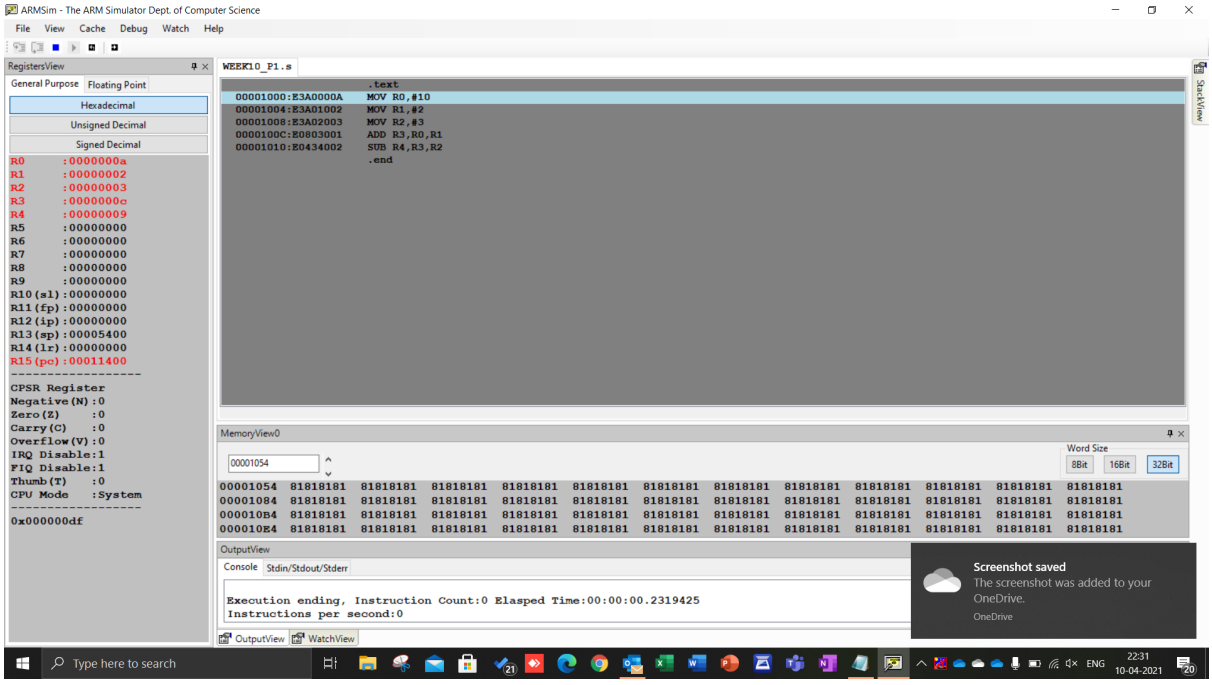
1) x = (a + b) - c;

ARM Assembly Language Code

WEEK10_P1.s

```
.text
MOV R0,#10
MOV R1,#2
MOV R2,#3
ADD R3,R0,R1
SUB R4,R3,R2
.end
```

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



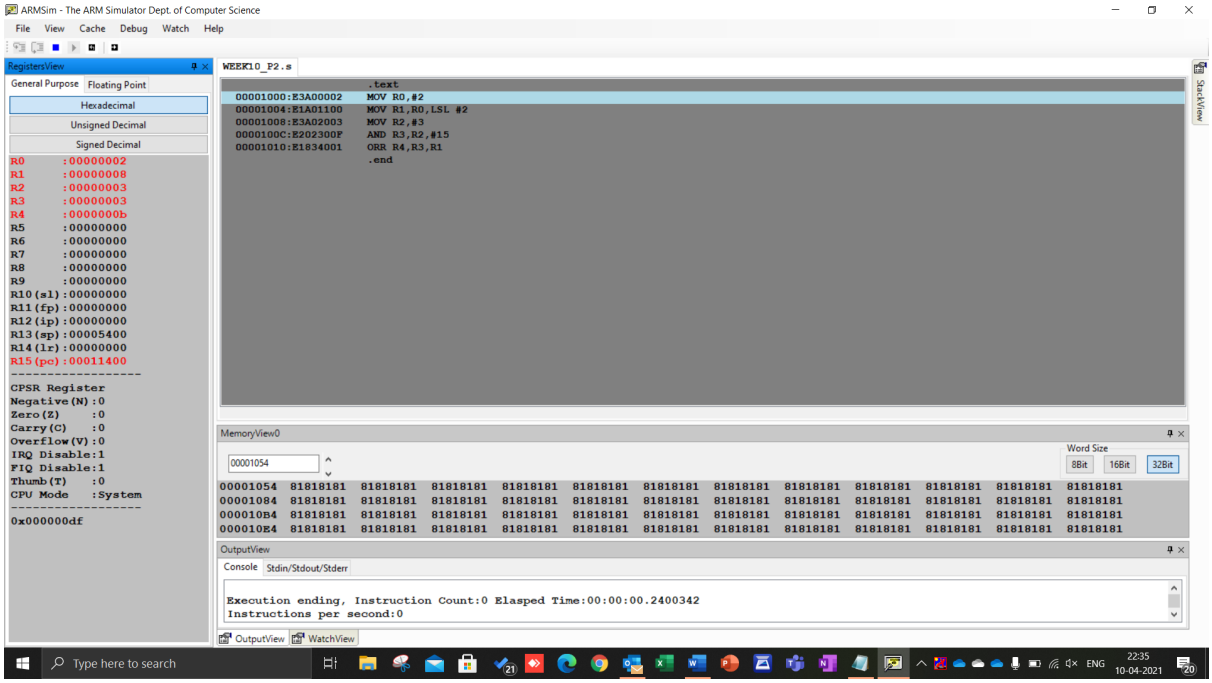
2) z = (a << 2) | (b & 15);

ARM Assembly Language Code

WEEK10_P2.s

```
.text
MOV R0,#2
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.


ARMSim - The ARM Simulator Dept. of Computer Science
File View Cache Debug Watch Help
RegistersView
General Purpose Floating Point
Hexadecimal
Unsigned Decimal
Signed Decimal
R0 :00000002
R1 :00000008
R2 :00000003
R3 :00000003
R4 :0000000b
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):00011400
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
WEEK10_P2.s
.text
00001000:E3A00002 MOV R0,#2
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
StackView
MemoryView0
00001054
Word Size
8Bit 16Bit 32Bit
00001054 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
00001084 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
000010B4 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
000010E4 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
OutputView
Console Stdin/Stdout/Stderr
Execution ending, Instruction Count:0 Elasped Time:00:00:00.2400342
Instructions per second:0
OutputView WatchView
Type here to search
ENG
22:35
10-04-2021

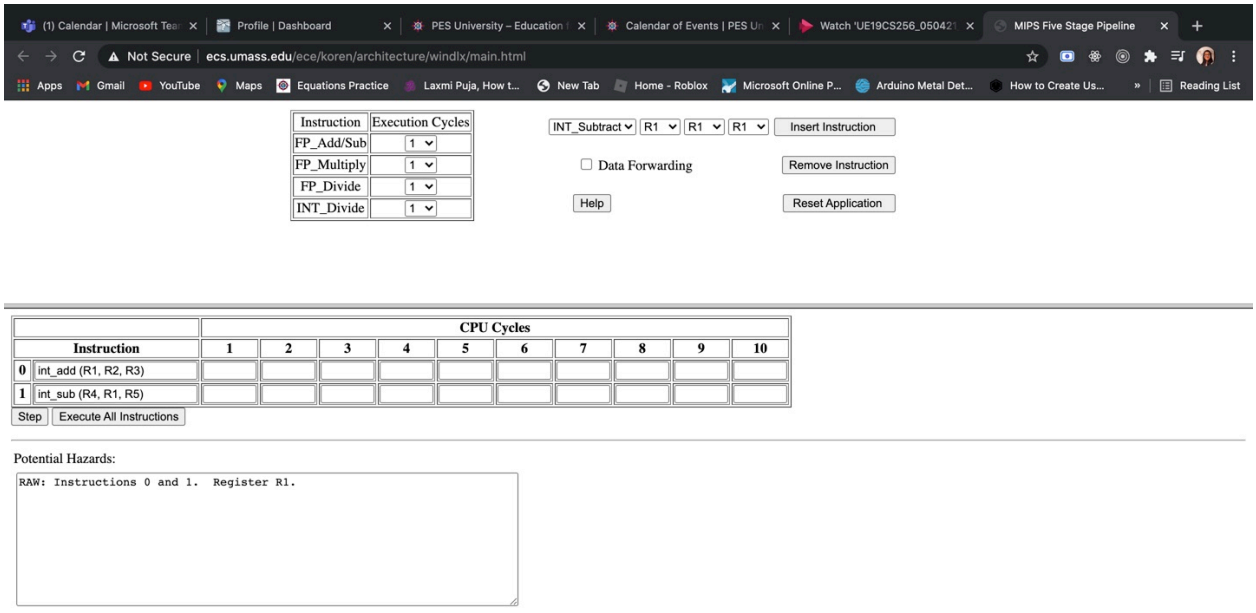
Week#____10_____ Program Number: ____2__

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

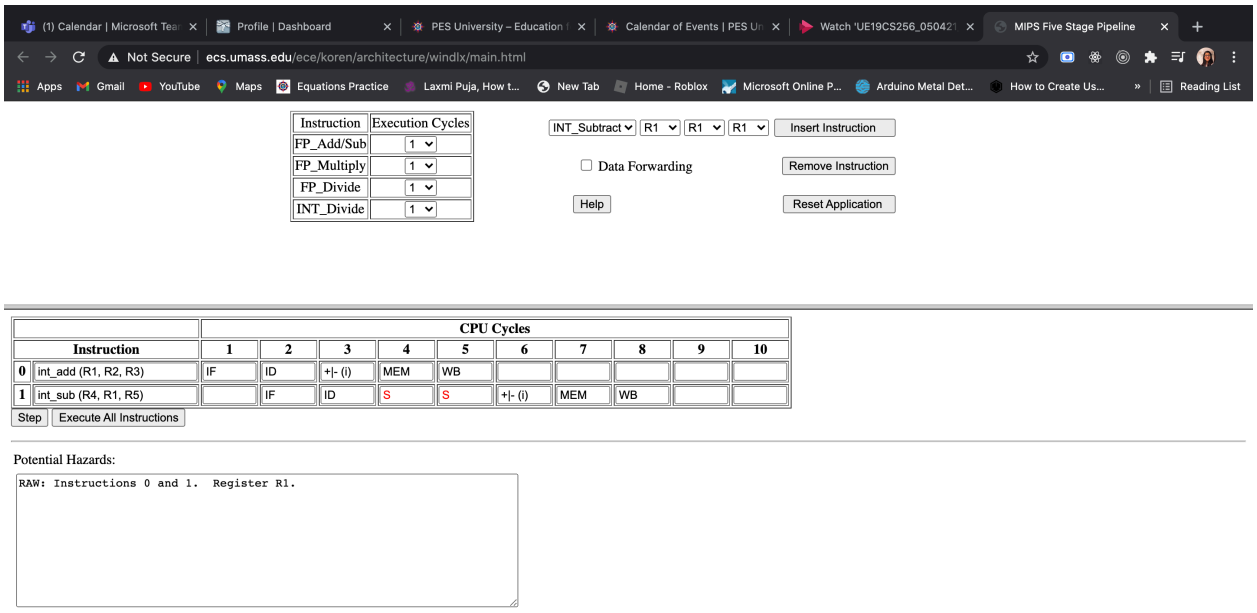
ADD R0, R1, R2
SUB R3, R0, R4.

Observe the following and note down the results.

a) Check whether there is data dependency for the second instruction?
Yes there is RAW data dependency
Related Screenshot



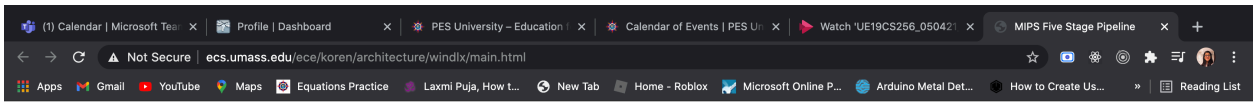
b) If yes, then, how many stall states have been introduced?
There are two stall states
Related Screenshot



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

There are no stall cycles when data forwarding was applied.

Related Screenshot



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

		CPU Cycles									
	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				

Step Execute All Instructions

Potential Hazards:

```
No Hazards Found.
```

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

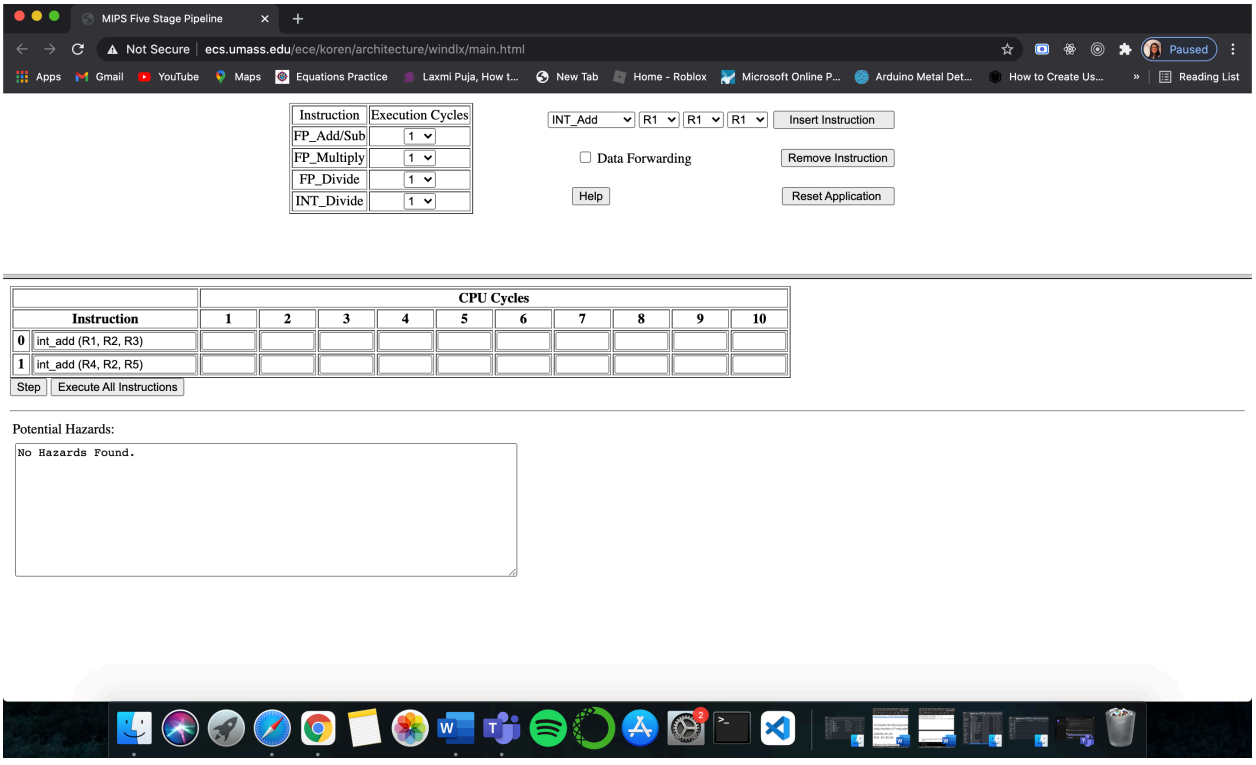
WEEK10_P3.s

```
.text
MOV R0,#2
MOV R1,#3
MOV R2,#4
ADD R3,R0,R1
ADD R4,R0,R2
.end
```

b) Find the hazards;

Related Screenshot

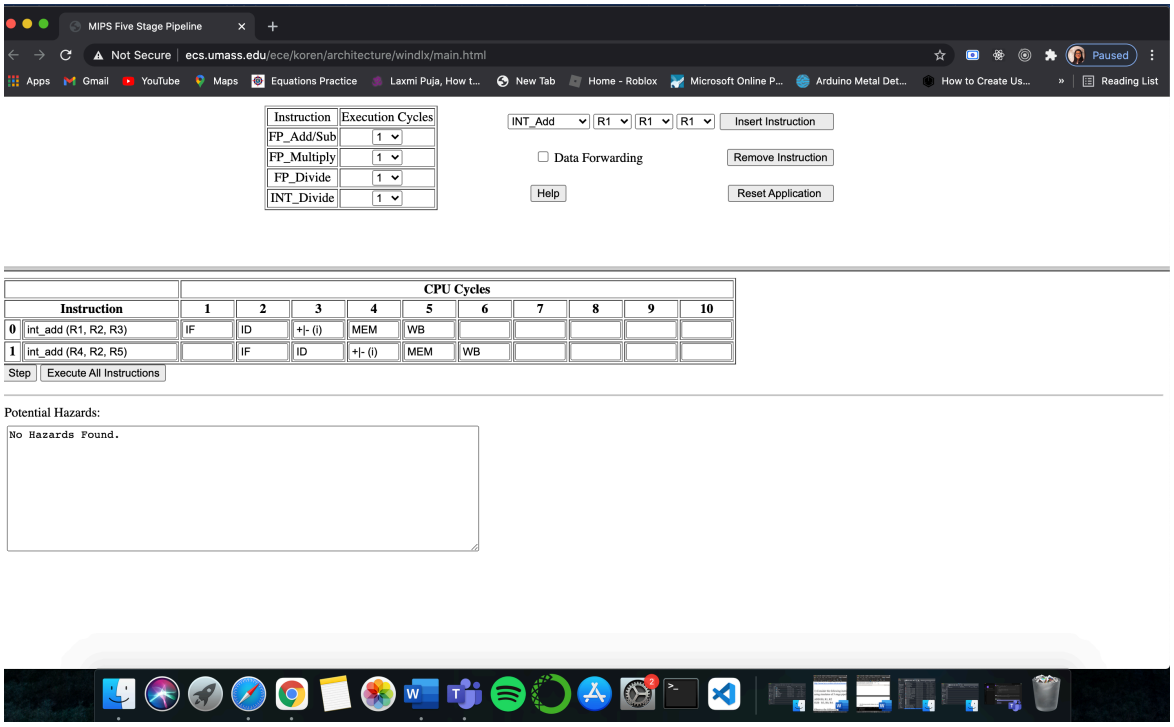
No hazards



c) Reorder the instructions to avoid pipeline stalls.

Related Screenshot

No reordering required



Week#____10_____ Program Number: ___4__

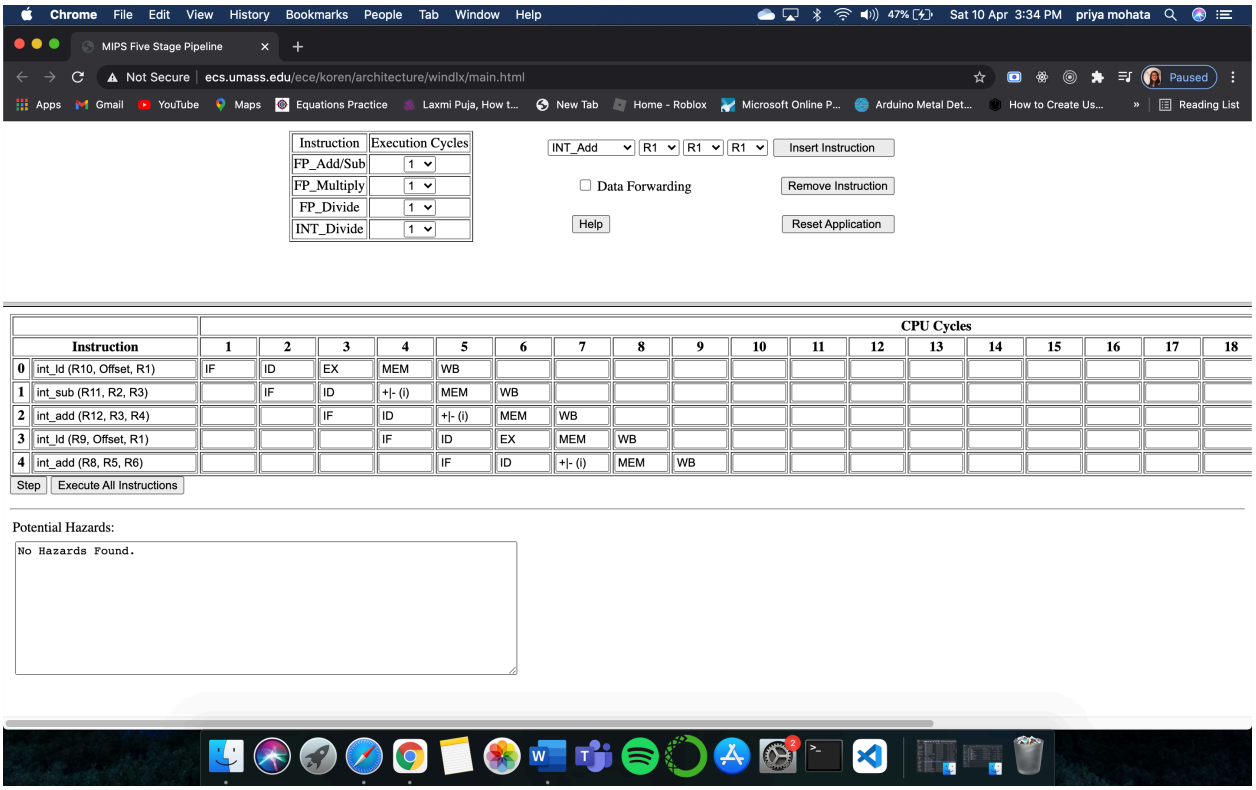
Using MIPS 5 stage pipeline architecture, execute the following instructions and avoid stall states if any.

```
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

c) Related Screenshot without stalls



Week#____10_____ Program Number: ___5__

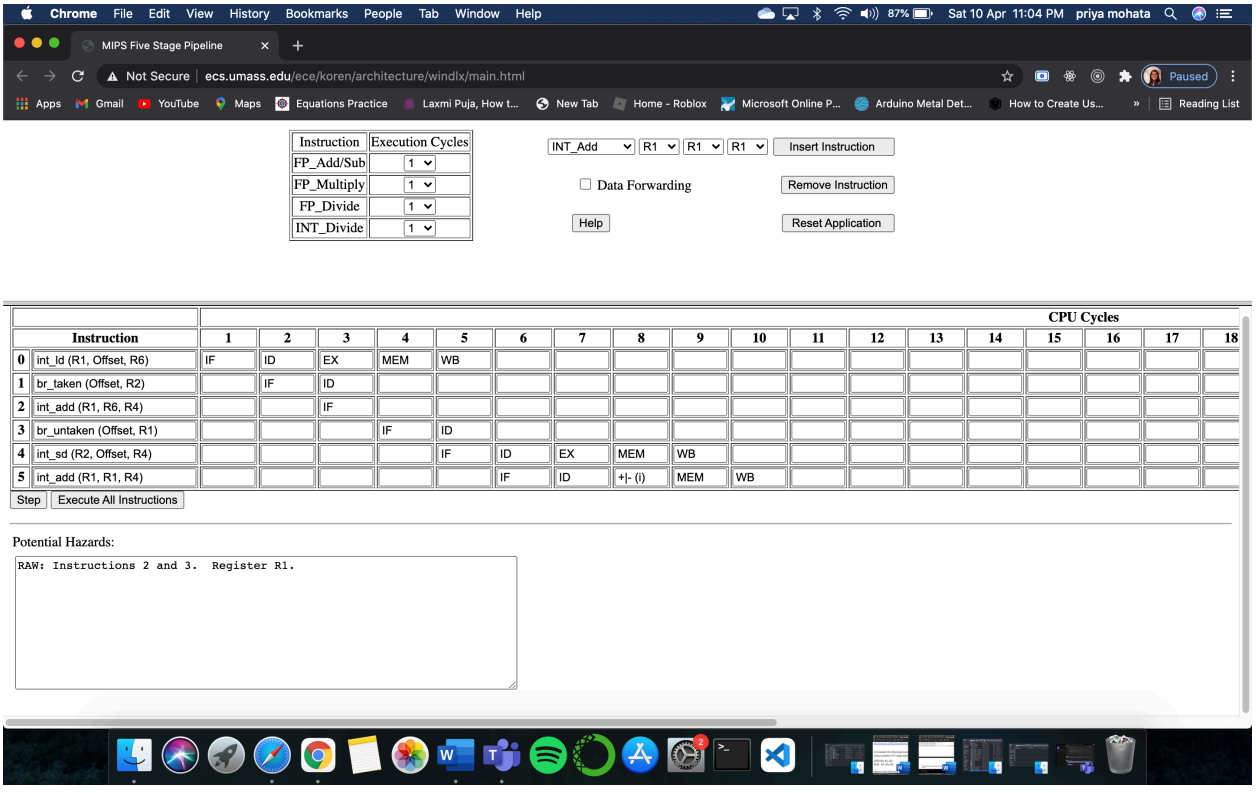
This exercise is to understand the relationship between delay slots, control hazards and branch execution in a 5 stage MIPS pipelined processor.

```
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



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: PRIYA MOHATA
Name: PRIYA MOHATA
SRN: PES2UG19CS301
Section: E
Date: 10/04/2021