

Name: Mani Sasthale

Roll No: 2021CB10095

(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]

Stall only in cases of load and R type
load is 20%. and ~~independent~~ compiler can
optimize for half of them. i.e. stall for
the other half = $20\% \times \frac{1}{2} = 10\%$.

$$\text{Total } \cancel{\text{slowdown}} \text{ CPI} = (0.8) \times 1 + (0.1) \times 1 + (0.1) \times 2$$

normal. lw needs stall
instru- optimized

$$= 1.1$$

similar for ideal case it should be 1.

$$\text{So slowdown} = \frac{1}{1.1} = 0.909 \approx 0.91$$

Mani Sasthale.

Name: 2021CS10095

Roll No: 2021CS10095

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

IF → It fetches the instruction solely on the input from the PC ~~and~~ to which no forwarding is used and it executes every cycle.

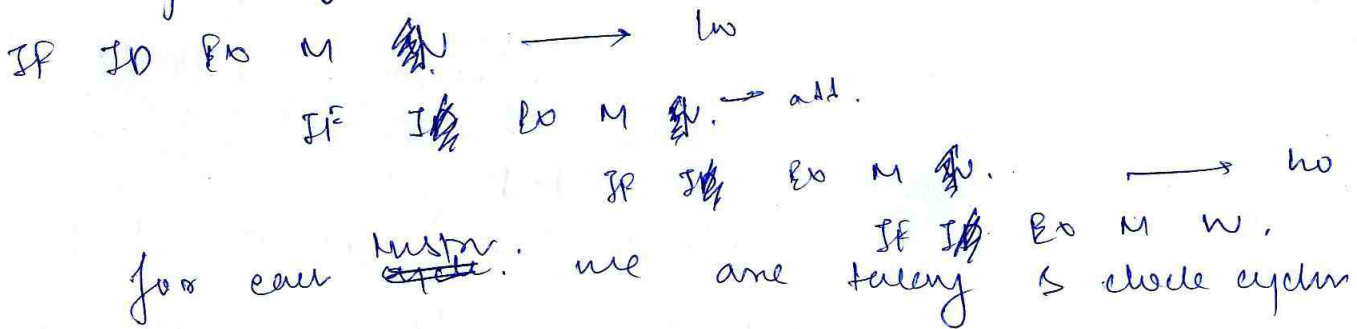
ID → It simply decodes whatever comes from the IF and because it is allowed to contain unnecessary info/wrong info assuming bypassing corrects it we ~~can't~~ bypass to it.

W → It writes the end result which is always correct as our implementation & needs no extra bypassing.

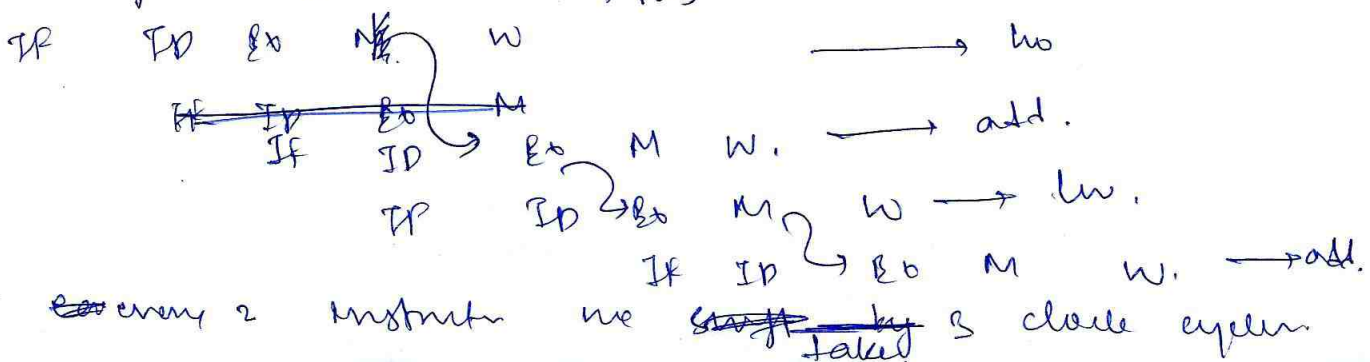
used in MEM: eg. lw \$2 0(\$4); sw \$2 4(\$4)
in EX: eg. (very common) from Mem to Mem, Ex to Mem, Mem to Ex.

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]

without forwarding. (CPI = 3)



with forwarding (CPI = 3/2 = 1.5)



for every 2 instructions we ~~take~~ take 3 clock cycles.

Name: Mansi Sonkar

Roll No: 201210095

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% |

Table 1: Instruction details

Because the single cycle we have to account for the largest time taken by any type the fastest cycle can have length of 800 ps minimum.

By the data 10 instructions will contain, 3 of type 1, 2 of type 2, 5 of type 3.

So total time taken = $(800 \text{ ps}) \times 10 = 8000 \text{ ps}$.

whereas individual time = $(3 \times 800 + 2 \times 200 + 5 \times 250) \text{ ps} = 4050 \text{ ps}$.

Wasted time = 3950 ps , total time = 8000 ps .

\therefore % wasted time in 10 ins = 49.375%. (almost $\frac{1}{2}$ of time)

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]

Original processor (100 ns, 100 pJ) for every instr.

for 100 stage processor we have to add 100 pipeline registers and split it in eq. parts.

ie each cycle = $1 \text{ ps} + 2 \text{ ps} = 3 \text{ ps}$. (assuming instr. is large)

energy now. each of the 100 pipelines will take 2pJ. along with the original processor = $100 \text{ pJ} + 20100 \text{ pJ} = 20200 \text{ pJ}$.

so the changes are:

latency 100 ns \rightarrow ~~300~~ 200 ns. $(30)(100 + 20100)$.

power 100 pJ \rightarrow 200 pJ. (30)

throughput. 100 ns \rightarrow 3ns. $(\frac{1}{300} \%)$ (very less?)

Name: Yami Sathale

Roll No: 2012CS10095

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]

I will split the longest time taking stage i.e. memory
 So that I have 6 stages of 250 + 20 (register)
 rather than 5 (500 + 20). Now the cycle time
 reduces from 520 to ~~420~~ 420 (decode + register)
 and ~~stages~~ stages increase from 5 to 6.

For single instruction.

latency prev = $(500 + 20) \times 5$ ps = 2600 ps.
 latency now = $(400 + 20) \times 6$ ps = 2520 ps.

cycle time (prev) = 520 ps
 " " (now) = 420 ps.

throughput earlier = $\frac{1s}{520ps} = 2.192$ BIPS (in long pipeline takes almost time for a single cycle assuming no dependency)
 " " now = $\frac{1s}{420ps} = 2.38$ BIPS.

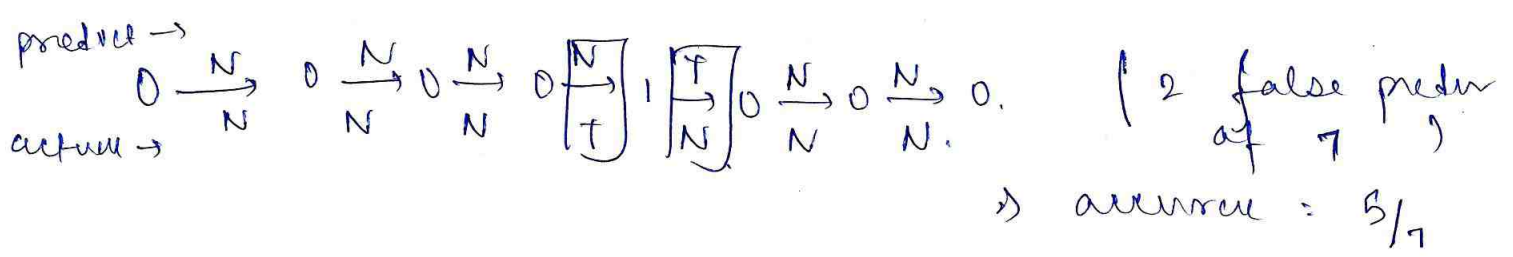
Hence all three metrics are in favour of pipelining.

Name: Mari Santine

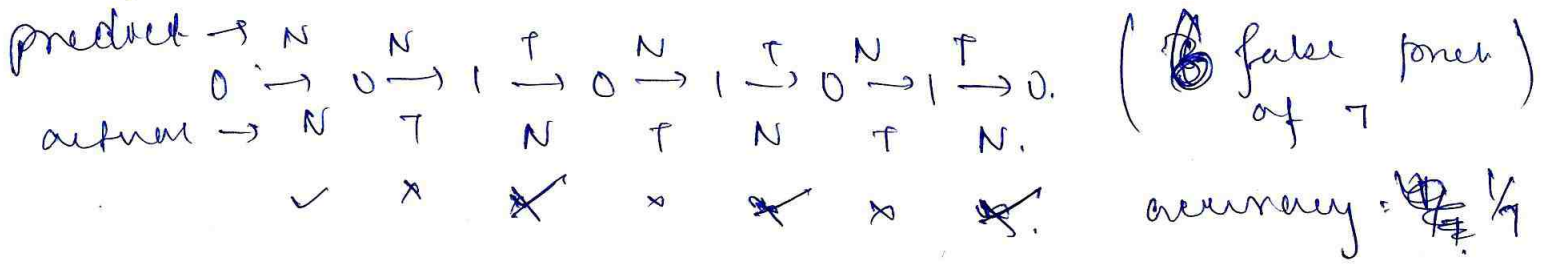
Roll No: 201CS0095

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: NNNNTNNN, Trace2: NTNTNTN. What will be the accuracies for each trace for each predictor? Assume each counter starts at strongly not taken state. 5 marks.

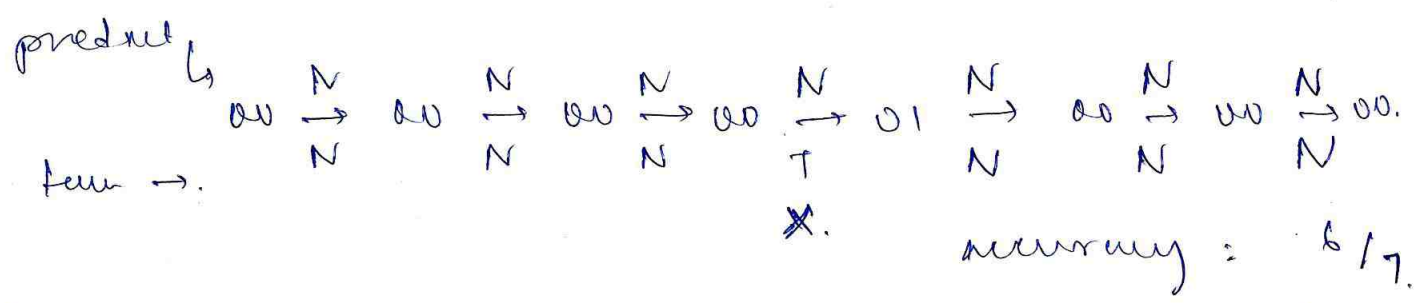
for 1 bit trace 1 NNNNTNNN. Start at not taken(0)



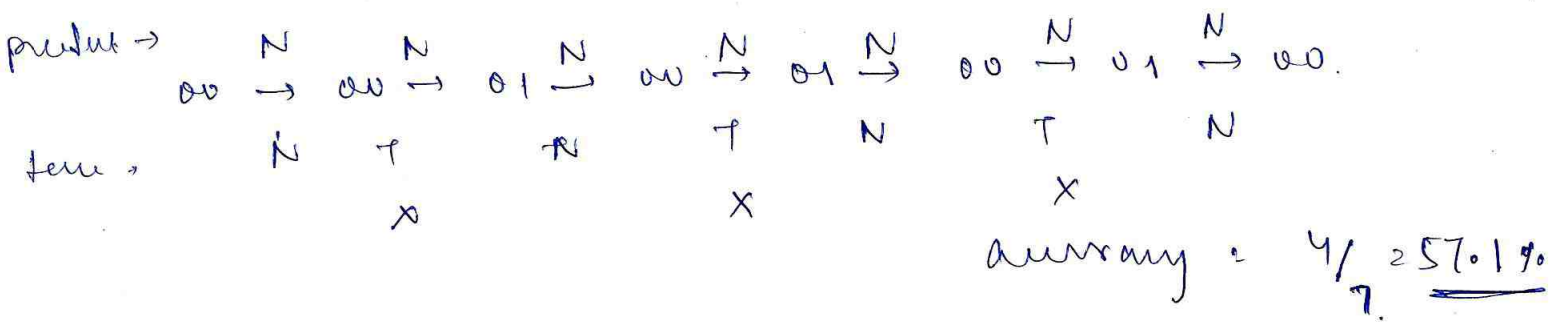
for 1 bit trace 2 NTNTNTN.



for 2 bit trace 1 NNNNTNNN.



for 2 bit trace 2 NTNTNTN.



Name: Mari Sorshall

Roll No: 201210095

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]

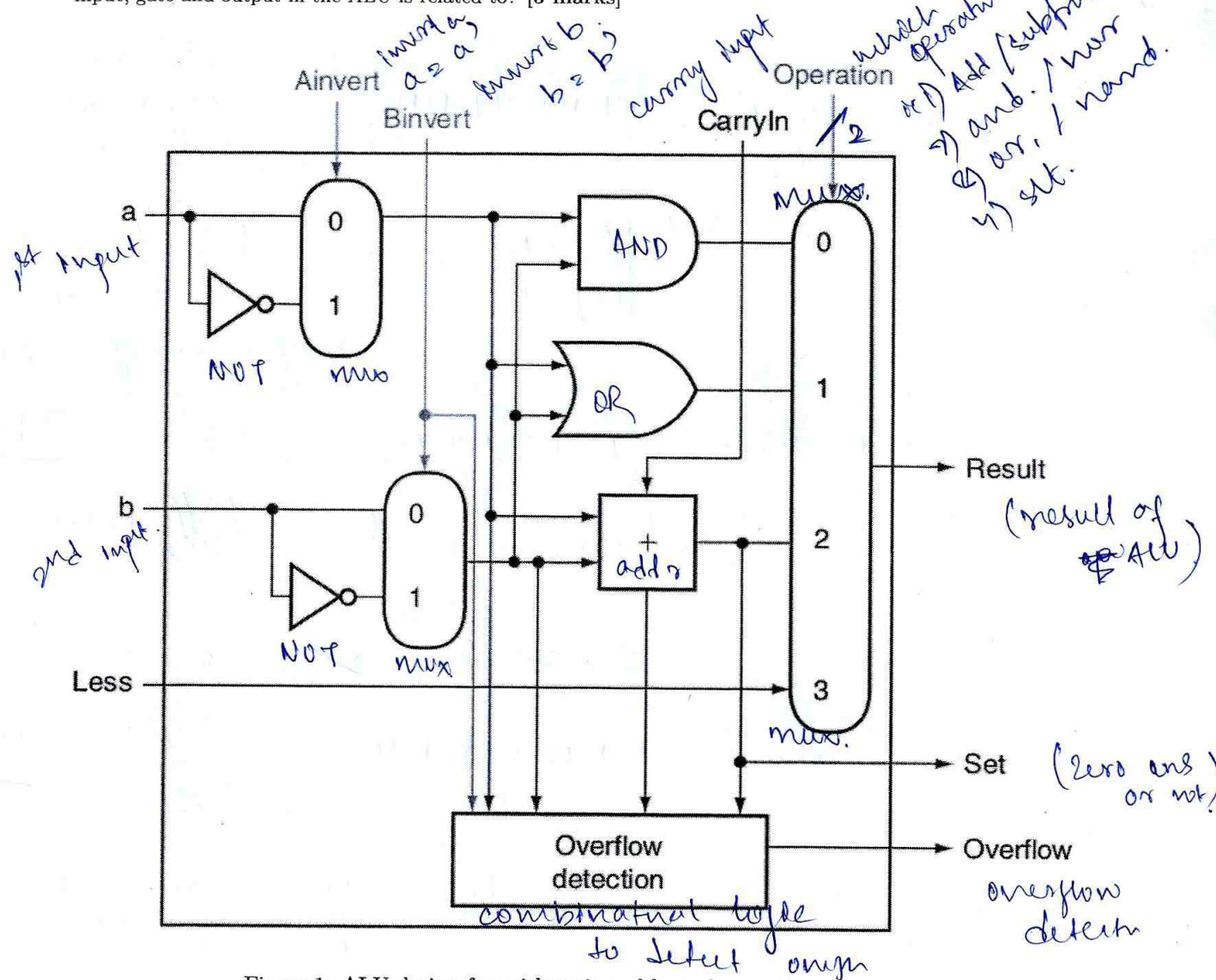


Figure 1: ALU design for arithmetic and logical operations.

| Operations | CIN | AIN | BIN | Less | opm |
|------------|-----|-----|-----|------|-----|
| Add | 0 | 0 | 0 | 0 | m1 |
| Subtract | 1 | 0 | 1 | 0 | m2 |
| AND | 0 | 0 | 0 | 0 | m3 |
| OR | 0 | 0 | 0 | 0 | m3 |
| NAND | 0 | 1 | 1 | 0 | m2 |
| NOR | 0 | 1 | 1 | 0 | m2 |
| SET | 1 | 0 | 1 | 1 | m4 |

operations of same kind represent similar lens of instructions