

Name: MALIK HAMMAD FAISAL

Roll No: 2021CS10559

(COL 216) Computer Architecture

March 26, 2023

Minor 2

Duration: 60 minutes

(30 marks)

**Beware:** Be concise in your writing. You can use rough sheets for calculations. But you cannot submit any additional sheet for grading on Gradescope. So make sure you are certain when you write something (after rough work, or use a dark pencil). If you cheat, you will surely get an F in this course.

1. A 5-stage MIPS pipeline has a 1 cycle **load-to-use** delay i.e. you have to wait for 1 cycle between a *lw* instruction and any ALU instruction that uses it, as you cannot forward from MEM where data is ready at the end of cycle to EX where data is needed at the beginning of cycle. The pipeline is otherwise ideal i.e. the pipeline is always full, there is no forwarding delay, and each instruction takes 1 clock cycle. A program has 20% loads, but the compiler can find independent instructions to put after the load only for half of them. What is the slowdown due to delay/NOP slots? [3 marks]

Out of 20% loads  $\frac{1}{2} \times 20\% = 10\%$  have independent instructions after them

So 10% loads result in 1 cycle delay of total instructions

Let total instructions be  $x$  each  $x$  cycle

$\frac{10}{100}x$  instructions add a delay

So total cycles =  $x + 0.1x = 1.1x$

CPI = 1.1

So ~~delay~~ is 1.1 times actual times

Slowdown = 0.1 cycles per instruction on average

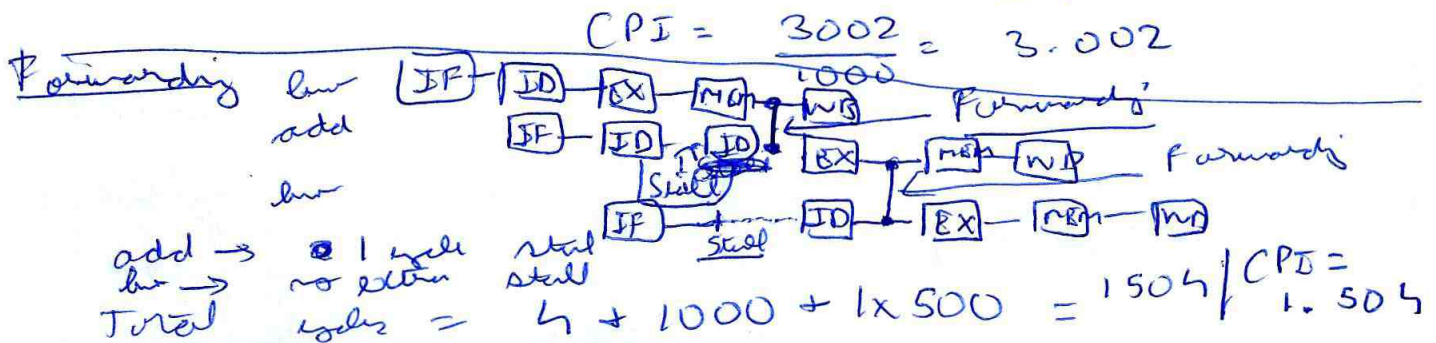
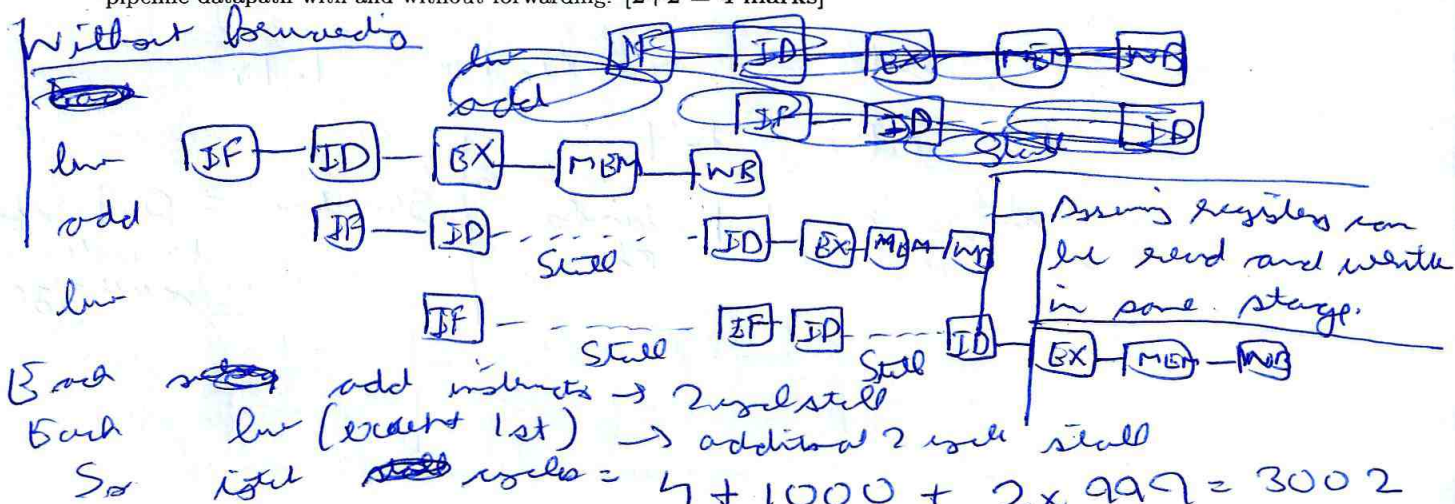
Name: MALIK NAWAB RAJAL

Roll No: 2021CS10559

2. Out of the 5 MIPS pipeline stages IF, ID, EX, MEM, W — which are never destinations for forwarding/bypassing? Why? [3 marks]

- IF :- because in this stage we do not know what registers are going to be used
- ID :- We know registers to be read only after execution of this stage
- W :- This is last stage so when an instruction is in this stage, previous instructions needed have completed.

3. Consider a 1000 instruction program of alternating *lw* and *add* instructions: *lw*, *add*, *lw*, *add* etc. The *add* instruction depends (and only depends) on the *lw* instruction immediately before it. The *lw* instruction depends (and only depends) on the *add* instruction immediately before it. Calculate CPI on MIPS 5-stage pipeline datapath with and without forwarding. [2+2 = 4 marks]



Name: MALIK NAMMAD RAISAL

Roll No: 2021CS10559

4. In a single cycle processor, all instructions use a common clock. The table shows the time needed by three instruction types, and their respective percentage in a set of instructions. What is the fastest clock cycle this single cycle processor can use? What percentage of time is wasted in executing 10 instructions? [3 marks]

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

→ Fastest

Table 1: Instruction details

Time of single cycle = Max of time of all possible instructions  
= 800 ps

Time taken for 10 instructions =  $10 \times 800 = 8000$  ps

Execution time required for 10 instructions =  $\left( \frac{30}{100} \times 800 + \frac{20}{100} \times 200 + \frac{50}{100} \times 250 \right) \times 10$   
= 4050 ps

Time wasted =  $8000 - 4050 = 3950$

$$\frac{3950}{8000} = 49.375\%$$

5. A processor takes 100ns and 100pJ for every instruction. It can be infinitely pipelined with each pipeline register taking 2ns and 2pJ. What is the throughput, latency per instruction and energy per instruction for a 100 stage processor, compared to the original processor? [3 marks]

Ideal pipeline as each stage takes  $\frac{100}{100} = 1$  ns, ~~100 ns~~ ~~100 pJ~~ ~~for registers~~  
+ 2 ns for registers = 3 ns

Latency = No. of stages  $\times$  Cycle time  
=  $100 \times 3$  ns = 300 ns

Throughput =  $\frac{1}{\text{Cycle time}} = \frac{10^9}{3} = 0.33 \times 10^9$  instructions per second

Pipeline registers only between stages as 99 registers  
Energy per instruction =  $1 \mu\text{J} \times 100 + 2 \mu\text{J} \times 99 = 298 \mu\text{J}$

Latency increases 3 times. | ~~Throughput~~ Final throughput =  $10^7$   
Energy rises  $\frac{298}{100} = 2.98$  times | ~~Throughput~~ ~~increases~~  $\frac{33.33}{3}$  times

Name: MALIK HAMMAD RAJUL

Roll No: 2021CS1055A

6. The 5 stages of a processor have the following latencies.

Fetch	Decode	Execute	Memory	Writeback
300ps	400ps	350ps	500ps	100ps

Table 2: Processor latencies

Assume that when pipelining, each pipeline stage costs 20ps extra for the registers between pipeline stages. If you could split one of the pipeline stages into 2 equal halves, which one would you choose? Do you see any improvement in cycle time, latency for one instruction and throughput, with this split? [4 marks]

If we do not split 'memory', cycle time = max of the of all stages would not change so throughput will also not change & latency = cycle time x pipeline stages increases.

So we should split the 'memory' stage into two 250 ps stages.

→ Since ~~each~~ registers are between stages, assuming no pipeline register after writeback stage.

Before Split

Fetch	Decode	Execute	Memory	Writeback
300+20	400+20	350+20	500+20	100

~~Latency~~ ~~300~~ ~~300+400~~  
 Clock cycle time = max of all stage = 520 ps.

Latency =  $520 \times 5 = 2600$  ps

Throughput =  $\frac{1}{\text{Cycle time}} = 1.92 \times 10^9$  instructions per second

After Split

Fetch	Decode	Execute	Memory	Memory	Writeback
300+20	400+20	350+20	250+20	250+20	100

Clock cycle time = 420 ps

Latency =  $420 \times 6 = 2520$  ps

Throughput =  $\frac{1}{\text{Cycle time}} = 2.38 \times 10^9$  instructions per second

Clock cycle time, latency & throughput all improve after split.

Name: MALIKA NAMMAD RAISAL

Roll No: 2021CS10559

7. A 1-bit saturation counter toggles between 0,1 states and a 2-bit counter toggles between 00,01,10,11 states. We build two branch predictors, one with 1-bit and the other with 2-bit counters. We have two traces of branch taken (T) and not-taken (N): Trace1: NNNTNNN, Trace2: NTNTNTN. What will be the accuracies for each trace for each predictor? Assume each counter starts at strongly not taken state. 5 marks.

1 bit ~~saturation~~ counter

States → Trace 1

Si	Predictor	Actual
0	N	N
0	N	N
0	N	N
0	N	N
1	T	T
0	N	N
0	N	N

For trace 1 →  $\frac{5}{7}$  predicts correct  
 1 bit counter  
 $= 0.7142 = 71.42\%$   
 ~~$= 0.2857 = 28.57\%$~~

~~Strongly Not Taken~~

N → we inputs from head  
T

2 bit counter

Si	Predictor	Actual
00	N	N
00	N	N
00	N	N
00	N	T
01	N	N
00	N	N
00	N	N

For trace 1  
 2 bit counter →  $\frac{6}{7}$  correct  
 $= 0.8571 = 85.71\%$

Trace 2

Si	Predictor	Actual
0	N	N
0	T	T
1	N	N
0	T	T
1	N	N

For trace 2 →  $\frac{1}{7}$  predicts correct  
 1 bit counter  
 $= 0.1428 = 14.28\%$

Trace 2

Si	Predictor	Actual
00	N	N
00	T	T
01	N	N
00	T	T
01	N	N
00	T	T
01	N	N

For trace 2  
 2 bit counter →  $\frac{4}{7}$  correct  
 $= 0.5714 = 57.14\%$

Name: MALIK NAMMAD RAJSAI

Roll No: 2021CS1055A

8. What operations can the following ALU perform? Please label the diagram to describe which operation each input, gate and output in the ALU is related to? [5 marks]

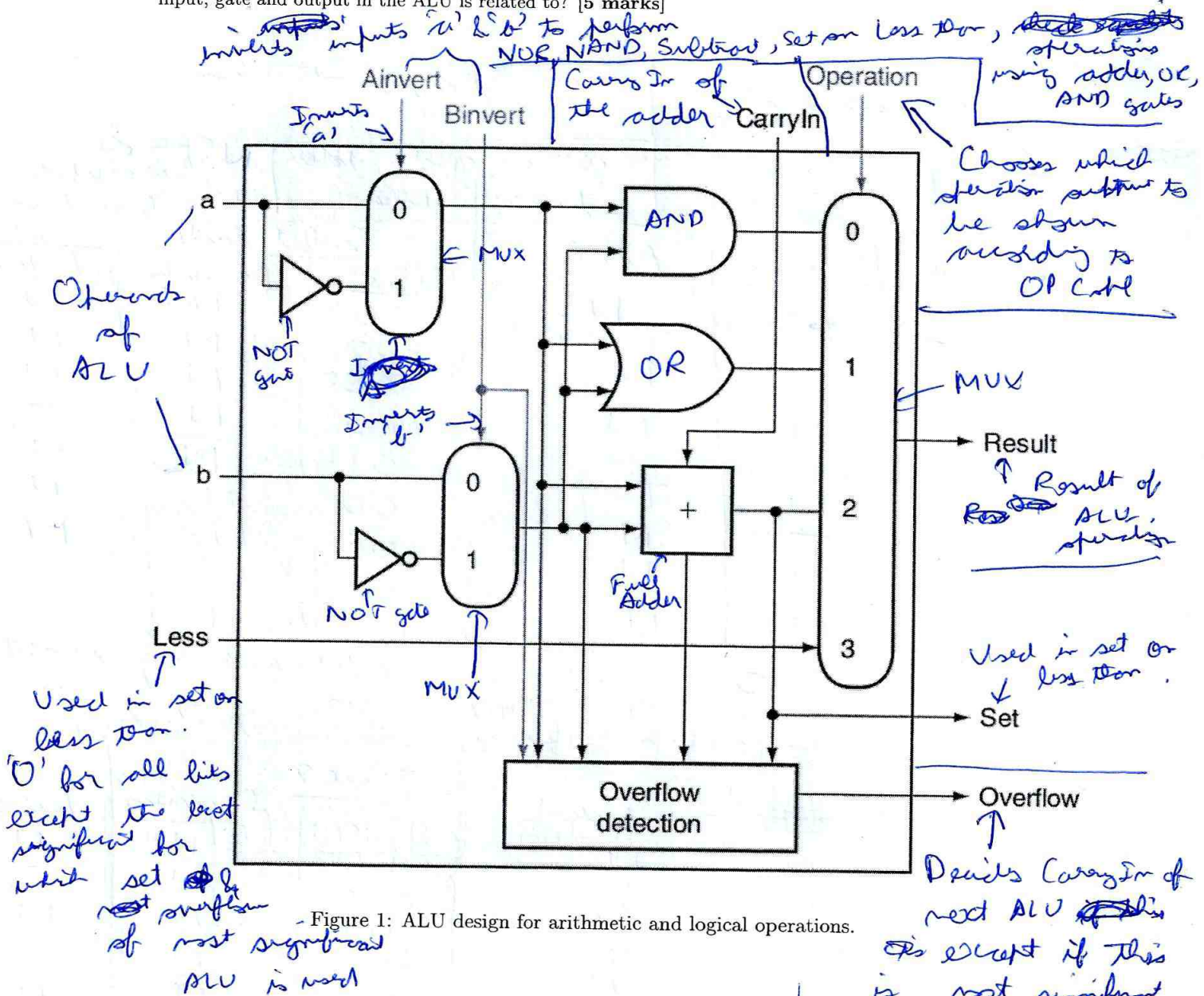


Figure 1: ALU design for arithmetic and logical operations.

ALU can perform set or less than, ~~check overflow~~, add, subtract, OR, AND, NOR, NAND.