Microprocessor and Computer Architecture Laboratory

UE19CS256

4th Semester, Academic Year 2020-21

Date:10-4-2021

Name: Achyut Jagini	SRN:PES2UG19CS013	Section:A

Week#____10_____Program Number: ____1__

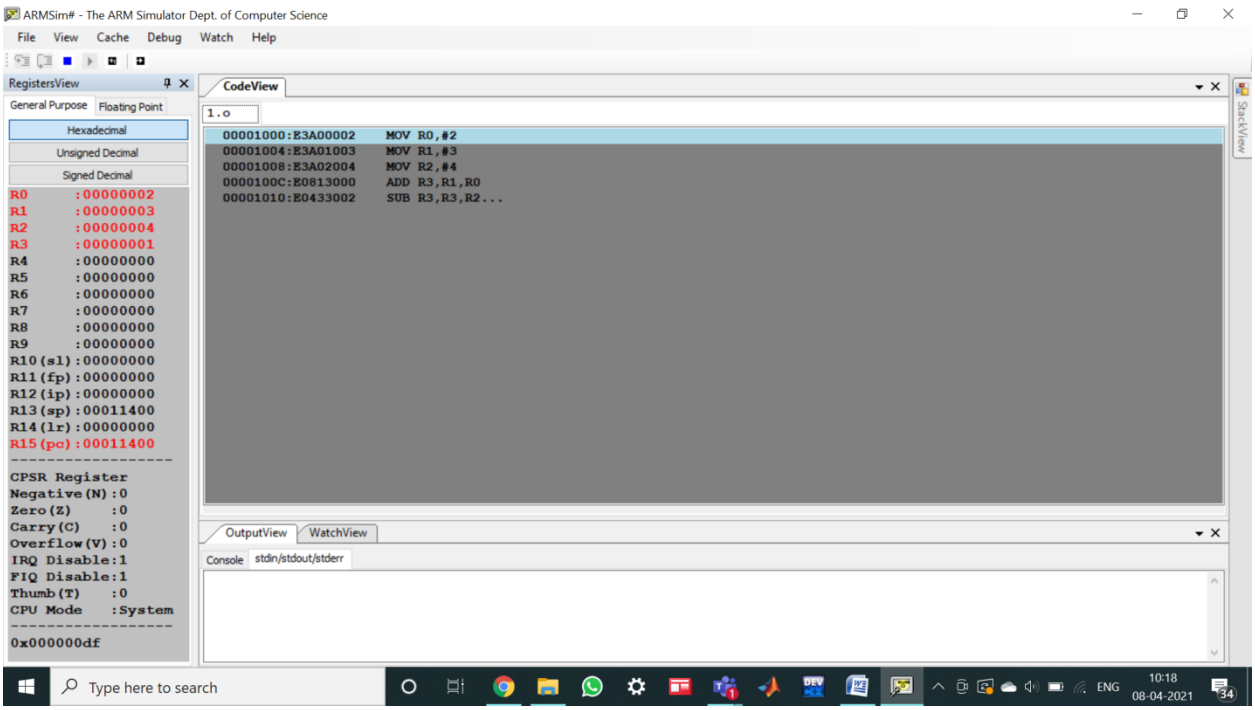
Given a C- Code convert it in its equivalent ARM Code.
These programs need to be executed on ARMSIM Simulator

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

R0=a
R1=b
R2=c
R3=x
ARM Assembly Language Code

```
MOV R0,#2
MOV R1,#3
MOV R2,#4
ADD R3,R1,R0
SUB R3,R3,R2
```

Screenshot showing the value of x, a, b, c 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 :00000003
R2 :00000004
R3 :00000001
R4 :00000000
R5 :00000000
R6 :00000000
R7 :00000000
R8 :00000000
R9 :00000000
R10(sl):00000000
R11(fp):00000000
R12(ip):00000000
R13(sp):00011400
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
CodeView
1.o
00001000:E3A00002 MOV R0,#2
00001004:E3A01003 MOV R1,#3
00001008:E3A02004 MOV R2,#4
0000100C:E0813000 ADD R3,R1,R0
00001010:E0433002 SUB R3,R3,R2...
OutputView WatchView
Console stdin/stdout/stderr
StackView
Type here to search
ENG
10:18
08-04-2021

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

R2=a

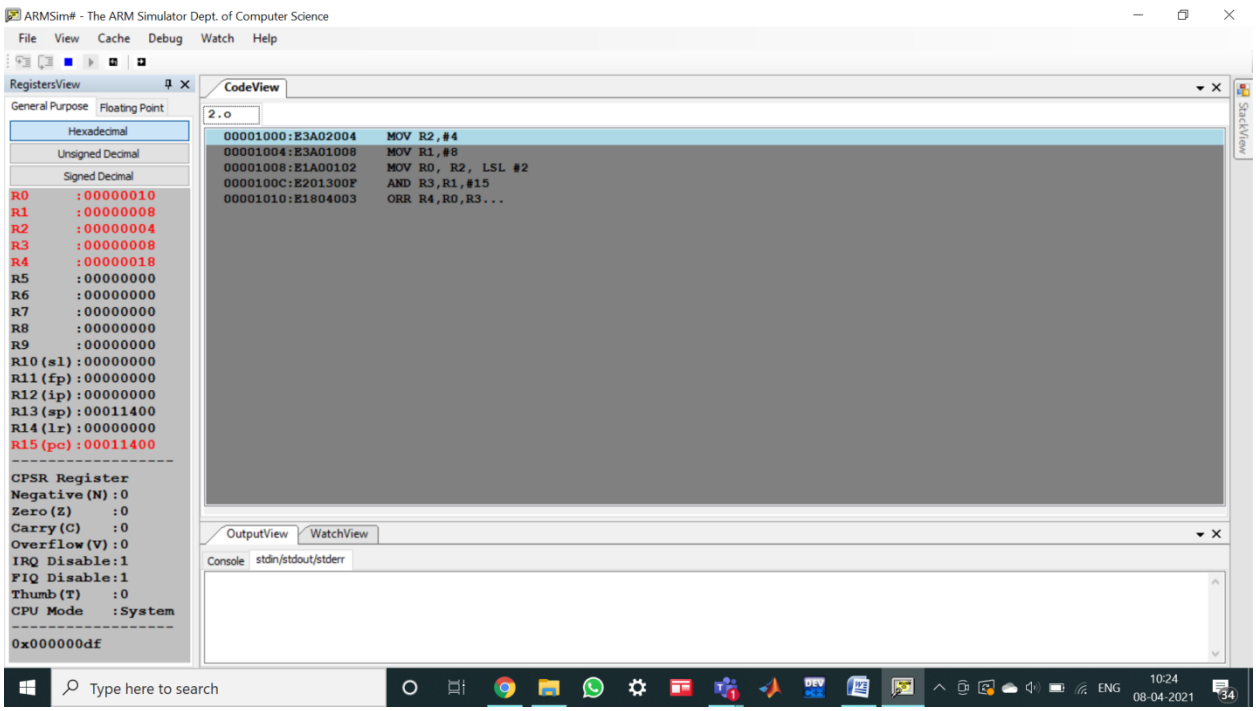
R1=b

R4=z

ARM Assembly Language Code

```
MOV R2,#4
MOV R1,#8
MOV R0, R2, LSL #2
AND R3,R1,#15
ORR R4,R0,R3
```

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



Microprocessor and Computer Architecture Laboratory

UE19CS256

4th Semester, Academic Year 2020-21

Date:10-4-2021

Name: Achyut Jagini	SRN: PES2UG19CS013	Section:

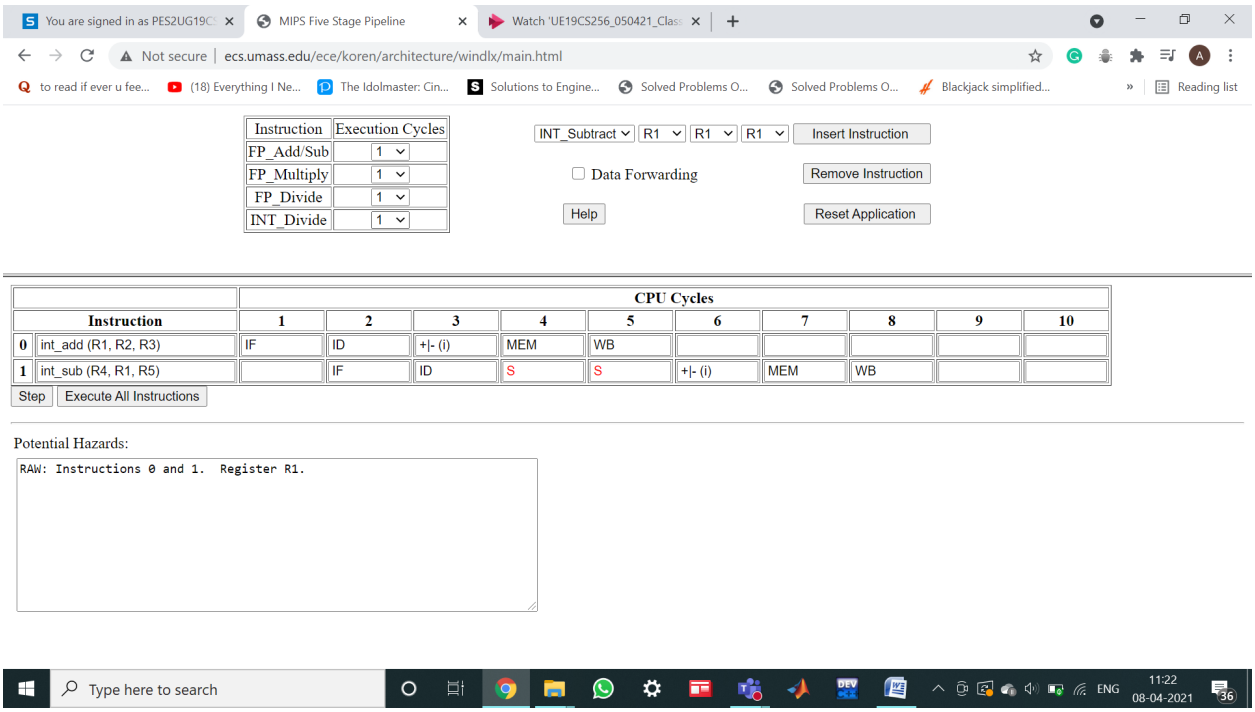
Week#____10_____ Program Number: ____2__

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.

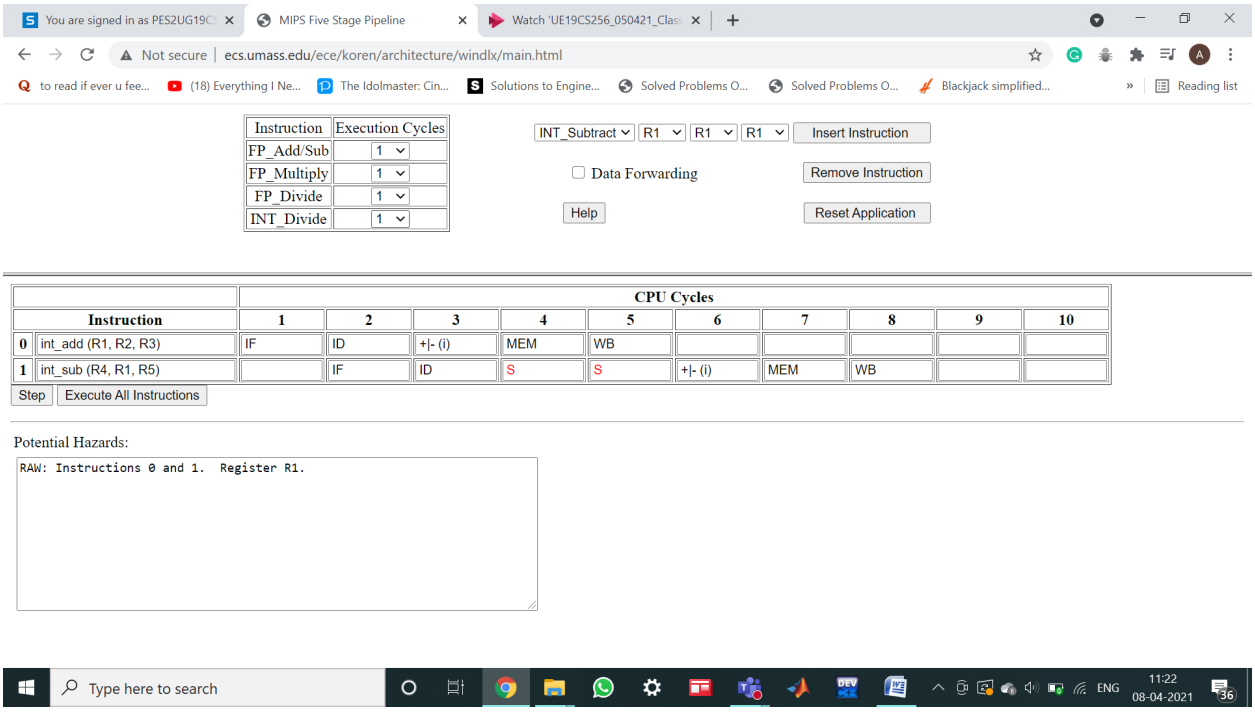
Observe the following and note down the results.

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



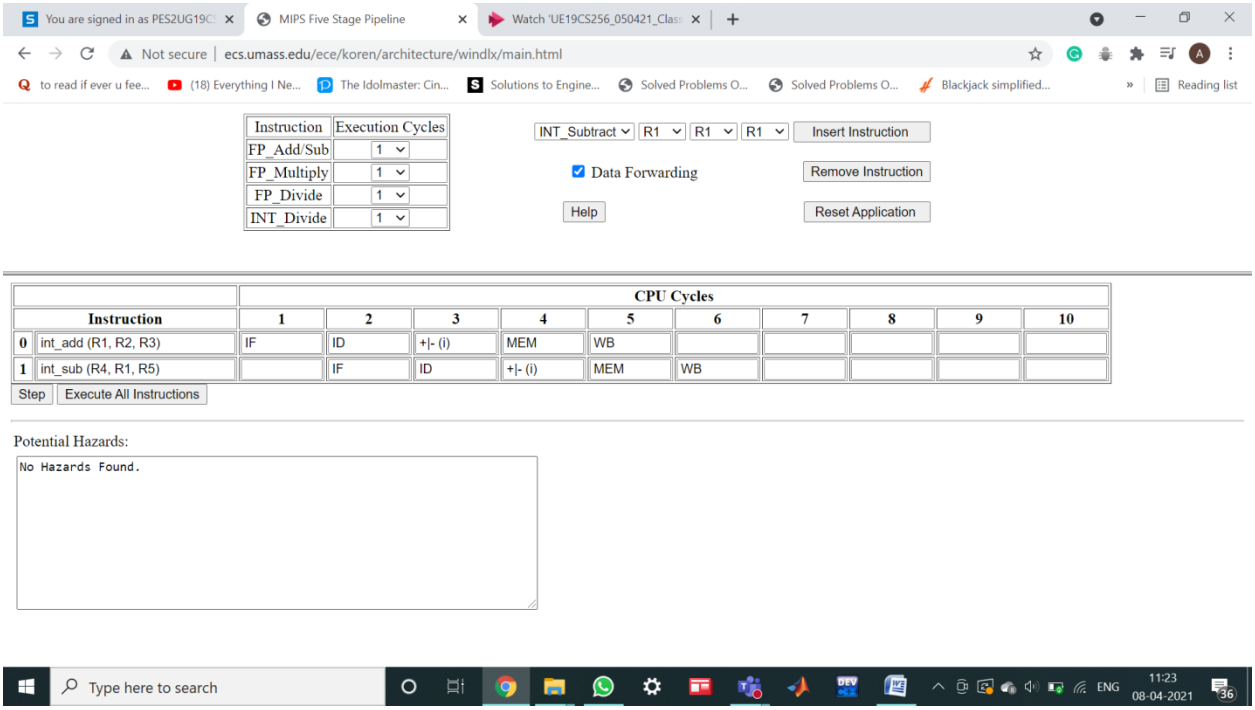
yes

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



2

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


You are signed in as PES2UG19C
MIPS Five Stage Pipeline
Watch 'UE19CS256_050421_Class
Not secure | ecs.umass.edu/ece/koren/architecture/windlx/main.html
to read if ever u fee...
(18) Everything I Ne...
The Idolmaster: Cin...
Solutions to Engine...
Solved Problems O...
Solved Problems O...
Blackjack simplified...
Reading list
Instruction
Execution Cycles
FP_Add/Sub
FP_Multiply
FP_Divide
INT_Divide
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.
Type here to search
ENG
11:23
08-04-2021

Microprocessor and Computer Architecture Laboratory

UE19CS256

4th Semester, Academic Year 2020-21

Date:10-4-2021

Name: Achyut Jagini	SRN: PES2UG19CS013	Section: A

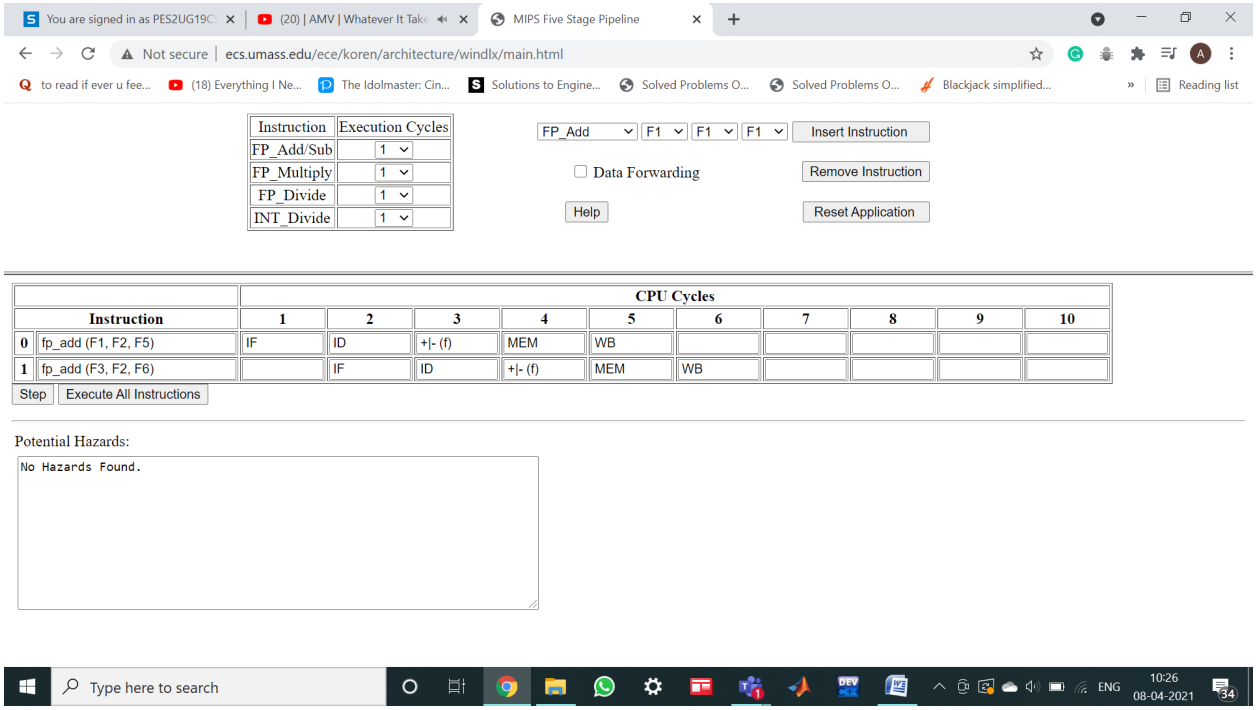
Week#____10_____ Program Number: ____3__

Consider the following code segment in C.

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

a) Write the code using MIPS 5 STAGE pipeline architecture.
ADD R1,R2,R5
ADD R3,R2,R6

b) Find the hazards;



c) Reorder the instructions to avoid pipeline stalls.

The solution is No Data dependency is seen in the above instructions.

Hence no stall states.

Microprocessor and Computer Architecture Laboratory

UE19CS256

4th Semester, Academic Year 2020-21

Date:10-4-2021

Name: Achyut Jagini	SRN: PES2UG19CS013	Section: A

Week#____10_____ Program Number: ___4__

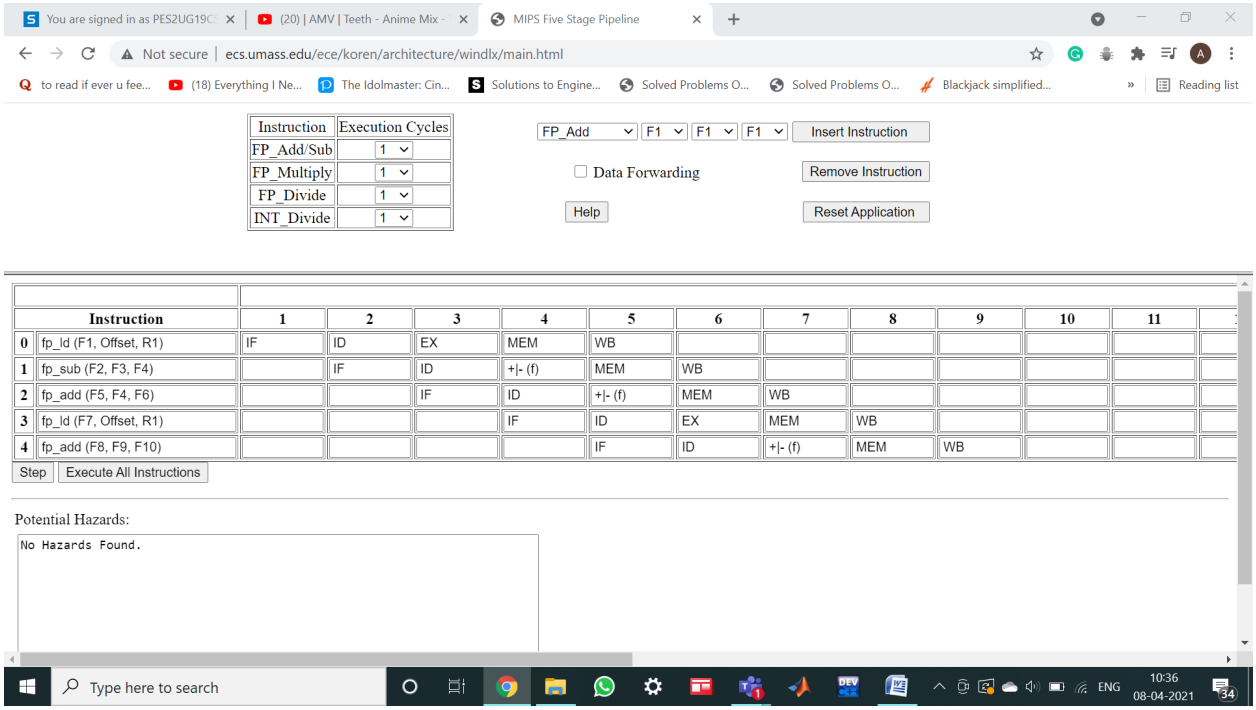
Using MIPS 5 stage pipeline architecture, execute the following instructions and avoid stall states if any.

```
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


You are signed in as PES2UG19C
(20) | AMV | Teeth - Anime Mix -
MIPS Five Stage Pipeline
Not secure | ecs.umass.edu/ece/koren/architecture/windlx/main.html
to read if ever u fee...
(18) Everything I Ne...
The Idolmaster: Cin...
Solutions to Engine...
Solved Problems O...
Solved Problems O...
Blackjack simplified...
Reading list
Instruction
Execution Cycles
FP_Add/Sub
FP_Multiply
FP_Divide
INT_Divide
FP_Add
F1
Insert Instruction
Data Forwarding
Remove Instruction
Help
Reset Application
Instruction
1
2
3
4
5
6
7
8
9
10
11
0 fp_ld (F1, Offset, R1) IF ID EX MEM WB
1 fp_sub (F2, F3, F4) IF ID +|- (f) MEM WB
2 fp_add (F5, F4, F6) IF ID +|- (f) MEM WB
3 fp_ld (F7, Offset, R1) IF ID EX MEM WB
4 fp_add (F8, F9, F10) IF ID +|- (f) MEM WB
Step
Execute All Instructions
Potential Hazards:
No Hazards Found.
Type here to search
ENG
10:36
08-04-2021

Microprocessor and Computer Architecture Laboratory

UE19CS256

4th Semester, Academic Year 2020-21

Date:10-4-2021

Name: Achyut Jagini	SRN: PES2UG19CS013	Section: A

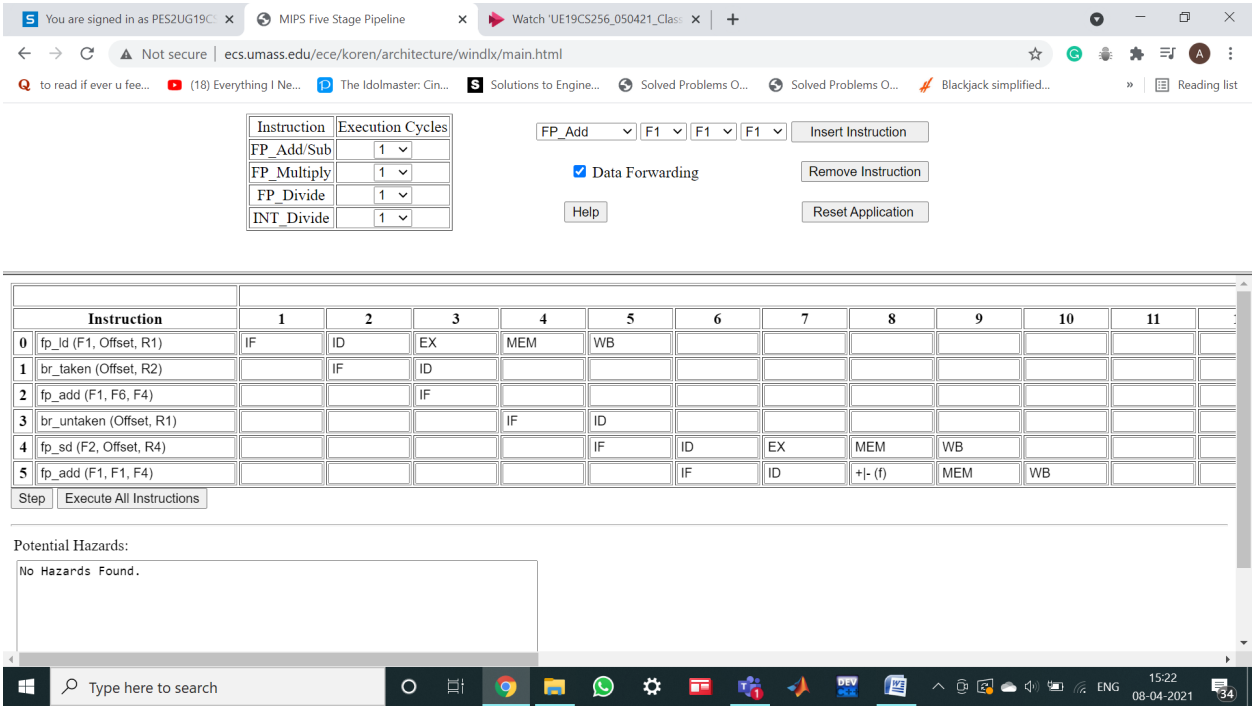
Week#____10_____ Program Number: ___5__

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.



No hazards found