Kuwait University

College of Engineering and Petroleum

Computer Engineering Department

CpE-469-Computer Architecture Laboratory

Semester: Spring

Section No. 03A

Lab Assignment #2

Student name:

Asmaa Adel Alazmi - 2201122708

Instructor Name: Dr. Imtiaz Ahmad

TA Name: Eng. Maryam Aljame

Date: 10/April/2025

(i) Introduction & Objective

The purpose of this assignment is to simulate and test the RISC-V pipelined data path and the forwarding and the hazard detection unit that was implemented in the previous lab sessions. In addition, this assessment will cover the modification of the forwarding and hazard detection unit to support more forwarding and stall cases.

(ii) Tested Instructions

The following table showcases the instructions used in testing the new pipeline with the corresponding instruction code in binary.

Address	Instruction	Instruction Code in Binary (Encoding)	Hazard Type
0	ADDI x1, x3, 2	0010 011 00 001 0010	
1	ADD x2, x1, x3	0 011 001 00 010 0001	Addi, subsequent R-Type
2	BEQ x1, x0, 0	0 000 001 00 000 0101	Addi, subsequent BEQ
3	ADD x5, x2, x2	0 010 010 00 101 0001	R-Type, subsequent R-Type
4	ADD x6, x5, x0	0 101 000 00 110 0001	R-Type, subsequent R-Type
5	LW x1, imm(x0)	0 000 000 00 001 0011	
6	ADD x7, x1, x2	0 010 001 00 111 0001	LW, subsequent R-Type
7	ADD x7, x2, x0	0 000 010 00 111 0001	R-Type, subsequent R-Type
8	JAL x7, 0	00000000 111 0110	
9	ADD x1, x7, x6	0 110 111 00 001 0001	JAL, subsequent R-Type

(iii) Results

1. Theoretical results

- In theory, instruction 9 should not be executed since the program jumps to address 0 before it. All the values in the registers are initialized as zero including the imm.

Addrs	Instruction	Changes in Registers (decimal values)	Jump	Data Hazard
0	ADDI x1, x3, 2	X1 = 2		
1	ADD x2, x1, x3	X2 = 2		X1
2	BEQ x1, x0, 0		False	X1
3	ADD x5, x2, x2	X5 = 4		X2
4	ADD x6, x5, x0	X6 = 4		X5
5	LW x1, imm(x0)	Mem[0] = 0		
6	ADD x7, x1, x2	X7 = 4		X1
7	ADD x7, x2, x0	X7 = 2		X7
8	JAL x6, 0	X6 = 9	True	
9	ADD x1, x6, x6	X1 = 18		X6

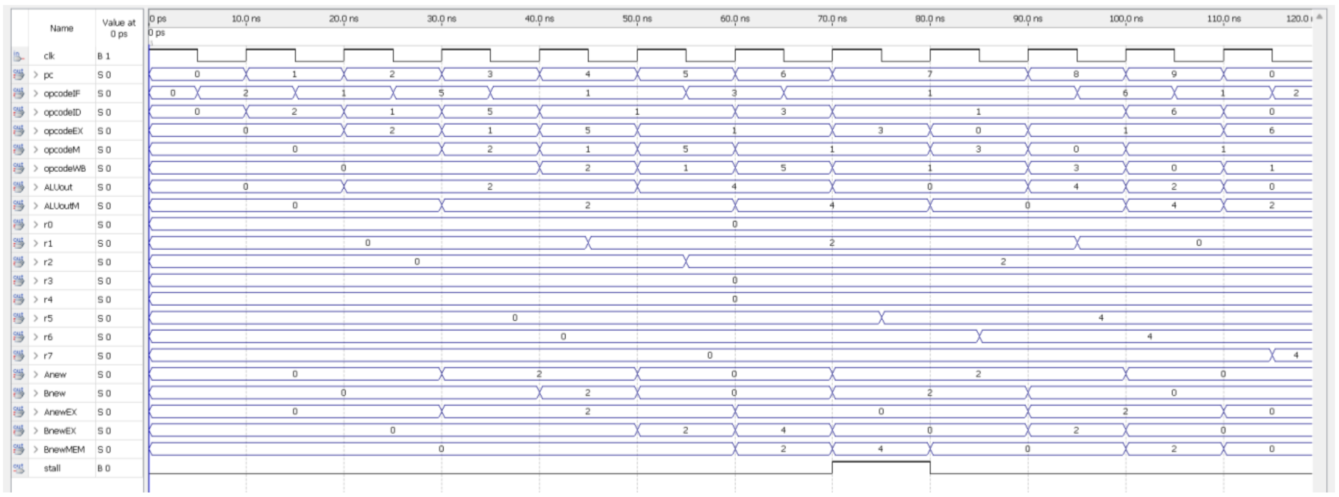
2. Pipeline

- A total of 10 instructions were used, 7 of which had data hazards and a single stall. The architecture we are using in this implementation only allows for memory stage and write back stage forwarding which limits some of the forwarding being executed.

Instruction		Clock Cycle														
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	ADDI x1, x3, 2	**IF**	**ID**	**EX**	**M**	**WB**										
1	ADD x2, x1, x3		**IF**	**ID**	**EX** ↓	**M**	**WB**									
2	BEQ x1, x0, 0			**IF**	**ID** ↓	**EX**	**M**	**WB**								
3	ADD x5, x2, x2				**IF**	**ID** ↓	**EX**	**M**	**WB**							
4	ADD x6, x5, x0					**IF**	**ID**	**EX** ↓	**M**	**WB**						
5	LW x1, imm(x0)						**IF**	**ID**	**EX**	**M**	**WB**					
6	ADD x7, x1, x2							**IF**	☁	**ID** ↓	**EX**	**M**	**WB**			
7	ADD x7, x2, x2									**IF**	**ID**	**EX** ↓	**M**	**WB**		
8	JAL x6, 0										**IF**	**ID**	**EX**	**M**	**WB**	
9	ADD x1, x6, x6											**IF**	**ID**	**EX** ↓	**M**	**WB**

3. Waveform results

- The result of the waveform matches the theoretical table we configured. Alas, the results for the JAL (jump and link) instruction can be seen right after the execution part is done (110-120ns) which is the total time needed to loop through our code.
- The outputs ALUout & ALUoutM are used for debugging, and validation.



(iv) Conclusion

In conclusion, this assignment paved the way for a better understanding on how data hazards affect a pipelined RISC-V processor. Improving the forwarding and hazard detection units to handle cases involving ADDI instruction, allowing for more accurate and efficient instruction execution. Writing our own RISC-V code to test these changes made the concepts more concrete and showed how important good hazard handling is for keeping things running smoothly without unnecessary delays. It was a great step toward building more reliable and efficient CPU designs.

Further improvements and modifications are needed to reach the full potential of this mini processor.

(v) References

- Kuwait University, CpE-469 Computer Architecture Laboratory, Lab Manual

(vi) Resources

- The waveform for this project is labeled **lab4.wvf**
- The full **FHD.v** code which includes the bigger part of hazard detection is commented out. Only the newly added modifications are running.