Data and Control Hazard Outline

ELITE SOURCE: [Pipeline Hazards]

Type of Hazards	Description of Hazard	Stall or Forward or Flush	Description of Solution
Control	This hazard is caused by a branch instruction being used in the hardware pipeline. The processor has yet to determine if the branch will be taken. So it has two cycles in the pipeline to fill while waiting for the branch to hit the execution stage. (Branch)	Flush	So we will fetch the next instruction in the program order so PC+4. Then the next instruction we will load in will be the instruction following the branch address. This way, we only have to flush one instruction. While it means we will always flush an instruction.
Control	Any Jump instruction will cause this hazard. The processor only knows the jump address once it gets calculated and returned in the execute stage. (Jump)	Stall, Stall	So, we will stall for 2 cycles so that jump, jump link, or jump return can be calculated properly in the execute stage. We need to disable updating the pc during the two NOP cycles.
Data	A data hazard with a distance of three aka. An instruction is writing back to a register currently being read in the decode stage.	None	So, we will modify the reg file to output the new write data to either R1 or R2 if the write address is the same as RS or RT.
Data	A data hazard with a distance of two aka. An instruction is in memory while an instruction is being decoded.	Forward	So we are going to forward from writing back to the ALU input. We are waiting for a cycle so that if it is a load word instruction it has time to get the data from memory. We need to ensure we forward to the correct ALU input, either iA or iB, whether or not it was RS or RT.
Data	A data hazard with a distance of one that is not a load word instruction.	Forward	So, we will forward from the memory address input back to the input of the ALU. This has the same thing to consider as the previous hazard.
Data	A data hazard with a distance of one is a load word instruction.	Stall, Forward	So, we will stop the entire pipeline in its tracks and add a nop between the loaded word

			and the other instructions to collect the data from memory. Then, it can be forwarded once the load word instruction is in write back.