2202-COL216 Minor2

UTKARSH SINGH 2021CS10558

TOTAL POINTS

20 / 30

QUESTION 1

1 Question 1 3 / 3

✓ **+ 3 pts** *Correct*

+ 2 pts Partially correct 2

+ 1 pts Partially correct 1

+ 0 pts Incorrect

QUESTION 2

2 Question 2 2 / 3

+ 0 pts Incorrect

+ 1 pts 1 stage mentioned

✓ **+ 2 pts** *2 stages mentioned*

+ 3 pts 3 stages mentioned

- 1 pts Reason not mentioned / more than 3 stages mentiond

QUESTION 3

3 Question 3 2 / 4

+ 4 pts Correct

+ 0 pts Incorrect Both/Missing/No Explanation

+ 2 pts One Correct

✓ **+ 2 pts** *both partial correct*

+ 1 pts without forward partial correct

+ 1 pts with forwarding partial correct

QUESTION 4

4 Question 4 3 / 3

✓ **+ 3 pts** *Correct*

+ 1.5 pts Partially Correct 2

+ 1 pts Partially Correct 1

+ 0 pts Incorrect

QUESTION 5

5 Question 5 3 / 3

✓ **+ 1 pts** *Latency*

✓ **+ 1 pts** *Energy*

✓ **+ 1 pts** *Throughput*

+ 1 pts Partially Correct

+ 0 pts Incorrect/Not Attempted

QUESTION 6

6 Question 6 1 / 4

✓ **+ 0.5 pts** *Memory selected*

✓ **+ 0.5 pts** *reason why selected memory*

+ 0.5 pts initial cycle time

+ 0.5 pts after split cycle time

+ 0.5 pts initial latency

+ 0.5 pts final latency

+ 0.5 pts initial throughput

+ 0.5 pts throughput after split

+ 0 pts incorrect or not attempt

QUESTION 7

7 Question 7 1 / 5

✓ **+ 0 pts** *Nothing Significant is answered.*

+ 2.5 pts Correct answer for any one of the trace.

+ 1 pts Tried trace 1 but wrong answer.

+ 1 pts Tried trace 2 but wrong answer.

+ 5 pts All correct.

+ 1 *Point adjustment*

QUESTION 8

8 Question 8 5 / 5

- 5 pts Not attempted.

✓ **- 0 pts** *Correct. 1 mark for operations AND, OR, SUM. 1 marks for mentioning that each of these operations can be done with COMPLEMENT i.e. NAND/NOR, 1 mark for set less than operation with output, 1 mark for MSB/overflow detection, 1 mark for labeling the gates and marking input/output.*

- 1 pts Missed one of the operations or missed mentioning complement, or missed MSB/overflow.

- 2 pts Missed more operations and details.

- 3 pts Mentioned a few operations.

- 4 pts Attempted, but hardly wrote anything or wrote wrong things.

Name: Utkarsh Singh Roll No: 2021CS10558

(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]**

Sol^n) Say there are 100 instruction. Out of which 20 loads are there. Since half of them is succeeded by instruction which don't use them, therefore only 10 loads are present which can contribute to delay.

Assume clockcycle = x

∴ Latency in the case when no delays are present = $100x$

Latency in the current scenario = $(100 + 10)x = 110x$ (10 stalls due to 10 loads)

[∴ Slowdown rate = $\frac{110x - 100x}{100x} \times 100 = 10\%$]

Name: Utkarsh Singh Roll No: 2021CS10558

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

IF : Instruction fetch can never be a destination because it just takes the ~~the~~ mips instruction as it is, it has nothing to do with the previous values of registers.

W: It writes the ~~input~~ data back to the register, the data either comes from the ALU or the memory which ~~are~~ are already ~~sour~~ destinations of forwarding or ~~many~~ ~~it~~ for the rest of the time forwarding takes place before them. Therefore W cannot be a destination.

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]**

Eg: [lw $t0 4($t1)
add $t2 $t1 $t3
lw $t4 4($t2)]

Sample instruction (My answer is based on this)

With forwarding!

$= 5 + (5+1) \times 500 + 499 \times (5+0)$

(5: first lw; (5+1): add instruction; (5+0): other lw)

$= 5 \times 500 + 6 \times 500 = 11 \times 500 = 5500$ (total clockcycles)

CPI $= \frac{5500}{1000} = 5.5$

Without forwarding!

$= 5 + (5+2)500 + 499(5+2)$ [2 stalls each for lw and add]

$= 5 + 7 \times 999$

$= 6998$ (total clock cycles)

CPI $= \frac{6998}{1000} = 6.998$

Name: Utkarsh Singh Roll No: 2021CS10558

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*

Sol^n fastest clock cycle = 800 ps

Actual time taken for (100 sample instructions) = 800 × 100 ps
= 80000 ps

Ideal time that should be taken = 800 × 30 + 200 × 20 + 250 × 50
= 24000 + 4000 + 12500
= 40500 ps

% of time wasted = $\frac{80000 - 40500}{80000} \times 100 = \frac{39500}{80000} \times 100 = 49.375\%$

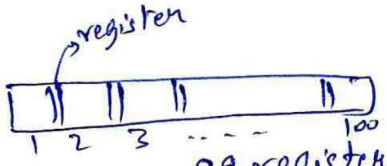
5. A processor takes 100ns and 100pJ for every instruction. It can be infinitely pipelined with each pipeline register taking 2ns and 2pJ. What is the throughput, latency per instruction and energy per instruction for a 100 stage processor, compared to the original processor? **[3 marks]**

Throughput of original = $\frac{1}{100 \times 10^{-9}} = 10^7 = \frac{1}{100\,ns}$

Latency = 100 ns (Original)

Energy of original = 100 pJ

Clock cycle of new processor = 1ns + 2ns = 3ns (1 ns for a stage, 2 ns for register)



99 registers used.

∴ New throughput = $\frac{1}{3\,ns}$ [$\frac{100}{3}$ times than before]

New latency = (3 × 100) ns = 300 ns (3 times the original)

New Energy = 99 × 2 pJ + 100 pJ = 298 pJ (2.98 times original)

Name: Utkarsh Singh Roll No: 2021CS10558

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

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

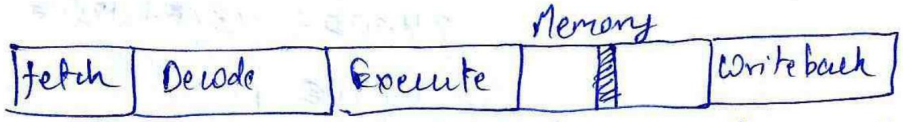
Table 2: *Processor latencies*

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**]

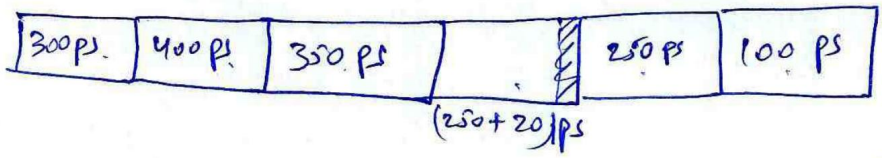
Sol^n Current cycle time = 500ps

Current latency for one instruction = 500ps × 5 = 2500ps

Current throughput = $\frac{1}{500} \times 10^{12}$



I would split Memory because this will reduce my clockcycle



New cycle time = 400ps (100ps Better!)

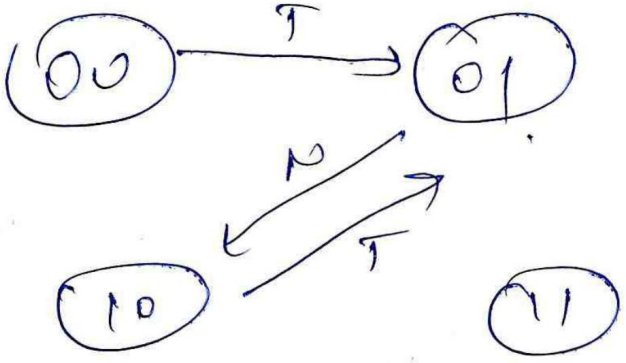
New Latency = 400ps × 6 = 2400ps (100ps Better!)

New Throughput = $\frac{1}{400} \times 10^{12}$ ($\frac{5}{4}$ times Better!)

Name: Utkarsh Singh Roll No: 2021CS10558

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**.

	Initial State	Input	final state
Trace 1	N	0	



Name: Utkarsh Singh Roll No: 2021CS10558

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]**

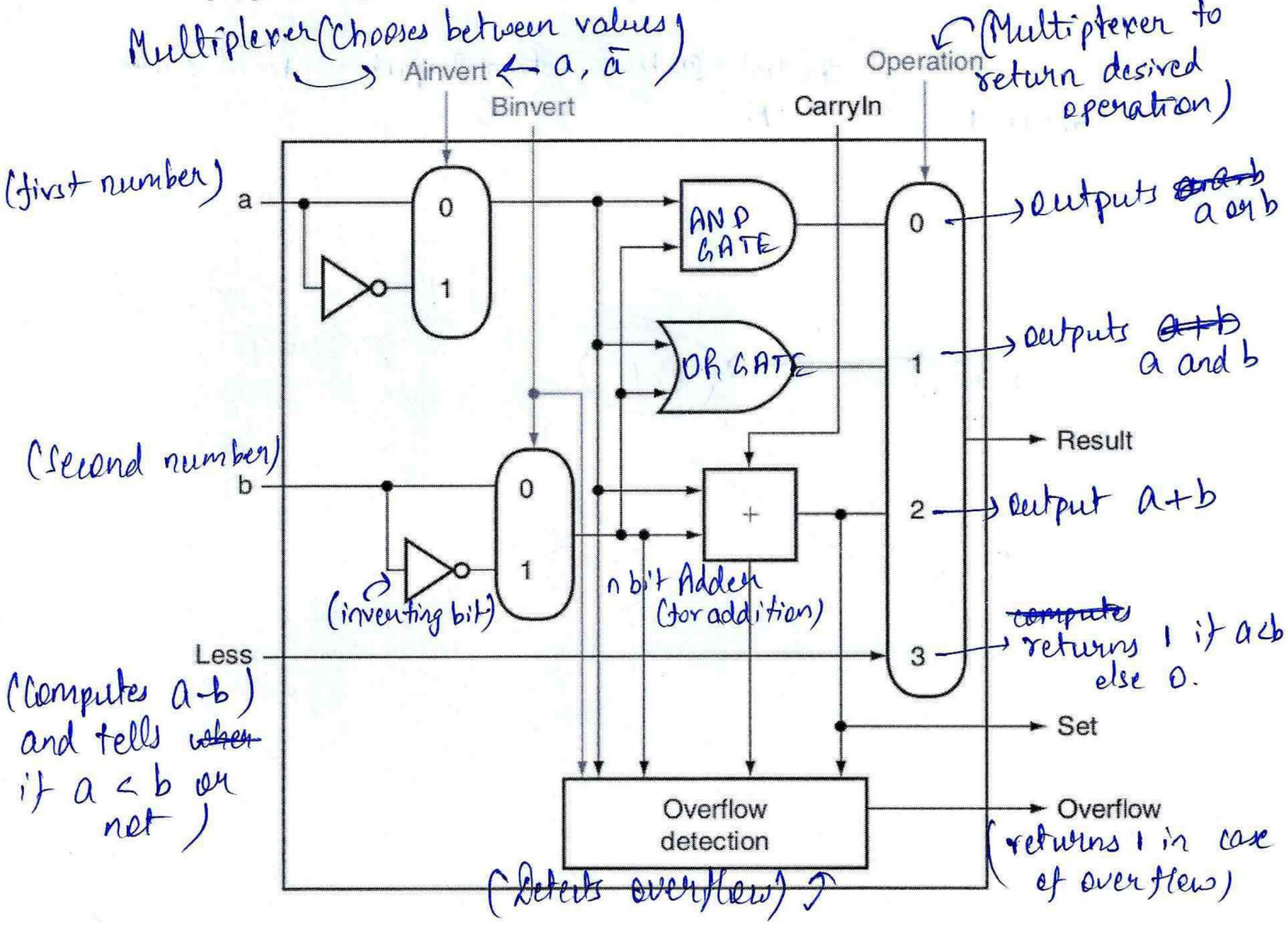


Figure 1: ALU design for arithmetic and logical operations.

→ The given ALU can perform AND, OR, addition, checkless than operations.

→ By default carryIn is zero but in case of less than operation we need to compute a−b so for that CarryIn is equal to 1.

→ If during addition a+b exceeds the number of bits. then overflow detector will set to 1.