

Q1) Match the following

- a) Solving quadratic equations — Algorithm level
- b) Design of control unit architecture — Micro-architecture level
- c) Design of an adder — Digital logic level
- d) System calls — Operating System level
- e) Assembler Directives — Assembly language level
- f) Format of machine code — Instruction Set Architecture

Q2) True or False

- a) If the earliest point that the branch target is known in the MIPS pipeline is the EX stage, then a two cycle branch penalty results — TRUE
- b) The python interpreter is compiled — TRUE
- c) Direct Execution uses Read Only Memory (ROM) but it is faster than that used in Microprogramming — FALSE
- d) Unconditional jumps need branch predictions — TRUE
- e) In RISC ISAs, only load & store instructions access data memory — TRUE
- f) ~~The loader reads a machine code file from disk to memory, resolves DLLs, & transfers control to it~~ — TRUE
- g) Assembly language explicitly lists the types of symbols declared in it — TRUE
- h) The loader reads a machine code file from disk to memory, resolves DLLs, & transfers control to it — TRUE
- i) An ALU is an example of a sequential circuit — FALSE
- j) In a MIPS pipeline, the WB stage writes to registers only in the second half of the cycle — FALSE
- k) Latches between stages in pipelines contain the result of the previous stages as well as the TREG of the instruction in that stage — TRUE

Q3) a) Let us say for the sake of simplicity that we have 100n instructions in the program.

Since 12% are control transfer instructions so,

$$\text{Control transfer instructions} = \frac{12}{100} \times 100n = 12n$$

$$\text{Other instructions} = \frac{88}{100} \times 100n = 88n$$

for our 12n control transfer instructions, we have 94% branch predict<sup>n</sup> accuracy

Therefore,

$$\text{Correctly predicted} = \frac{94}{100} \times 12n = 11.28n$$

$$\text{Incorrectly predicted} = 12n - 11.28n = 0.72n$$

With a branch penalty of 2 cycles, the incorrectly predicted instructions take 3 cycles to complete whereas all the other instructions take 1 cycle (Assuming none of these are boot up instructions)

$$\therefore \text{Total no of instructions} = 88n + 11.28n + 0.72n \times 3 \\ = 101.44n$$

With branch penalty of 1 cycle, the incorrectly predicted instructions take 2 cycles to complete

$$\therefore \text{Total no of instructions} = 88n + 11.28n + 0.72n \times 2 \\ = 100.72n$$

$$\text{Speed} = \frac{101.44n - 100.72n}{101.44n} \times 100$$

$$= \frac{0.72n}{101.44n} \times 100 = \frac{0.71\%}{\hookrightarrow \text{Ans}}$$

b) Considering a 25 stage pipeline where an error results in 10 cycle penalty, each wrong instruction takes 11 cycles:

$$\therefore \text{Total} = 88n + 11 \cdot 28n + 11 \times 0.72n = 107.2n$$

Reducing this to 5 stage penalty, each wrong instruction takes 6 cycles

$$\therefore \text{Total} = 88n + 11 \cdot 28n + 6 \times 0.72n = 103.6n$$

$$\text{Speed up} = \frac{107.2n - 103.6n}{107.2n} \times 100$$

$$= \frac{3.6n}{107.2n} \times 100 = \frac{3.36}{\square} \%$$

Ans

Q4(a) To resolve the condition in the DE stage we need an extra hardware (namely an adder) to calculate PC + target & any comparison may also be handled at that stage.

However, if ~~the~~ we are to resolve the branch at the 3<sup>rd</sup> (EX stage), then no extra hardware is required and PC + target ~~as well~~ as well as the comparison (if any) must be moved from the 2<sup>nd</sup> to the 3<sup>rd</sup> stage where it is resolved by the ALU.

Additionally, when we move branch resolution to the 3<sup>rd</sup> stage, we need to kill the next 2 instructions in the DE and IF register and a delay of 2 cycles results.

- b) Adding the additional calculation for comparison happens at the

Conditional comparison (in case of conditional branch statements) must happen at the end of DE stage when branch resolve takes place in the 2<sup>nd</sup> stage and may increase the cycle time. This is something that is very harmful as it slows the entire pipeline (since each stage of the pipeline executes in 1 cycle) and to avoid this, we may prefer branch resolution in the EX stage.

Q5)

- a) The given program fragment iterates over the "loop" segment and adds 10 numbers whose results are stored in \$4 register.
- b) The loop executes for 10 iterations before it exits
- c) On a non pipelined machine :

$$1 \text{ iteration} \rightarrow 5 \text{ instructions}$$

$$10 \text{ iterations} \rightarrow 10 \times 5 = 50 \text{ instructions}$$

$$1 \text{ instruction} \rightarrow 5 \text{ cycles}$$

$$50 \text{ instructions} \rightarrow 50 \times 5 = 250 \text{ cycles}$$

$\underbrace{\hspace{1cm}}$   
Ans

d)

Pipeline stall & delays are used here to resolve data & control dependencies respectively.

Instruct <sup>n</sup>	Number of stall/delay	Type of hazard
LW \$5, 0(\$3)	0 - for 1 <sup>st</sup> iterat <sup>n</sup> 2 - common case (all other iterat <sup>e</sup> )	control
ADD \$4, \$4, \$5	2	data
SUB \$3, \$3, 4	0	-
SGTI \$6, \$3, 100	2	data
BNEZ \$6, -loop	2	data

c) Cycles taken:

$$1^{\text{st}} \text{ iterat}^n : 1 + \underbrace{3 + 1}_{\substack{\text{No branch prediction} \\ \text{in 1st cycle}}} + 3 = 11 \text{ cycles}$$

$\downarrow$  2 stalls  
 $\downarrow$  + 1 cycle to execute

$$2^{\text{nd}} - 10^{\text{th}} \text{ iterat}^n : (3 + 3 + 1 + 3 + 3) \times 9 = 13 \times 9 = 117 \text{ cycles}$$

[No extra ~~cycles~~ after loop exits because there is no branch predict<sup>n</sup> so PC always goes to PC+4 which is the correct instruct<sup>n</sup> after loop exits]

$$\text{Total} = 117 + 11 = \underline{128 \text{ cycles}} \rightarrow \text{Ans}$$

d) Hypothetical ideal Pipeline

$$= 10 \times 5 \times 1 \quad (1 \text{ cycle for each instruct}^n \text{ ideal pipeline})$$

$$= 50 \text{ cycles}$$

~~Speedup in ideal~~ Instruction reduction

$$= \frac{250 - 50}{250} \times 100 = \frac{200}{250} \times 100 = \cancel{80\%}$$

⇒ 80% reduction in the number of instructions  
(or in other words; ideal pipeline executes 5x faster)

ii) Real pipeline instruction reduction

$$= \frac{250 - 128}{250} \times 100 = \underline{48.8\% \text{ reduction}}$$

(Speed =  $\frac{250 - 128}{128} \times 100 = \underline{95\% \text{ increase in speed}}$ )

g) Improved pipeline has bypass paths & branch predictor

$$1^{\text{st}} \text{ iteration: } 1 + 2 + 1 + 1 + 1 = 6 \text{ cycles}$$

↓

→ ADD instruction still has 1 cycle stall because we cannot allow ~~use of~~ \$5 register till the previous LW instruction loads from memory in the 4<sup>th</sup> stage

$$2^{\text{nd}} \text{ iteration: } 3 + 2 + 1 + 1 + 1 = 8 \text{ cycles}$$

↓

→ 1<sup>st</sup> prediction is wrong for BTB (PC+4)

$$3^{\text{rd}}-9^{\text{th}} \text{ iteration: } (1 + 2 + 1 + 1 + 1) \times 7 = 6 \times 7 = 42 \text{ cycles}$$

$$10^{\text{th}} \text{ iteration: } (1 + 2 + 1 + 1 + 1 + 2) = 8 \text{ cycles}$$

↓

→ 2 cycle delay because branch predictor predicts incorrectly when loop exits

$$\text{Total} = 6 + 8 + 42 + 8 = \underline{\underline{64}} \text{ cycles} \rightarrow \text{Ans}$$

b) Instruct<sup>n</sup> reduction

$$= \frac{250 - 64}{250} \times 100 = 74.4\% \text{ reduction}$$

$$\left( \text{Speed} = \frac{250 - 64}{64} \times 100 = 290.6\% \text{ faster} \right)$$