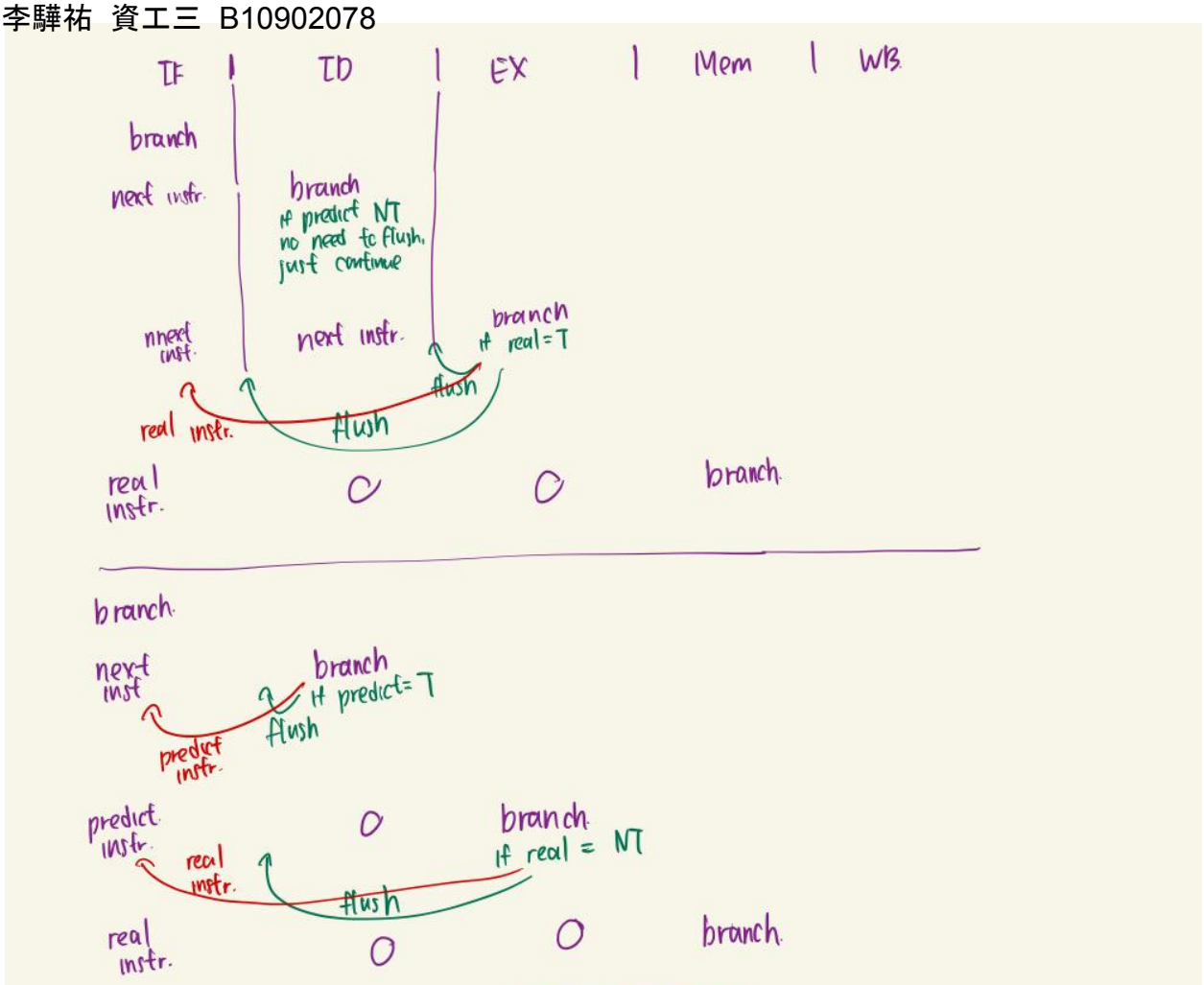
李驊祐 資工三 B10902078



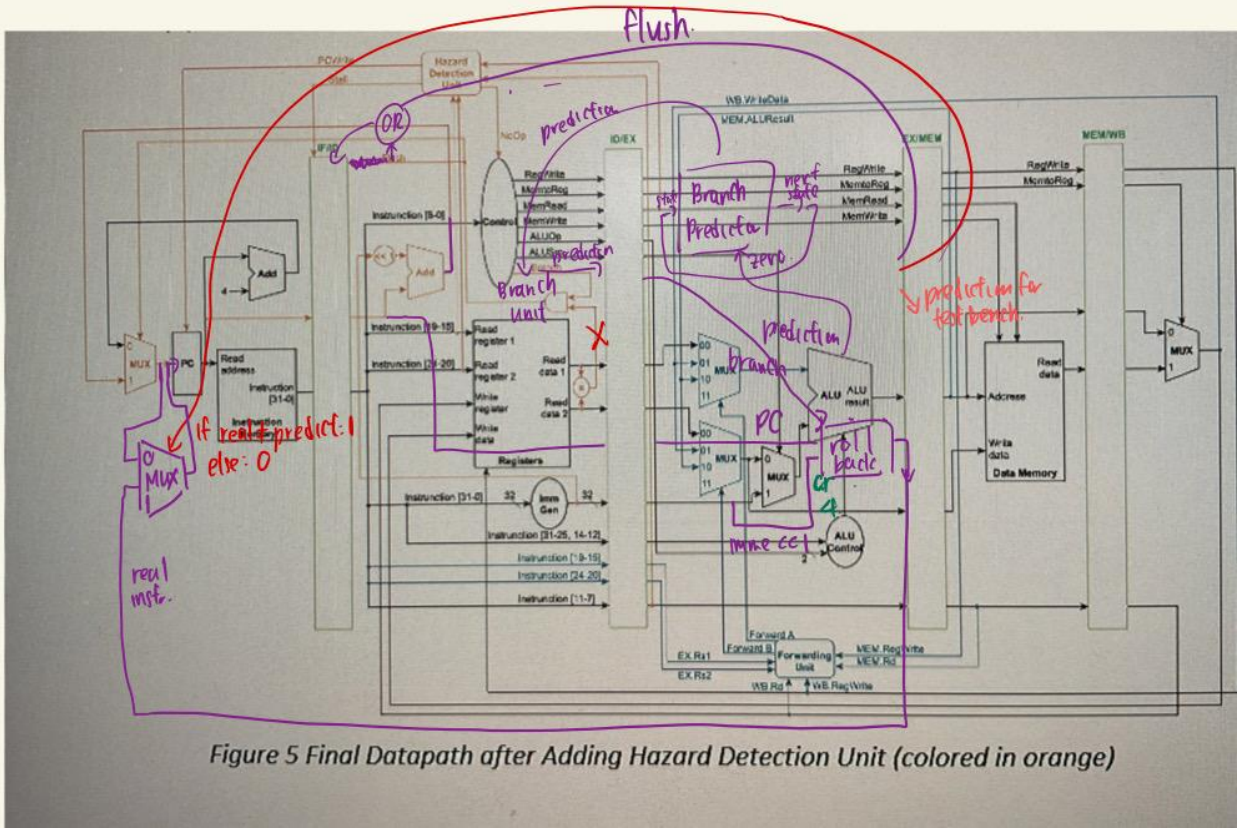


Figure 5 Final Datapath after Adding Hazard Detection Unit (colored in orange)

Modules Explanation:

From the image above, it can be seen that when the prediction from the ID stage and the real calculation in the EX stage is opposite, that means we need to flush both the IF_ID_register and ID_EX_register, and roll back. Hence, I add a rollback, a branch_predictor, and an extra MUX module.

Branch_predictor.v:
This module accepts zero_in, branch_in as input and has predict_o as output. The branch_in is the branch signal coming out from the ID_EX_pipeline, and zero is the signal from ALU. It also has a branch state reg. If the branch_state is either ST or T, then predict_o is T. If it is either NT or SNT, then predict_o is NT. At every positive clock edge, it will update the branch_state according to branch_in and zero_in.

Rollback.v:
This module has branch_in, zero_in, branch_prediction_in, pc_in, and imme32_in as input, and has IF_ID_flush, ID_EX_flush, mux_select, next_pc as output. The branch_prediction is the signal coming out from ID_EX_pipeline (which is the previous branch prediction in the ID stage). If branch_prediction_in is not equal to zero_in, that means the prediction in ID stage and real calculation in EX stage is opposite, so IF_ID_flush, ID_EX_flush, and mux_select is set to 1, otherwise 0.

If branch_in is 1 and zero_in is 1 and branch_prediction is 0, that means the previous prediction is NT but real prediction is T, so next_pc is set to pc + (imm32 << 1). If branch_in is 1 an dzero_in is 0 and branch_prediction is 1, that means the previous prediction is T but real prediction is NT, so next_pc is set to pc + 4.

ID_EX:
In this module, I add an extra branch_prediction_in and branch_prediction_out wire. This is because I need to send the previous prediction in ID stage to Rollback so that Rollback can check whether the real calculation is EX stage is different from the previous prediction in ID stage or no.

Extra_mux:
I add an extra mux before PC so that when mux_select of rollback is 1, it will select the rollback address. Otherwise, it will just select the address from the leftmost MUX.

Branch_unit:
In this module, it checks the branch_prediction of branch_predictor as input. When branch_prediction is 1 and it is an branch instruction in ID stage, it will set flush to 1.

IF_ID_flush:
In this module, I adjust the flush signal to become the output of flush from branch_unit OR flush from rollback.

Difficulties:
This is the hardest lab for me, because there are no diagrams provided and I have to draw it myself. I spend a lot of time debugging because the timing to update branch_state and branch_prediction is very complicated. Also, I made a very stupid mistake because I randomly named the wires at first. When I made modifications, I forgot the delete the wires at the bottom of the code because it was too long, and I spent two whole days debugging just to find out this stupid mistake.

Environment:
Ubuntu, Ubuntu 22.04.1 LTS (GNU/Linux 5.10.16.3-microsoft-standard-WSL2 x86_64).