Computer Architecture (COL 216)

Assignment 9: Overcoming Data Hazards in a Pipelined Processor

ENTRY NO. 2018CS10354_2018CS10348

FORWARDING :

In Assignment 8, We were using a stall to tackle some of the problems of dependencies but because of stalling, we have to give some extra cycles to complete the instruction while some of the stages remain quiet. To overcome this issue we have to implement the ***forwarding*** so no stage remains still.

As we know that most of the instructions finished it's job of ALU part in execution state so what we are doing is that we will check the instruction in memory stage part and write back stage that if there is any instruction which can provide the it's output as a input to instruction which is at execution stage, if it happens then we directly take the output of memory or write back stage's instruction as a input for the instruction which is in execution.

Same process we do for the memory stage and check that if we can use the output of WB stage as input for the memory stage.

For example:

Add $t1, $t2, $t3

Add $t4, $t1, $t1

So both instructions are dependent as their input and output depend. As in pipelining, 'add' finish it's job at WB stage but it's result is required by the coming instruction at the execution because there only it would do it ($t1+$t1) so waiting

for two cycle for completion of first and stalling second at the execution, what we will is that we will take it's output from memory stage and use it so that first inst don't have to wait for completion of second stage.

And we have done forwarding for every dependency so that we don't have to stall at any cost.