

Aidan Lopez

Assignment: Processing Pipelining

Instructor: Elaheh Sadredini

elaheh@cs.ucr.edu

University of California, Riverside

Grading (total 4.2 points)

- Question 1: 0.3 points
- Question 2: 0.3 points
- Question 3: 0.3 points
- Question 4: 1.5 points
 - Part A (0.6), part B (0.6), part C (0.3)
- Question 5: 1 point
 - Part A (0.2), part B (0.3), part C (0.3), part D (0.2)
- Question 6: 0.8 points

Question 1: Single-cycle processor clock

Question 1: Single-cycle processor clock

A processor has the following mix of instruction types, which require different amounts of time to complete each instruction.

What is the fastest clock frequency this single-cycle processor can use?

Instruction	Time	% of Instructions
Type 1	800ps	30%
Type 2	200ps	20%
Type 3	250ps	50%

Answer

- Briefly explain.

The fastest clock frequency is $1/200\text{ps}$. Even though there are different percentages of instructions, each instruction type can still be ran on their own. Because of this, if only instructions of type 2, which have the fastest time, are run, the fastest clock frequency will be $1/200\text{ps}$.

Question 2: Single-cycle processors efficiency

A processor has the following mix of

Question 2: Single-cycle processors efficiency

A processor has the following mix of instruction types, which require different amounts of time to complete each instruction.

What % of the time is wasted when executing 10 instructions?

Instruction	Time	% of Instructions
Type 1	800ps	30%
Type 2	200ps	20%
Type 3	250ps	50%

Answer

- Provide a brief explanation.

The percent of time wasted is $50\% * \frac{4}{5} + 30\% * \frac{1}{4} = 47.5\%$. Since type 2 is the fastest, type 3 is 1.25 times slower than type 2 and makes up 50% of the instructions, and type 1 is 4 times slower than type 2 and makes up 30% of the instructions. Since there are 10 instructions we can assume there are 3 type 1 instructions running, 2 type 2 instructions, and 5 type 3 instructions due to the percentages.

Question 3

- We have a processor that has a 200ns clock frequency and execute instructions

Question 3

- We have a processor that has a 200ns clock frequency and execute instructions sequentially (i.e., no pipeline). If we design a 25-stage pipeline (assume delay of all stages are equal, i.e., $200/25 = 8\text{ns}$) with pipeline registers that take 2ns each, how much speedup do we get? Assume the pipeline is full.

Answer

- Provide a brief explanation.

The amount of speedup is 10x. Since the pipeline is full we do not have to wait for it to fill, thus getting the most speedup out of it. The speedup is then the $200\text{ns}/(8\text{ns delay} + 2\text{ns register}) = 10$.

Question 4: Pipeline overheads

A processor takes **100ns** and **100pJ** for each instruction. The processor can be infinitely pipelined with each pipeline register taking **2ns** and **2pJ**. What is the **throughput**

Question 4: Pipeline overheads

A processor takes **100ns** and **100pJ** for each instruction. The processor can be infinitely pipelined with each pipeline register taking **2ns** and **2pJ**. What is the **throughput** (average time per instruction), **latency** (time to complete each instruction), and **energy per instruction** for a **100-stage** processor compared to the original processor?

- A. Original:
 - **Latency:**
 - **Throughput:**
 - **Energy:**
- B. Pipelined:
 - **Latency:**
 - **Throughput:**
 - **Energy:**
- C. Is this a good design? Which design do you choose for a server? Which design do you choose for battery-operated mobile devices? Why?

Answer

- Briefly discuss part C
 - A. Latency: 100ns
Throughput: 1 instruction/100ns
Energy: 100pJ
 - B. Latency: 100ns
Throughput: 1 instruction/2ns
Energy: 100pJ
 - C. This is a good design because it does increase the average time per instruction.

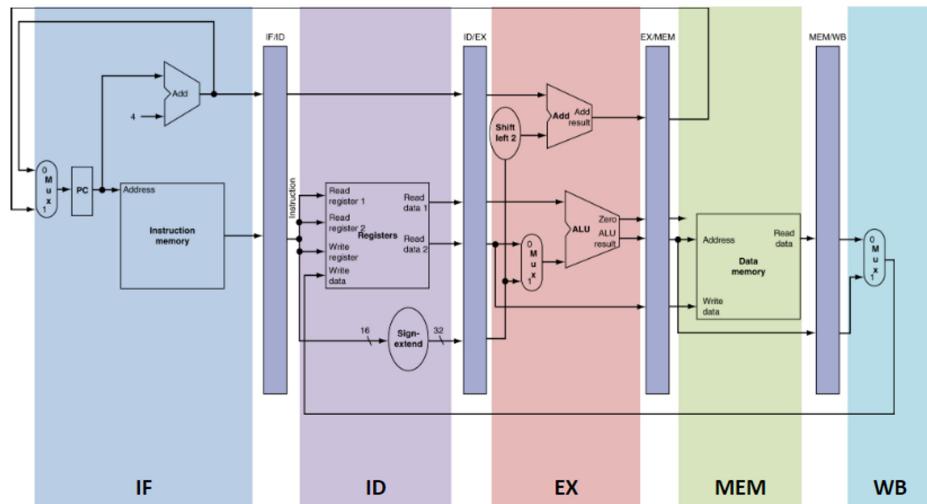
Question 5: Basic pipeline branch determination

- A. In which stage is the branch instruction when its result is applied?

Question 5: Basic pipeline branch determination

A. In which stage is the branch instruction when its result is applied?

1. IF
2. ID
3. EX
4. MEM
5. WB



Question 5: Basic pipeline branch determination

B. Fill in the pipeline for the following instructions:

A, B, beq G, C, D, E, F, G

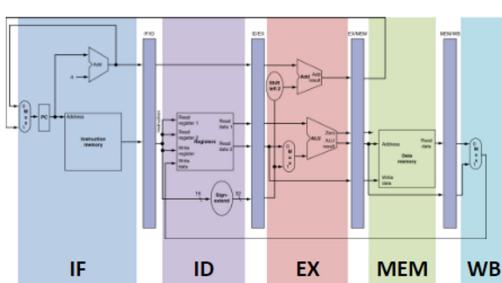
C. Assume **beq G** is taken. Does this program execute correctly?
Why?

Hint: beq G means jump to the instruction G

D. What do you need to do (and when, i.e., which cycle) to get the correct behavior?

Hint: What should happen to instructions C, D, E

Cycle	IF	ID	EX	MEM	WB
0	A				
1		A			
2			A		
3				A	
4					A
5					
6					
7					



Answer

- Answer Part A, B, C, D. Briefly explain your answer for each part.

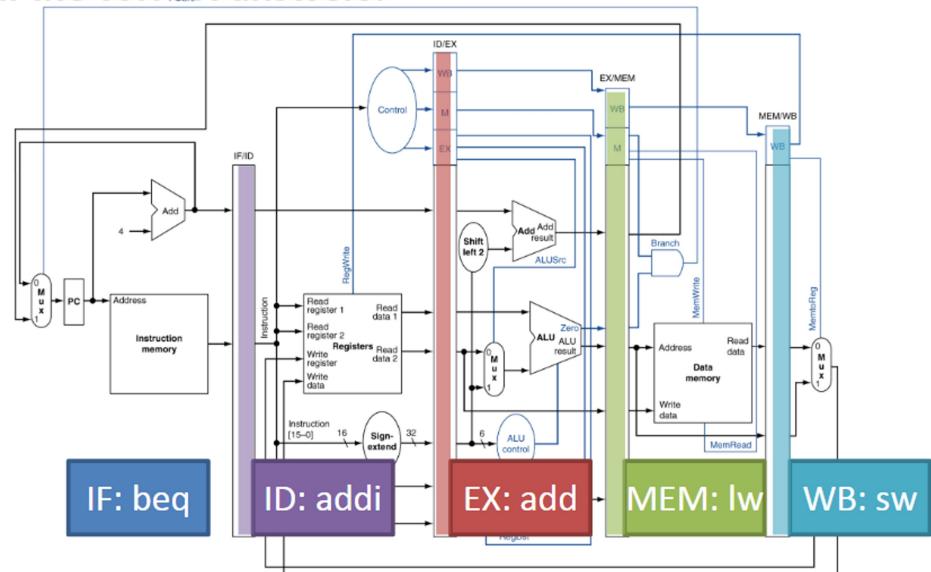
- It is in the EX stage since this is the final stage where the branch address is calculated and it does not have a MEM or WB stage since it doesn't use memory or write to the RF.
- See table.
- The program does not execute correctly since G has not been passed into the pipeline yet.
- To get the correct behavior, you would need to have the instruction G come before CDEF so that the address can be loaded into the pipeline before beq is executed in the pipeline

Cycle	0	1	2	3	4	5	6	7
	A	B	A	B	A	B	A	B
IF		beq						
ID			beq					
EX				beq				
MEM					beq			
WB						beq		

Question 6: Pipeline control

The following instructions are in the pipeline from newest to oldest: beq, addi, add, lw, sw. Which pipeline register(s) have RegWrite == true? **More than option can be correct. Select and explain all the correct answers.**

1. PC
2. IF/ID
3. ID/EX
4. EX/MEM
5. MEM/WB



Answer

- Briefly explain all selected answers.

The pipeline registers with RegWrite == true are IF/ID, ID/EX, and EX/MEM because IF has beq which regwrite is false, ID holds addi which regwrite is true, EX holds add which regwrite is true, MEM holds lw which regwrite is true, and WB holds sw which regwrite is false.