Report - (Lab5_Pipeline)

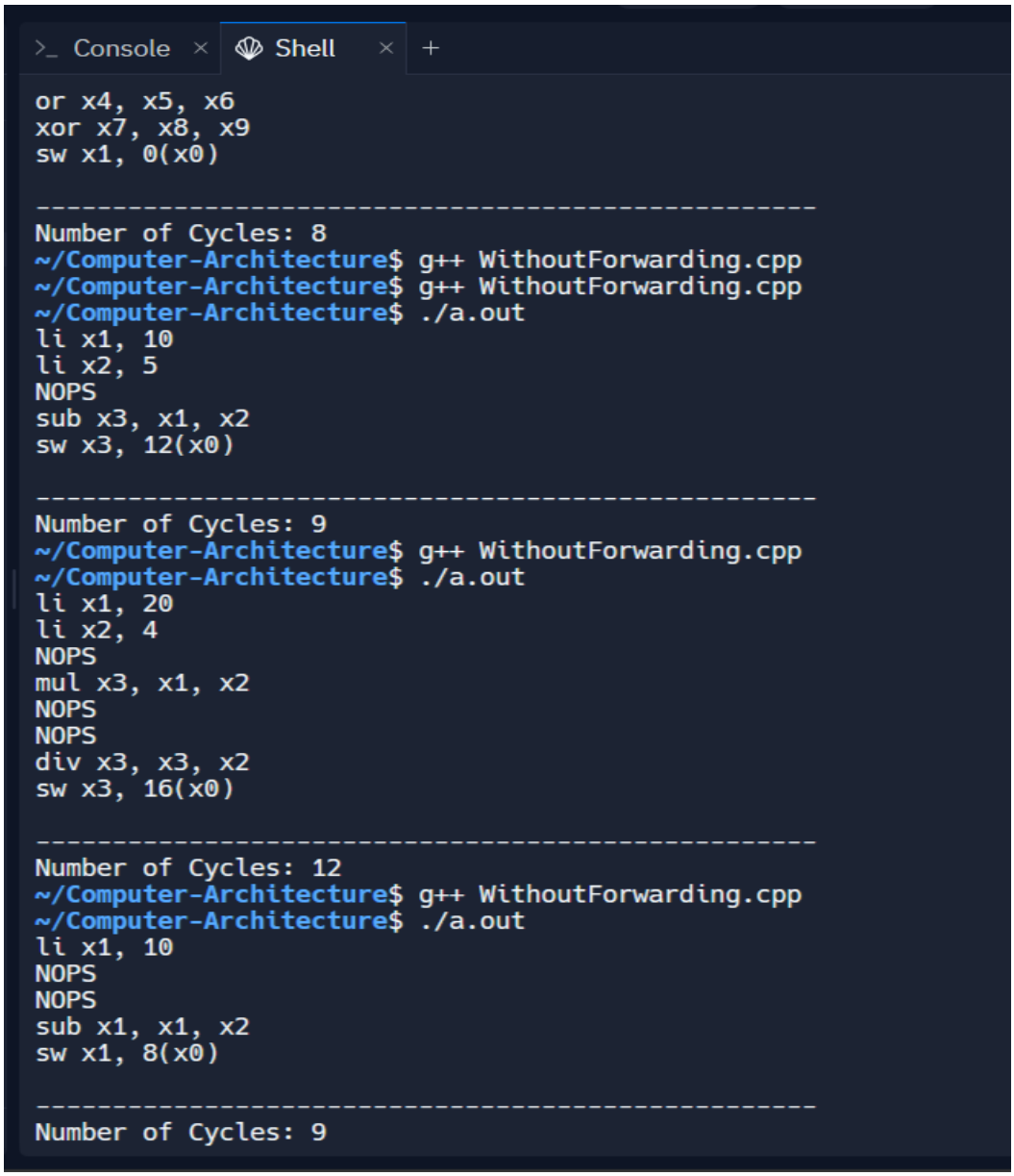
AMAN BHASIN
es21btech11006

Approach

1. Main Idea is to maintained the state of time when the current register will get free.
2. In case of any dependencies (hazards), i.e. when the previous stored value exceed the current estimated time of free, we have to insert the NOPS and update the free state of all the register in the current instructions (dest, src1 and src2(if needed))
3. Total time or Number of cycles is basically the time when destination register of last instruction will get free. Can be calculated by the formula: (no. of nops) + (no. of instructions) + (pipeline stages) - 1.
4. Each instruction is fetched and processed from the instruction object list, which is created by parsing the instruction file.
5. In case where forwarding is allowed, the estimate calculate time of getting free gets decreased as the concept of forwarding says, transfer the data after execution and update(write back) later.

Running

1. For testing the program, I have run it on various test cases, the file of test cases is attached in the folder.
2. results observed are below:



3. I tested the codes by the normal conventional approach of drawing the pipelined execution and finding the where to place NOPS