**Given a C- Code convert it in its equivalent Arm Code.**

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

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

**Introduction to 5 stage pipeline Simulator.**

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

ADD R0, R1, R2

SUB R3, R0, R4.

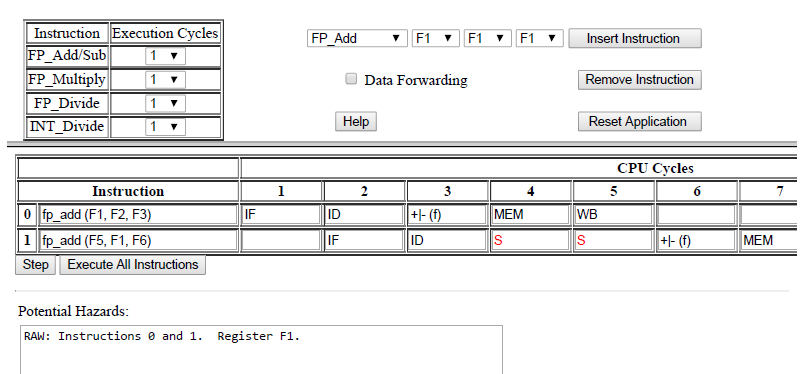
Observe the following and note down the results.

Check whether there is data dependency for the second instruction?

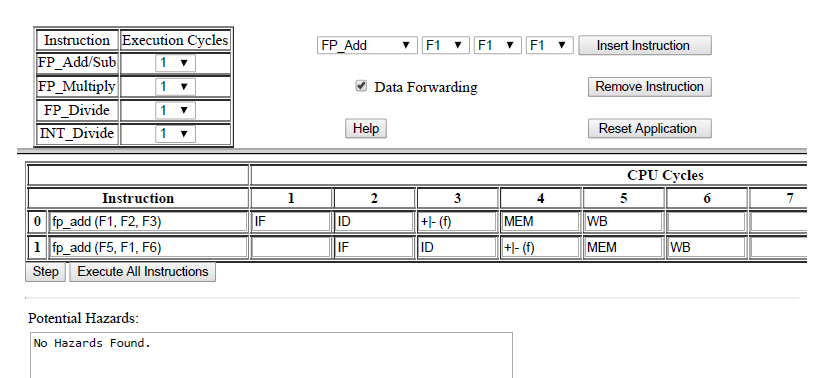
If yes, then, how many stall states have been introduced?

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

1. The second instruction has data dependency. It introduces 2 stall clock cycles. This is without data forwarding.



1. 2 stall states have been introduced.
2. With data forwarding, it introduces no stall state.



2) Consider the following code segment in C.

A = B + E;

C = B + F;

3) 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

4) 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.