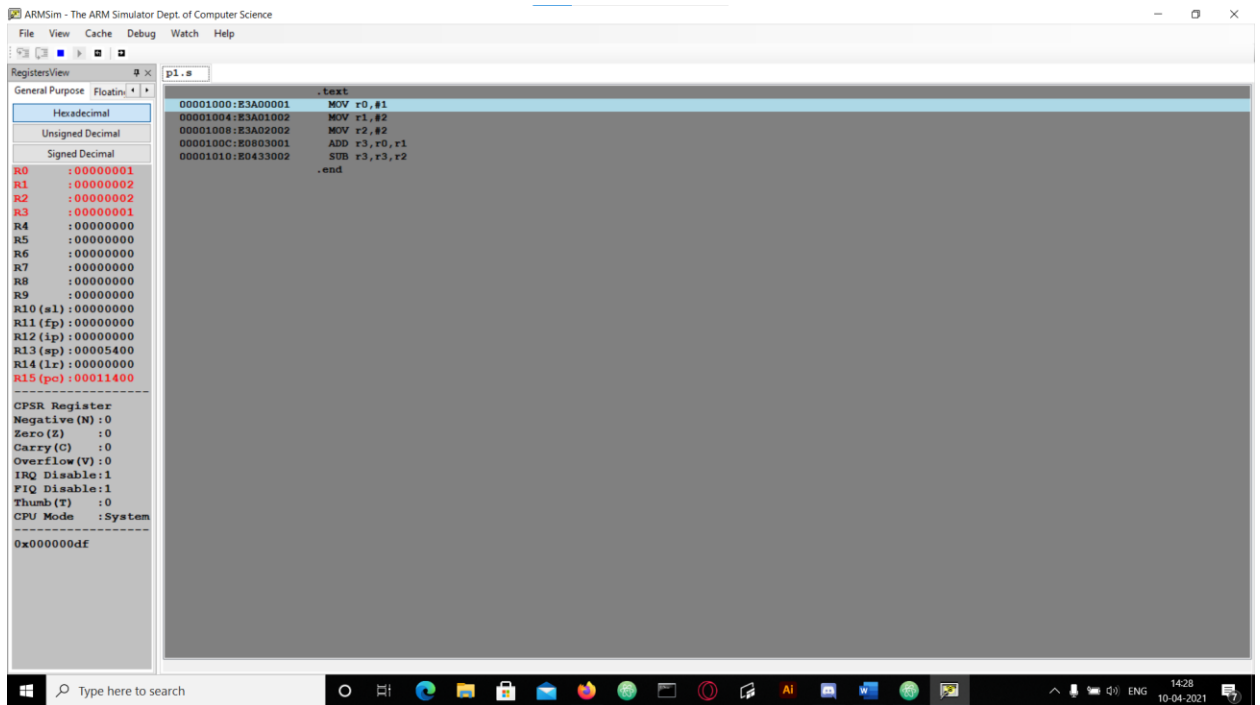
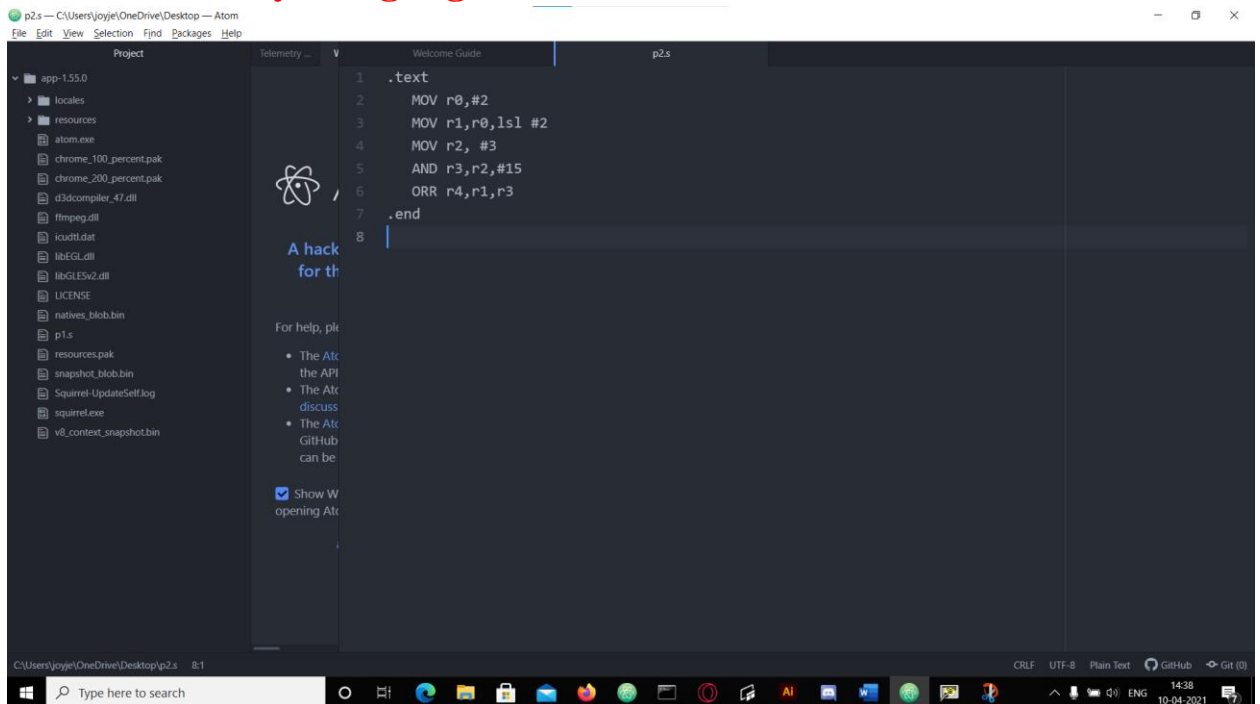


Screenshot showing the value of x, a, b, c in the register window.

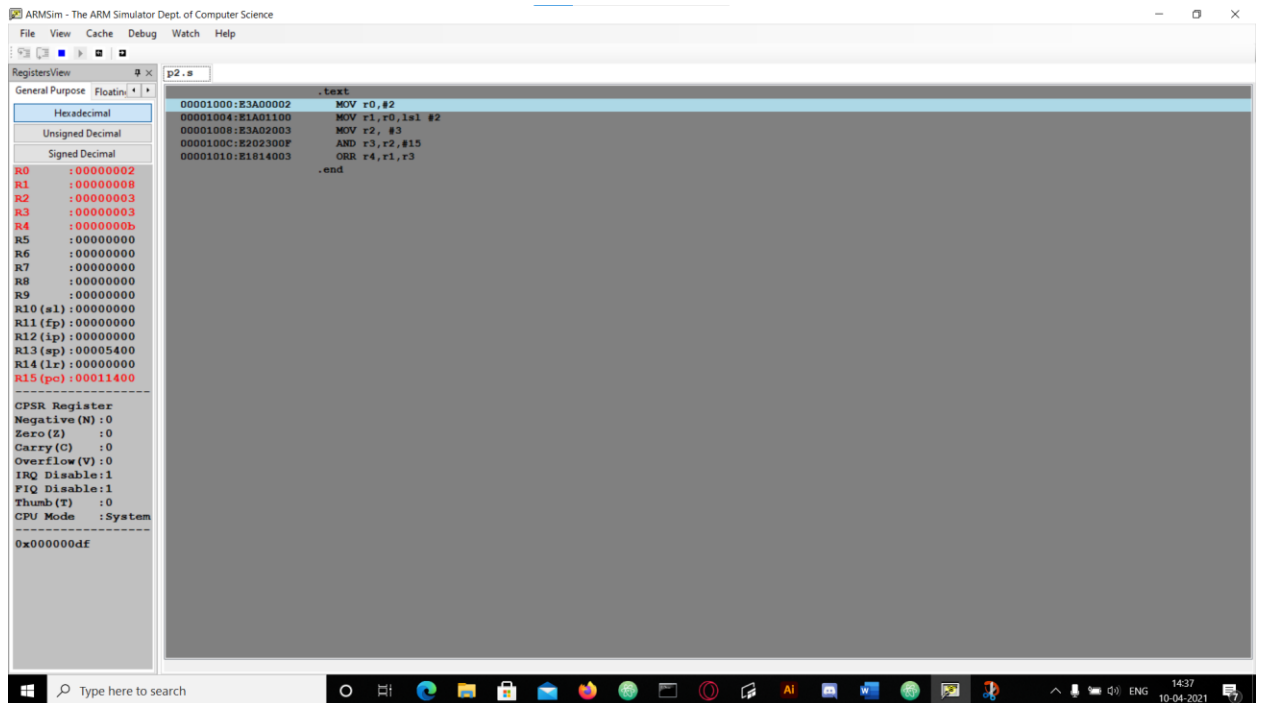


2) $z = (a \ll 2) | (b \& 15);$

ARM Assembly Language Code



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:

Name:OP JOY JEFFERSON	SRN:PES2UG19CS270	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.

MIPS Five Stage Pipeline

www.ecs.umass.edu/ece/koren/architecture/windib/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 Remove Instruction

Help Reset Application

Instruction	CPU Cycles									
	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		

Step Execute All Instructions

Potential Hazards:

RAW: Instructions 0 and 1. Register R1.

Observe the following and note down the results.

- Check whether there is data dependency for the second instruction?
Yes, there is data dependency (RAW) for the second screenshot.

Related Screenshot

(inserted above).

- If yes, then, how many stall states have been introduced?
2 stalls

Related Screenshot

(inserted above).

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

Related Screenshot

MIPS Five Stage Pipeline

www.ecs.umass.edu/ece/koren/architecture/windbx/main.html

Instruction	Execution Cycles
FP_Add/Sub	2
FP_Multiply	1
FP_Divide	1
INT_Divide	1

FP_Subtract

F1

F1

F1

Insert Instruction

☒ Data Forwarding

Remove Instruction

Help

Reset Application

Instruction	CPU Cycles									
	1	2	3	4	5	6	7	8	9	10
0 fp_add (F1, F2, F3)	IF	ID	+- (f)	MEM	WB					
1 fp_sub (F4, F1, F5)		IF	ID	+- (f)	MEM	WB				

Step

Execute All Instructions

Potential Hazards:

No Hazards Found.

Microprocessor and Computer Architecture Laboratory

UE19CS256

4th Semester, Academic Year 2020-21

Date:

Name: OP JOY JEFFERSON	SRN:PES2UG19CS270	Section:E
------------------------	-------------------	-----------

Week# 10

Program Number: 3

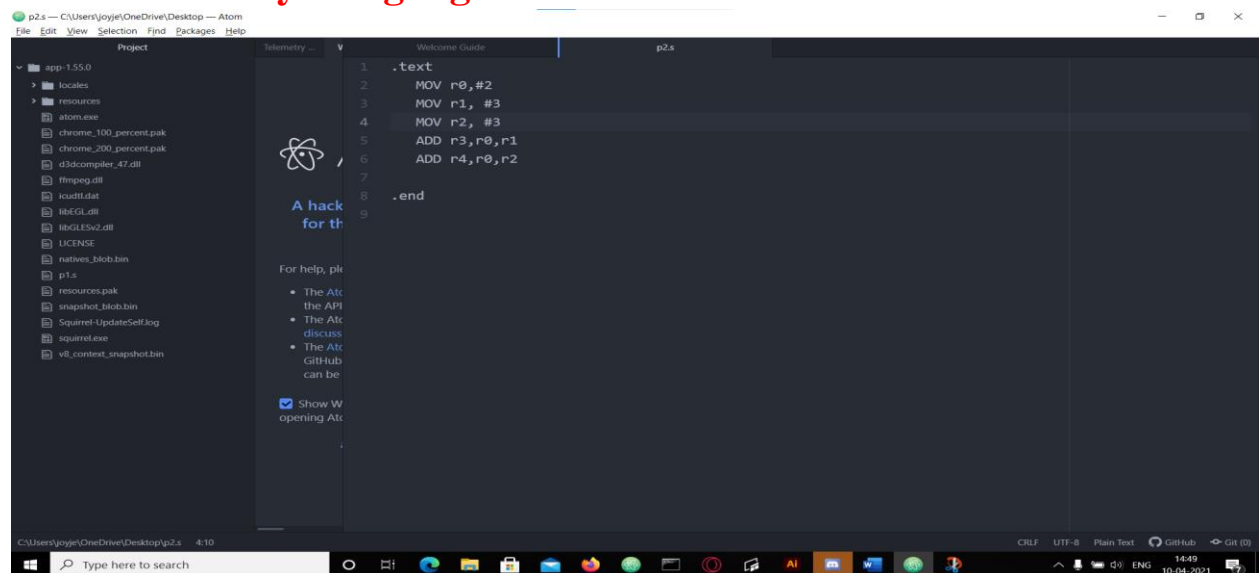
Consider the following code segment in C.

A = B + E;

C = B + F;

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

ARM Assembly Language Code



b) Find the hazards;

Related Screenshot

The screenshot shows the MIPS Five Stage Pipeline simulator interface. At the top, there is a browser window with the URL www.ecs.umass.edu/ece/koren/architecture/windlx/main.html. Below the browser, there is a control panel with a table for instructions and execution cycles, a dropdown menu for the instruction type (FP_Add), and buttons for inserting, removing, and resetting instructions. The main part of the interface is a large table showing the execution of two instructions over 10 CPU cycles. The first instruction is 'fp_add (F1, F2, F3)' and the second is 'fp_add (F4, F2, F5)'. The table shows the stages of execution (IF, ID, EX, MEM, WB) and the values of the registers (F1, F2, F3, F4, F5) at each cycle. Below the table, there is a section for 'Potential Hazards' which states 'No Hazards Found.'.

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

FP_Add F1 F1 F1 Insert Instruction
☐ Data Forwarding Remove Instruction
Help Reset Application

		CPU Cycles									
Instruction		1	2	3	4	5	6	7	8	9	10
0	fp_add (F1, F2, F3)	IF	ID	+ - (f)	MEM	WB					
1	fp_add (F4, F2, F5)		IF	ID	+ - (f)	MEM	WB				

Step Execute All Instructions

Potential Hazards:
No Hazards Found.

c) Reorder the instructions to avoid pipeline stalls.

Related Screenshot

No reordering required.

Microprocessor and Computer Architecture Laboratory

UE19CS256

4th Semester, Academic Year 2020-21

Date:

Name: OP JOY JEFFERSON	SRN:PES2UG19CS270	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

Microprocessor and Computer Architecture Laboratory

UE19CS256

4th Semester, Academic Year 2020-21

Date:

Name: OP JOY JEFFERSON	SRN: PES2UG19CS270	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

FP_Add F1 F1 F1

☒ Data Forwarding

Help

Insert Instruction

Remove Instruction

Reset Application

		CPU Cycles																	
Instruction		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
0	fp_ld (F1, Offset, R6)	IF	ID	EX	MEM	WB													
1	br_taken (Offset, R2)		IF	ID															
2	fp_add (F1, F6, F4)			IF															
3	br_untaken (Offset, R1)				IF	ID													
4	fp_sd (F2, Offset, R4)					IF	ID	EX	MEM	WB									
5	fp_add (F1, F1, F4)						IF	ID	+ - (f)	MEM	WB								
Step	Execute All Instructions																		

Potential Hazards:

No Hazards Found.

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: joy

Name: OP JOY JEFFERSON

SRN: PES2UG19CS270

Section: E

Date: 10/04/2021