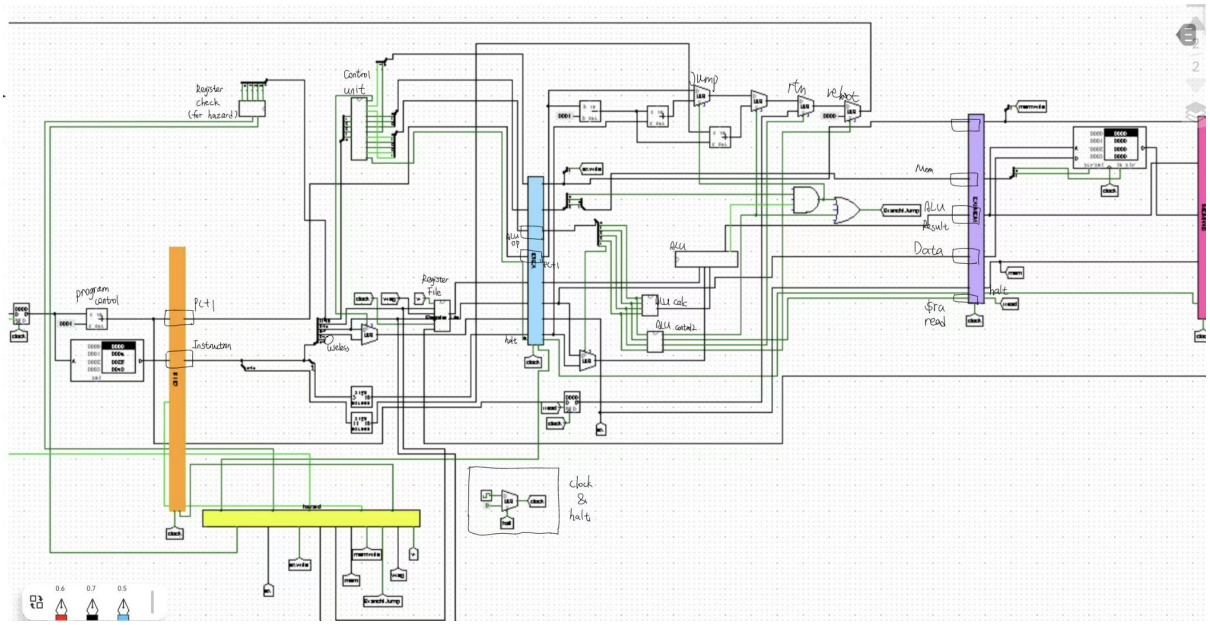
Homework 4 Answer Sheet for the Bonus Question

Please state the name and SID of all members of your group.

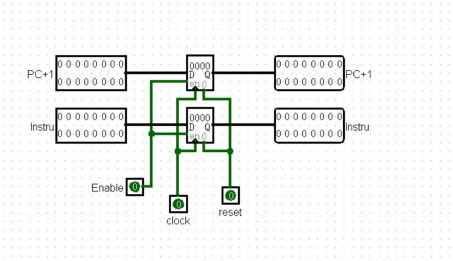
member	name	SID	email
#1 (contact person)	LIU Hengche	57854329	hengchliu2-c@my.cityu.edu.hk
#2	CHEN Yihuan	57853935	yihuachen6-c@my.cityu.edu.hk
#3	XIA Shujun	57854422	shujunxia2-c@my.cityu.edu.hk

A. Please graphically explain the part of circuits included in each step of the pipeline.



Step1: Instruction Fetch

In the IF stage, the PC+1 and instruction would be stored in the register of the pipeline IF/ID, and when we have a branch or jump command, we will flush IF/ID pipeline (Resetting the registers)



Step2: Instruction Decode & Register File Read

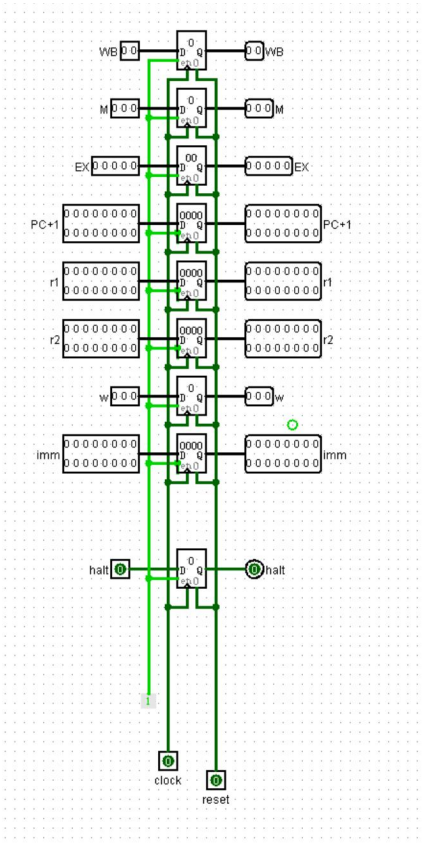
1. We get the OP codes from the instructions and store the generated corresponding outputs in the Control Unit in the ID/EX pipeline.

2. WB(write back): Bit 0 is RegWrite, bit 1 is MemToReg;

3. M(Main Commands): Bit 0 is branch, bit 1 is MemRead and bit 2 is MemWrite;

4. PC+1 stores the programme control; r1, r2 stores the read data from the register file;

5. w stores the destination register to write (Rt or Rd) (for the future possible stall); Imm stores the extended 0~4 bits of immediate; and halt stores the halt signal.



Step3: Execute and Programme Control

In this step,

1. WB still stores the write back signals;

2. After executing in the ALU we have the result of BLE, together with the “Branch” signal we determine if we branch or not through the AND gate; in the logic gate of ALU control2, we determine whether the command is jump or not using the ALUops; connect both through OR gate, we send this back to the “Hazard” section to determine if we flush or not;

3. The ALU result stores the result of ALU;

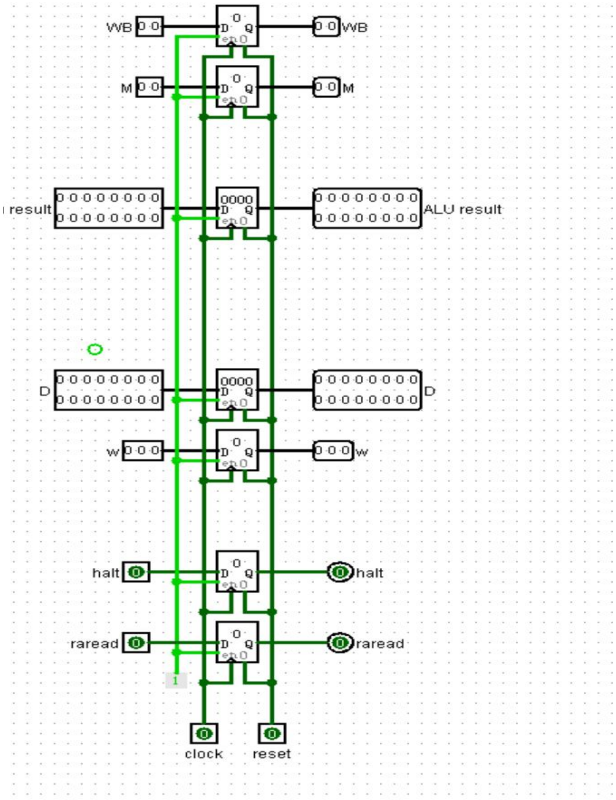
4. And data stores the data to write in the memory if needed;

5. W stores the destination;

6. Halt stores the halt signal;

7. Raread (Return Address Read) stores the signal of the “rtn” instruction after the calculation of ALUops in the ALU control2;

8. The multiplexers determine the branch and jump details of the next programme control. They are controlled by the calculations of ALUops in the ALU controls.



Step4: Memory Access

1. WB stores the write back signals

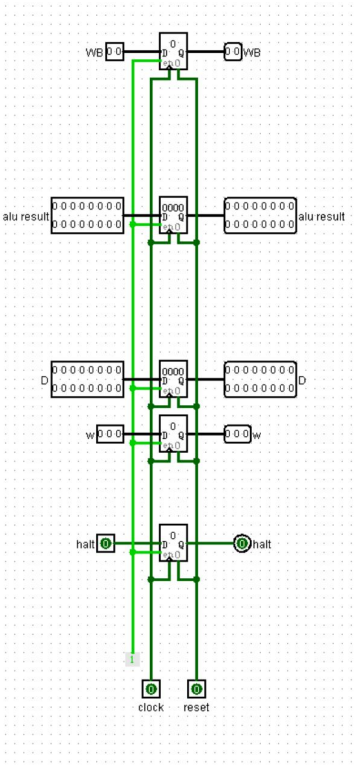
2. ALU result stores the result of ALU

3. Data stores the data to be written in the memory if needed.

4. W stores the register destination

5. Halt stores the halt signal.

6. In this step, the main commands are split into MemRead and MemWrite to determine to load or store in the memory.

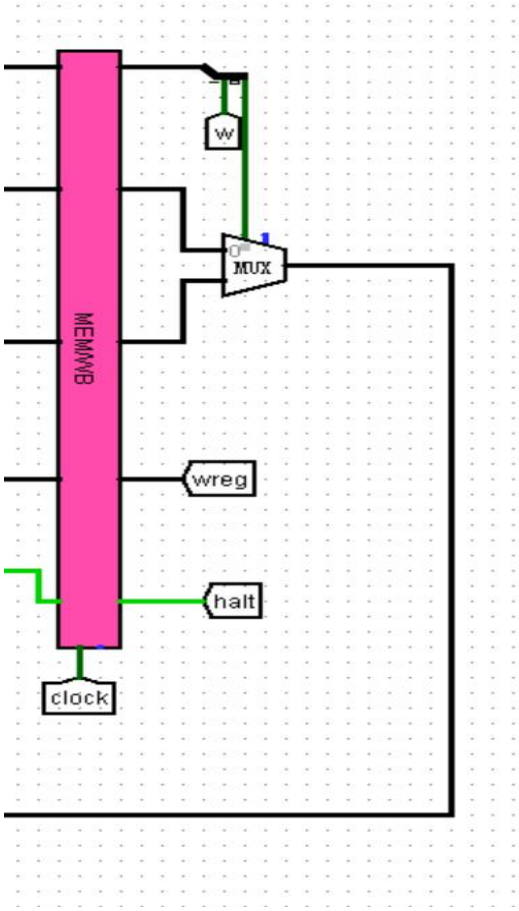


Step5: Write Back

1. In this step WB will split the branch of MemToReg to determine the output of the multiplexer is ALU result or Data read from the RAM.

2. The wreg(write register) will send to the hazard to see if stall is needed.

3. Finally halt could be used and it is connected to the multiplexer of the clock.

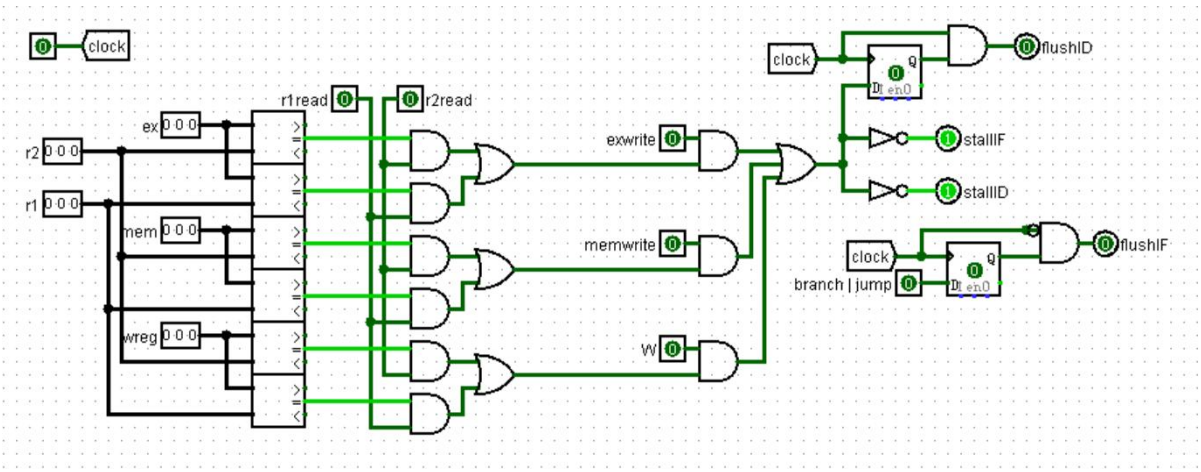


B. Please explain how does your processor deal with different types of hazards.

The hazards comes from two situations:
1. Register needed is being used (halt and flush)
2. Branch or Jump instructions (flush)

To solve the first hazard, we stall the ID stage and IF stage until the other progress requiring the register is finished. For example, in MemWrite, if the register needed(r1 and r2) is the same as Mem, we'll need to stall IF and stall ID.

For the second hazard, if branch or jump is encountered, we will flush all the registers in the IF pipeline to prepare for the fetch of new instructions



C. Fill the following table with the machine codes of each instruction of the testing program:

Test program 5:

instruction	machine code (binary)	machine code (hex)
li $r1, 10	0000 0000 0000 1010	000A
li $r2, 8	0000 0000 0010 1000	0028
li $r3, 0	0000 0000 0100 0000	0040
store $r1, $r3	0011 0010 0000 0000	3200
load $r4, $r3	0010 1010 0110 0000	2A60
add $r5, $r4, $r2	0000 1011 0011 0000	0B30
ble $r2, $r4, 2	0101 1011 0010 0010	5B22
and $r6, $r4, $r2	0001 0011 0011 0100	1334
halt	1001 1000 0000 0000	9800