**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**

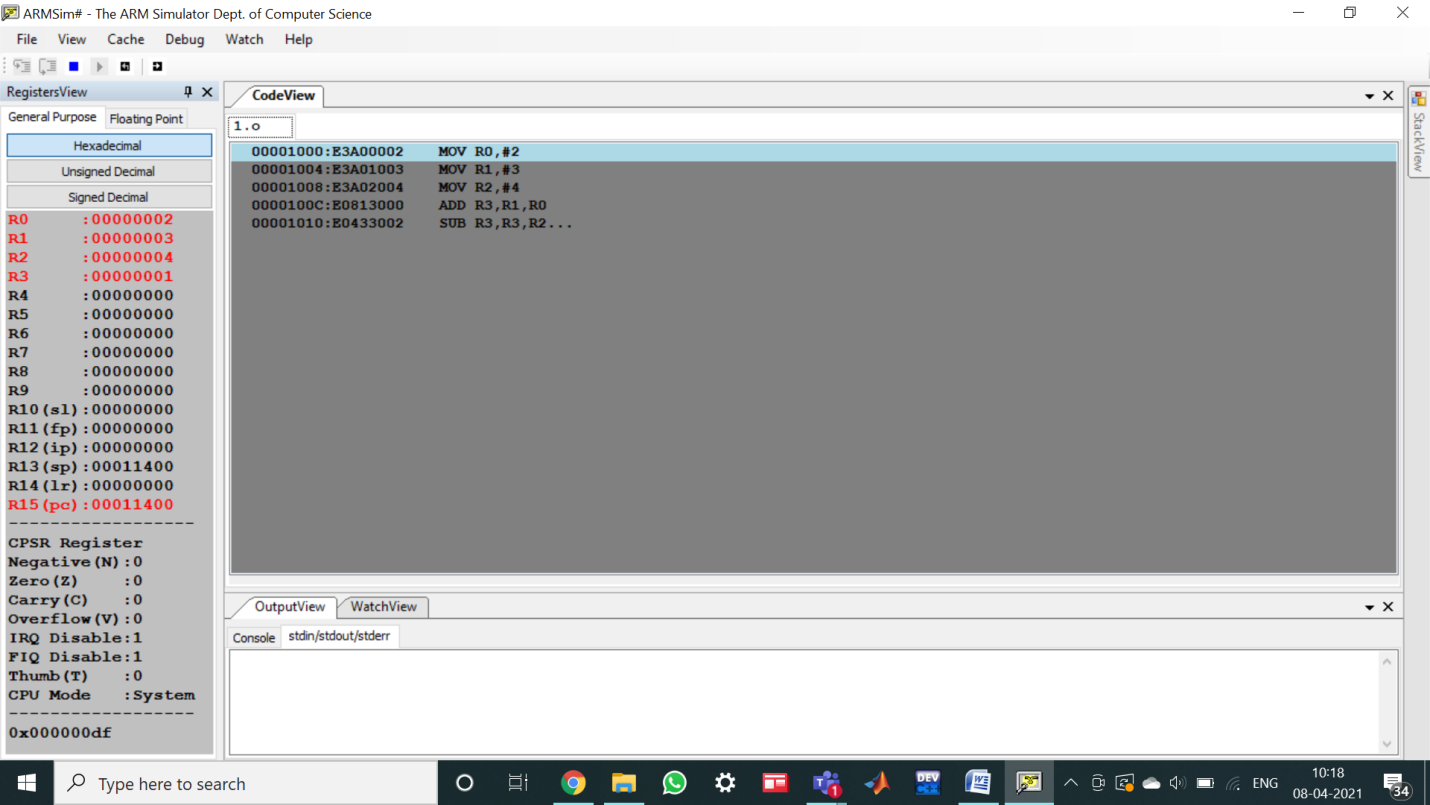
**MOV R1,#3**

**MOV R2,#4**

**ADD R3,R1,R0**

**SUB R3,R3,R2**

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

****

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

R2=a

R1=b

R4=z

**ARM Assembly Language Code**

**MOV R2,#4**

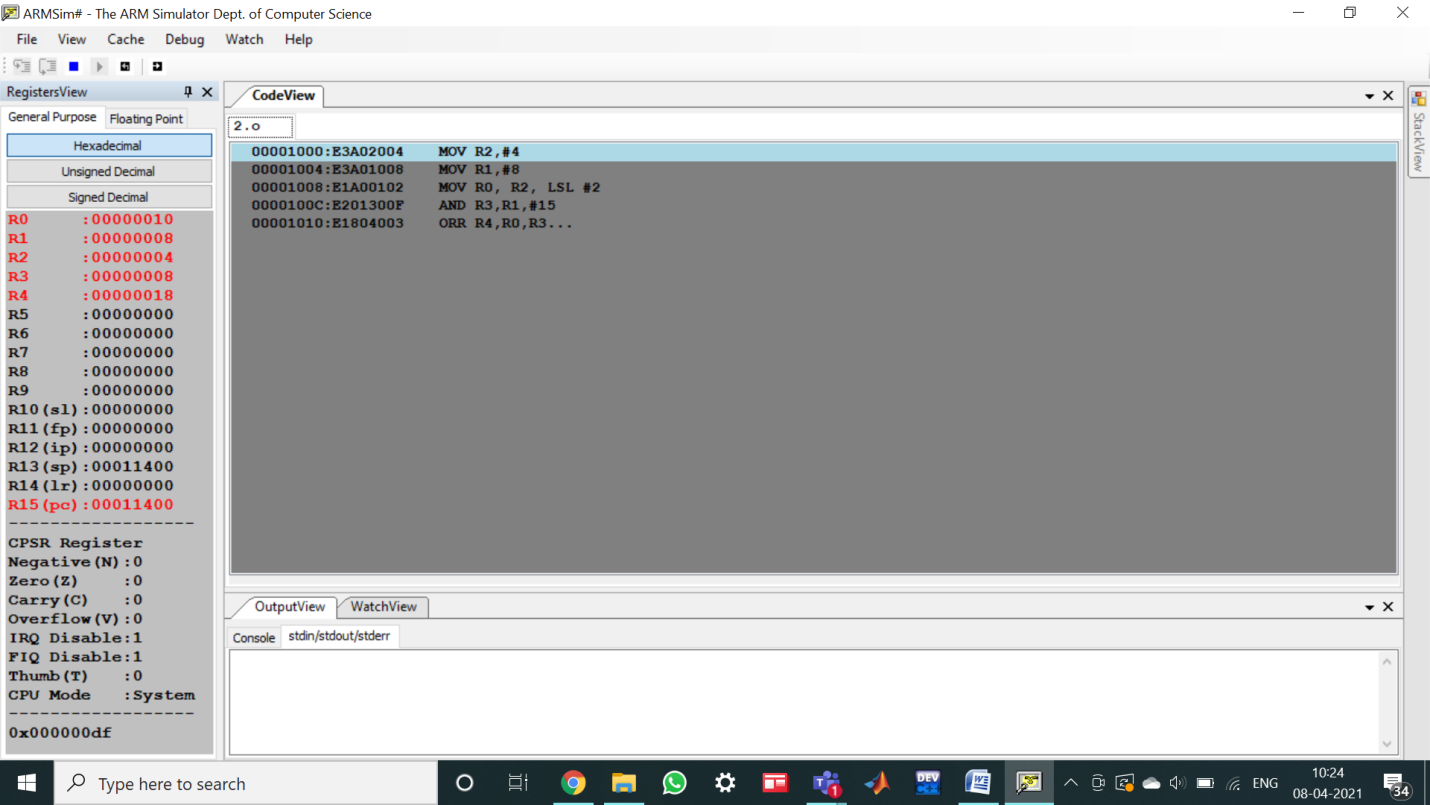
**MOV R1,#8**

**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\_\_

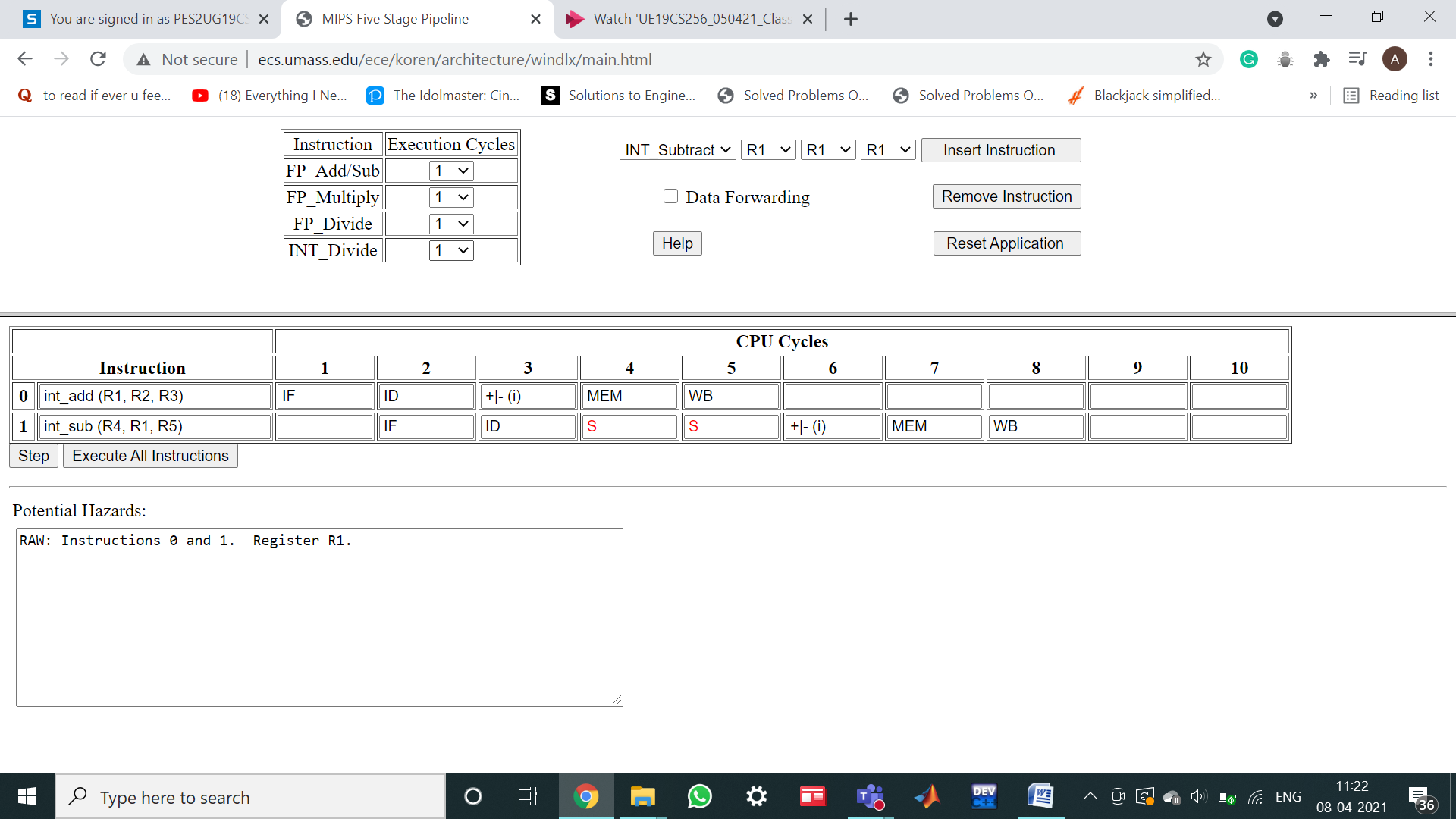
1) Consider the following instructions. Execute these instructions using simulator of 5 stage pipeline of MIPS architecture.

ADD R0, R1, R2

SUB R3, R0, R4.

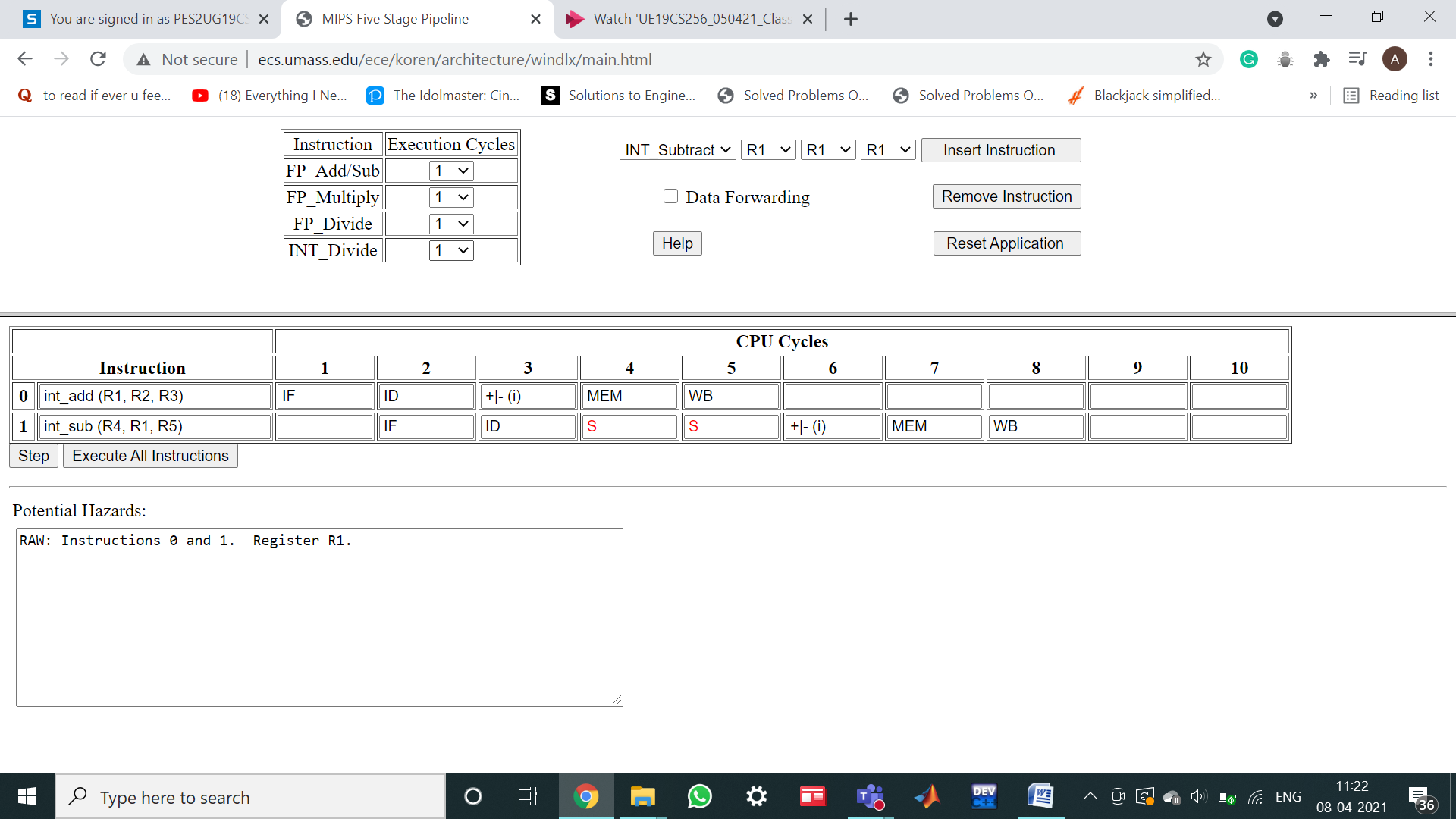
Observe the following and note down the results.

1. Check whether there is data dependency for the second instruction?



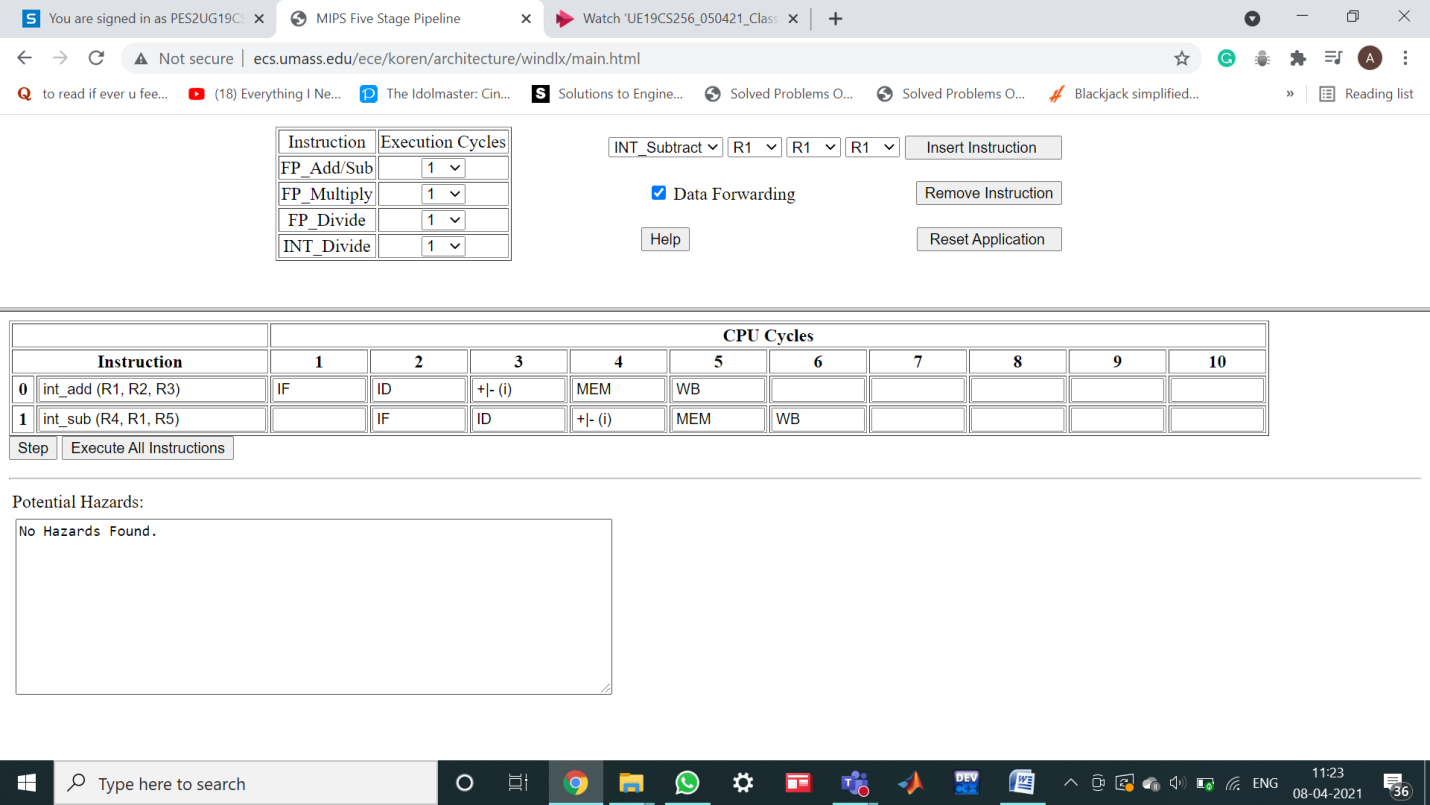
yes

1. If yes, then, how many stall states have been introduced?



2

1. If data forwarding is applied how many stall states have been reduced?



**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;

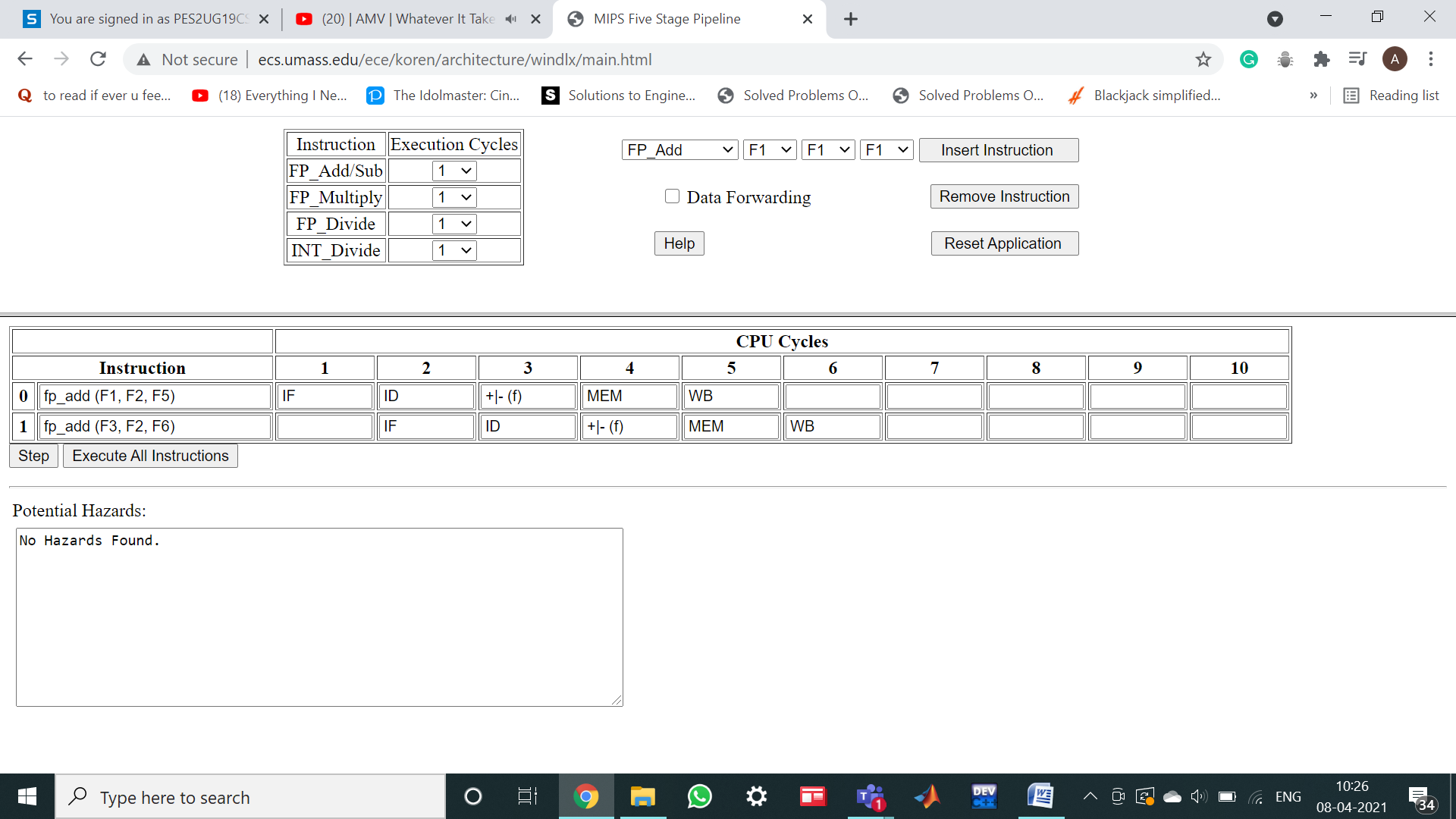
C = B + F;

1. Write the code using MIPS 5 STAGE pipeline architecture.

**ADD R1,R2,R5**

**ADD R3,R2,R6**

1. 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

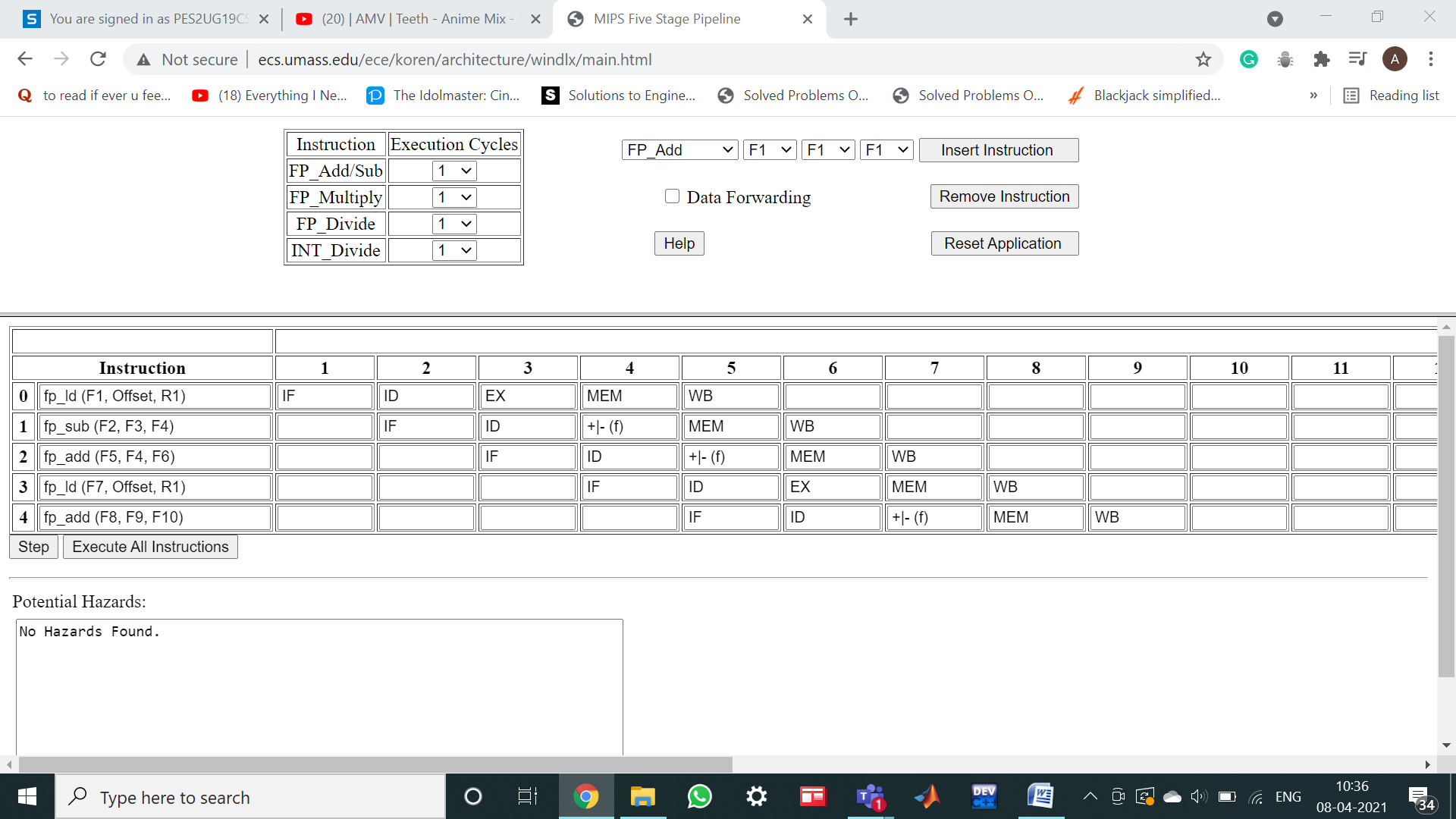
LW $13, 24($1)

ADD $14, $5, $6

**a)Related Screenshot with stalls**

**No stalls**

**b)Related Screenshot without stalls**



**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

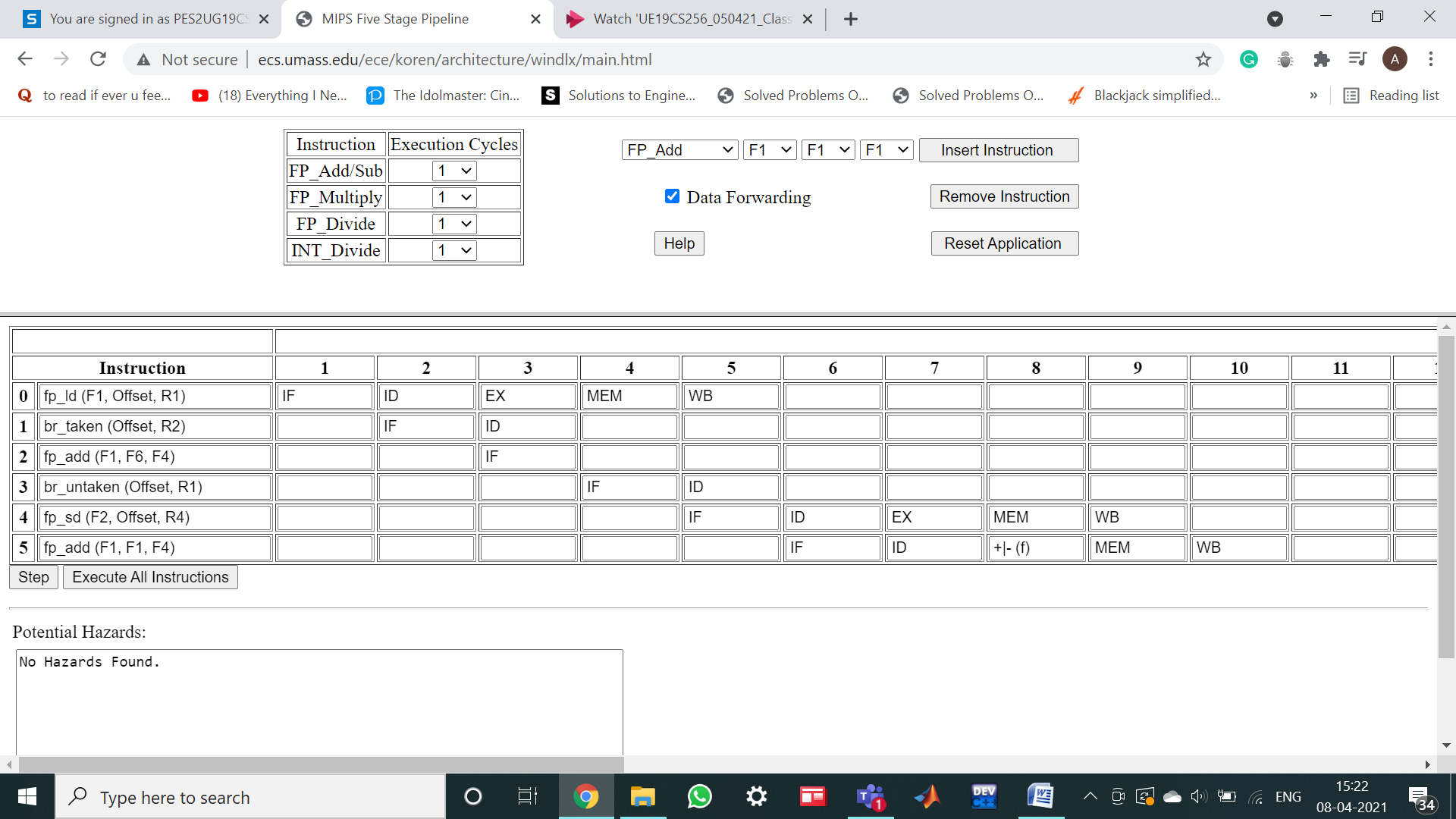
Label2: BEQ $1, $2, Label1 : branch not taken

SW $2, 20($4)

ADD $1, $1, $4

Assume full data forwarding and predict- taken branch prediction.

Note the observations.

****

**No hazards found**