Faculty of Media Engineering and Technology
Dept. of Computer Science and Engineering
Dr. Milad Ghantous

CSEN 702: Microprocessors
Winter 2024

Practice assignment 2-Solution

Exercise 1

Consider the following code:

```
Loop: LD R1,0(R2)          #load R1 from address 0+R2

DADDI R1, R1, 1            #R1=R1+1

SD R1, 0(R2)                       #store R1 at address 0+R2

DADDI R2, R2, 4            #R2=R2+4

DSUB R4,     R3,     R2            #R4=R3-R2

BNEZ R4,     Loop          #branch to Loop if R4!=0
```

Assume that the initial value of R3 is R2 + 396.

A) List all data dependencies, regardless if they cause a hazard or not.
R1 in DADDI depends on R1 from LD before

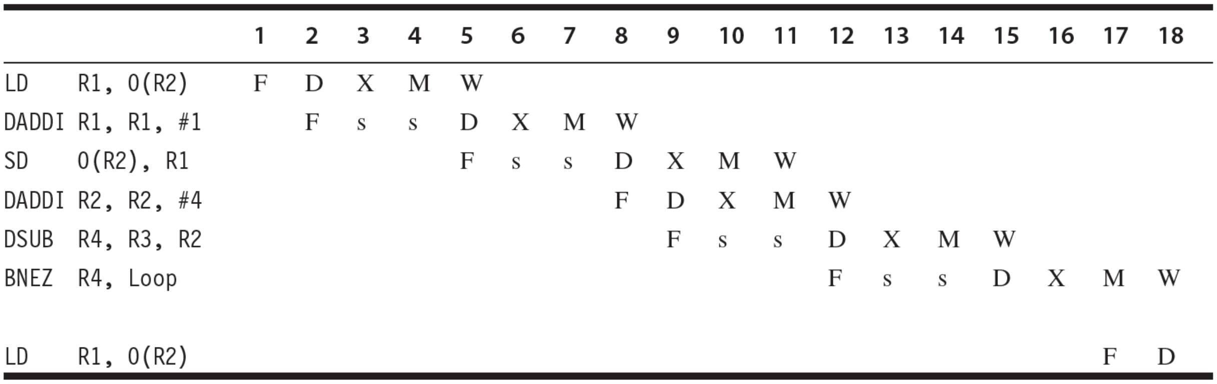
R1 in SD depends on R1 in DADDI

R2 in DSUB depends on R2 in DADDI before

R4 in BNEZ depends on DSUB

B) Show the timing of this instruction sequence for the 5-stage RISC pipeline without any forwarding or bypassing hardware but assuming that a register read and a write can happen in the same clock cycle as mentioned in the lecture. Assume branches are handled by stalling the pipeline until the outcome is known. Branch outcomes are known at the end of EX stage.

Solution:



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
`LD    R1, 0(R2)`	F	D	X	M	W													
`DADDI R1, R1, #1`		F	s	s	D	X	M	W										
`SD    0(R2), R1`					F	s	s	D	X	M	W							
`DADDI R2, R2, #4`								F	D	X	M	W						
`DSUB  R4, R3, R2`									F	s	s	D	X	M	W			
`BNEZ  R4, Loop`												F	s	s	D	X	M	W
`LD    R1, 0(R2)`																	F	D

Stall as soon as we detect dependency.

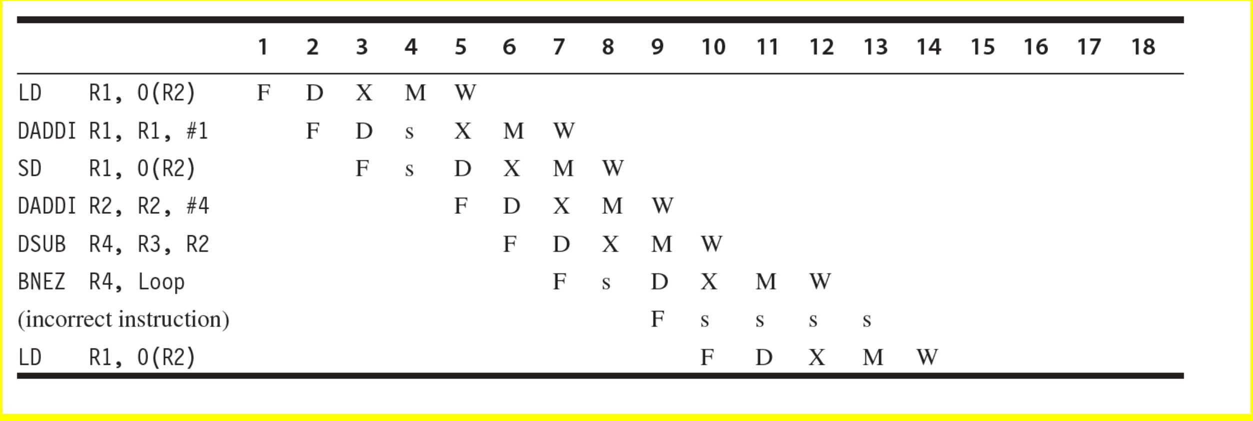
Two stalls are needed each time, not 3, since in the same cycle (W) or Write back, we can perform read and write so no need to wait.

Also, branch outcomes are known at the end of execute stage.

C) Show the timing of this instruction sequence for the 5-stage RISC pipeline with full forwarding and bypassing hardware. Assume that the branch is handled by predicting it as **not taken**. Branch outcomes are known at the end of the Decode stage.

Solution:

We are using forwarding and branch outcomes are known at the end of **decode** stage.



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
LD R1, 0(R2)	F	D	X	M	W													
DADDI R1, R1, #1		F	D	s	X	M	W											
SD R1, 0(R2)			F	s	D	X	M	W										
DADDI R2, R2, #4					F	D	X	M	W									
DSUB R4, R3, R2						F	D	X	M	W								
BNEZ R4, Loop							F	s	D	X	M	W						
(incorrect instruction)									F	s	s	s	s					
LD R1, 0(R2)										F	D	X	M	W				

The stall in DADDI is needed because R1 is not available until after M stage of LD

The stall in BNEZ is needed because R4 value from the ALU is not ready yet to be used in the decode stage of BNEZ so we delay the decode stage of BNEZ till after the EX stage of DSUB

Since in the D stage we know the outcome, only one incorrect instruction is fetched after branch before the loop loops again.

D) Show the timing of this instruction sequence for the 5-stage RISC pipeline with full forwarding and bypassing hardware. Assume that the branch is handled by predicting it as **taken**. Branch outcomes are known at the end of the Decode stage.

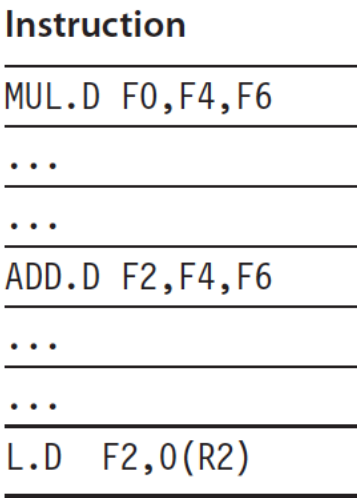
Solution:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
LD R1, 0(R2)	F	D	X	M	W													
DADDI R1, R1, #1		F	D	s	X	M	W											
SD R1, 0(R2)			F	s	D	X	M	W										
DADDI R2, R2, #4					F	D	X	M	W									
DSUB R4, R3, R2						F	D	X	M	W								
BNEZ R4, Loop							F	s	D	X	M	W						
LD R1, 0(R2)									F	D	X	M	W					

Cycle 10 is ok since only one them is trying to use the memory
But cycle 11 is a problem. All 3 instructions are trying to write back to the register file.

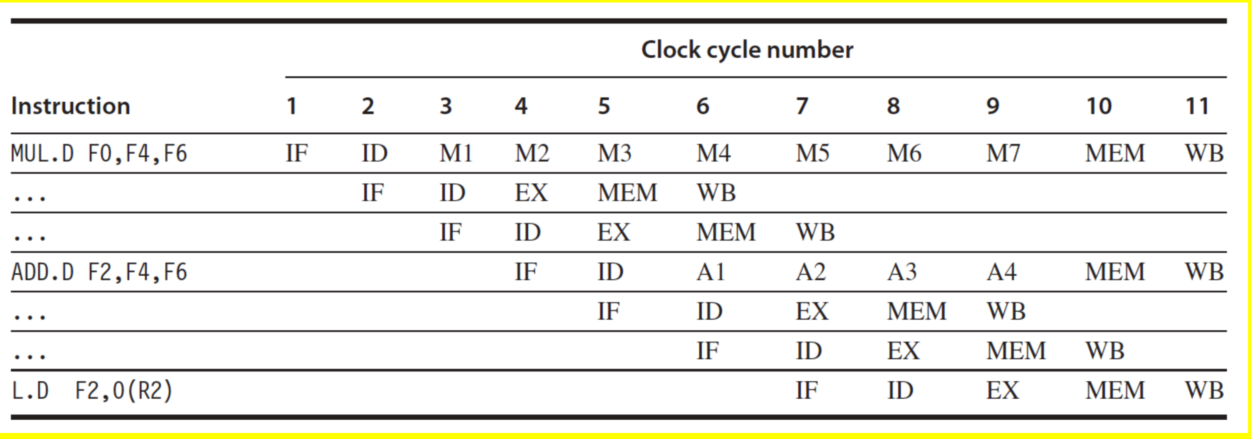
Exercise 2

Complete the timing sequence of this code.
Each (…) represents a normal integer instruction with no dependency. Notice the floating point operations.



Instruction
MUL.D F0,F4,F6
...
...
ADD.D F2,F4,F6
...
...
L.D F2,0(R2)

solution:



	Clock cycle number										
Instruction	1	2	3	4	5	6	7	8	9	10	11
MUL.D F0,F4,F6	IF	ID	M1	M2	M3	M4	M5	M6	M7	MEM	WB
...		IF	ID	EX	MEM	WB					
...			IF	ID	EX	MEM	WB				
ADD.D F2,F4,F6				IF	ID	A1	A2	A3	A4	MEM	WB
...					IF	ID	EX	MEM	WB		
...						IF	ID	EX	MEM	WB	
L.D F2,0(R2)							IF	ID	EX	MEM	WB

Cycle 10 is ok since only one them is trying to use the memory
But cycle 11 is a problem. All 3 instructions are trying to write back to the register file.

Exercise 3

A microprocessor is designed to have an adjustable voltage. It was observed that reducing the voltage to 80% would result in a 25% reduction of the frequency. Calculate the impact of this reduction on power and energy.

Solution:

Impact on Energy:

Energy_{new} / Energy_{old} = (1/2 * Cap. Load * Voltage^2_{new}) / (1/2 * Cap. Load * Voltage^2_{old}).
Since load doesn't change, then
Energy_{new} / Energy_{old} = (0.8 * V_{old})/ (V_{old}) = 0.8^2 = 0.64
□New energy is 36% less

Impact on power:

Power new / Power old =
(1/2 * Cap. Load * Voltage^2_{new} * Frequency_{new}) / (1/2 * Cap. Load * Voltage^2_{old} * Frequency_{old}).
= 0.64 * (0.75* Frequency_{old})/(Frequency_{old}) = 0.64*0.75 = 0.48
□ *Power is almost divided by half. (52% less)*

Exercise 4

Suppose we have made the following measurements.

- Frequency of FP operations = 20%
- Average CPI of all FP operations = 5.0
- Average CPI of other instructions = 1.5.
- Frequency of FP division is 4% out of the entire program, with a CPI of 14.

A) Compute the average CPI.
B) Consider two enhancements E1 and E2, where in E1, we try to minimize the CPI of FP divisions to 4, while in E2, we try to minimize the overall FP CPI to 2.5. Compare the impact of both enhancements on the performance.

Solution:

A) average CPI = 0.2*5 + 0.8 * 1.5 = 2.2 [because 20% FP, 80% others]

B) Computations for E1: minimize CPI of division down to 4, from original 14, hence saving 10 cycles per division. Divisions occur 4% of the time.
Enhanced CPI using E1 = CPI original – 4% * (cycles saved) = 2.2 – 0.04*(10) = 1.8

Computations for E2: minimize overall FP CPI to 2.5
 new enhanced CPI using E2 = 0.2*2.5 + 0.8 * 1.5 = 0.5 + 1.2 = 1.7

Verdict: E2 is slightly better than E1 since it resulted in a lower overall CPI.

Exercise 5

Consider the following benchmark execution times for processors A and B, against the reference processor R.

Benchmark	**R**	**A**	**B**
wupwise	1600	45	55
swim	900	40	20

A) Compare the performance of processors A and B, using the geometric mean.
B) Discuss the result.

Solution:
A) let’s compute the SPECRatios of both computers.

Benchmark	**R**	**A**	**SPECRatio A**	**B**	**SPECRatio B**
wupwise	1600	45	35.5	55	29.09
swim	900	40	22.5	20	45

Geometric mean A = (35.5 x 22.5) $^{(1/2)}$ = 28.26
Geometric mean B = (29.09 x 45) $^{(1/2)}$ = 36.1

B) Processor B has a higher performance since its mean is higher than A. however, having only 2 samples in the benchmark is not a definite indicator that B is overall better. More samples are needed.

Exercise 6 [optional]

Consider the following prices and measures:

- Cost of wafer: 50$
- A wafer can be divided into 1200 dies. 34% of them are usually defective on average.
- Testing costs 4$ and packaging cost an extra 3$
- A final yield test is found to be 97%.

Calculate the cost of the integrated circuit.

Solution:

Cost of die = cost of wafer / (dies per wafer x die yield) = 50 / (1200 * 0.66) = 0.063 $

Cost of integrated circuit = (cost of die + testing + packaging) / final yield
= (0.063 + 4 + 3)/0.97 = 7.28$