CS230 DLDCA Quiz-3, Fri 11 Oct 2024, 08:40am-09:25am, Max. Marks: 5

Name: SAMPLE ANSWERS Roll number: _____ House: _____

General instructions

- Write only in the space provided. Answer briefly but crisply (not lengthily or loosely).
- You are allowed to refer to your own hand-written notes only.
- Write neatly and clearly. Up to **+2 HP** for neat handwriting, neat/crisp answers.
- Answers generally have to be (briefly) explained. State any necessary assumptions.

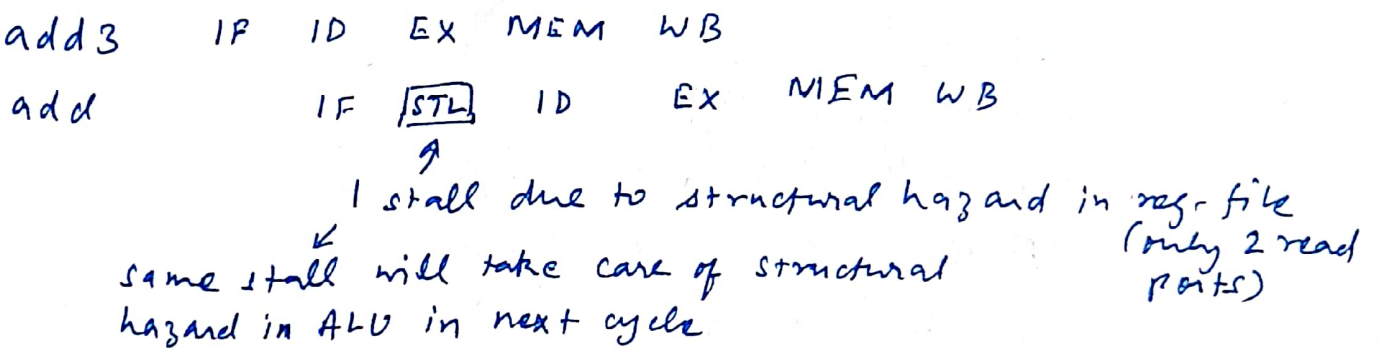
1. Consider the add3 instruction as an extension to the MIPS32 instruction set. It also has the R-format, but with the 5-bit shamt field being called as Rs1, and used as the third register to be added. The destination register remains the same as earlier, i.e. Rd.

opcode	Rs	Rt	Rd	Rs1	funct

We are now going to consider a pipelined implementation of add3, as an extention to the original 5-stage MIPS pipeline. We will assume a register file with only two read ports. The execution plan is that the third register is going to be read in the EX stage, while the first two registers are being added in the ALU. And the third register will be added using the same ALU in the 4th cycle (i.e. the MEM stage). The pipeline remains of 5 stages.

(a) **[1 mark]** For the following instruction sequence, show the pipeline timing diagram with the necessary stalls/bubbles. Indicate the reason for any stall.

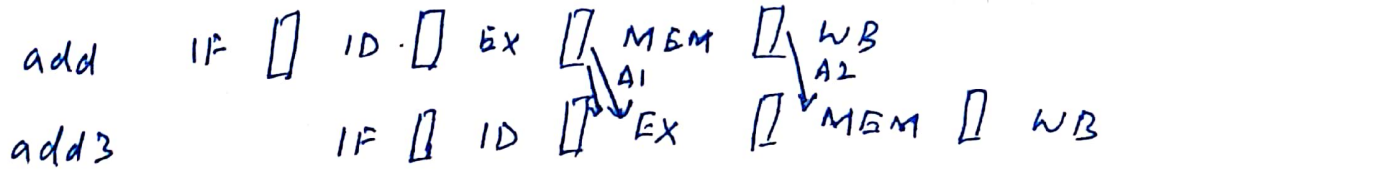
```
add3    $t0, $t1, $t2, $t3      # t0 = t1+t2+t3
add     $s0, $s1, $s2           # s0 = s1+s2
```



IMPORTANT: For ALL the remaining parts below, assume that the required hardware enhancements are done to completely avoid structural hazards involving add3.

(b) **[1 mark]** For the following instruction sequence, show the necessary data-forwarding to minimize stalls. For each data forwarding, show the source latch and the place (i.e. hardware unit) where it is forwarded.

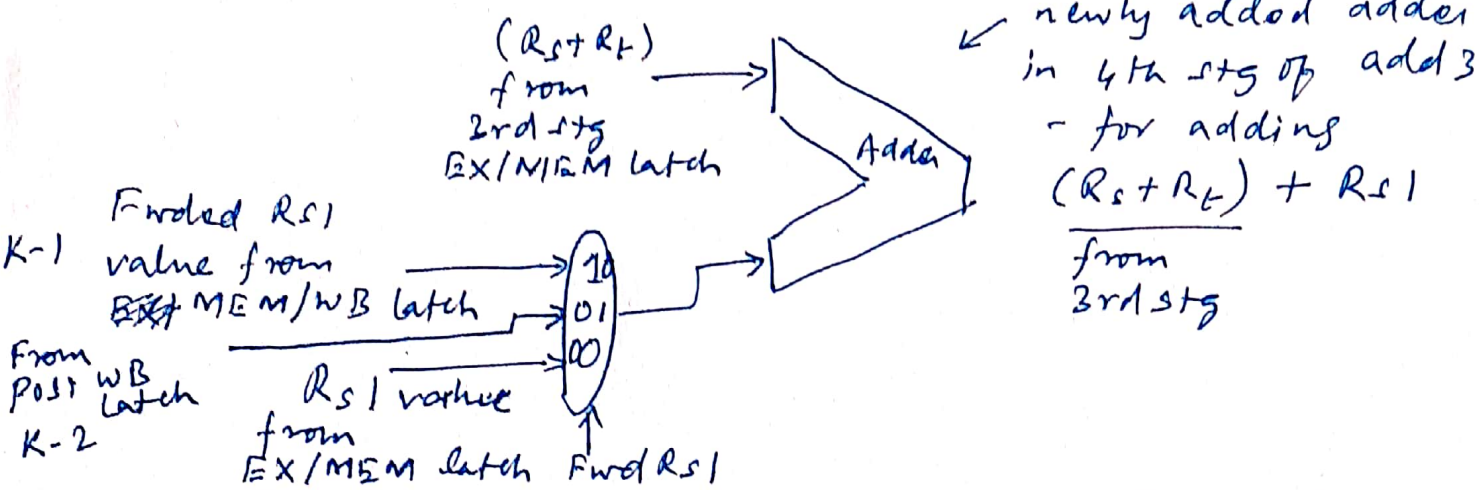
```
add     $t0, $t1, $t2           # t0 = t1+t2
add3    $s0, $t0, $t0, $t0      # s0 = t0+t0+t0
```



A1 has 2 forwardings - one to reg val 1, one to reg val 2 in front of main ALU

A2 has forwarding to input Rs1 of (newly added) adder used in MEM stg of add3 4th

(c) **[1 mark]** In such a machine, what data-path *changes* are needed to enable forwarding to the 4th stage of add3? Draw the mux/muxes needed in front of the relevant hardware unit. You need to consider only the case of a register-register instruction (such as add) being the data producing instruction.



(d) **[2 marks]** Write the pipeline control pseudocode logic, to be executed in the ID stage of add3, to generate the control line(s) for the above mux/muxes. Here too, you need to consider only the case of a register-register instruction (such as add) being the data producing instruction.

```
if ((IF/ID.opcode == add3) &&
    (IF/ID.Rs1 == ID/EX.Rd) &&
    (ID/EX.RegWr == 1) &&
    (ID/EX.MemRd == 0))
    FwdRs1 = 10  // from MEM/WB latch, from K-1
else if ((IF/ID.opcode == add3) &&
    (IF/ID.Rs1 == EX/MEM.Rd) &&
    (EX/MEM.RegWr == 1) &&
    (EX/MEM.MemRd == 0))
    FwdRs1 = 01  // from Post WB latch, from K-2
else
    FwdRs1 = 00  // no forwarding
```

2. **Optional, for House Points (up to 5HP):** Starting Mar 2020, for about 2 years, schools and colleges were shut in India (and also in many parts of the USA). For the school/college student age-group, what is the relative risk of Covid death versus traffic accident death? Answer in terms of a number (your best guess). E.g. 100 times higher risk of Covid death compared to traffic accident death.

Risk of Covid death $< \frac{1}{10}$ Risk of traffic accident death as per official flawed definition of Covid death.

Risk of Covid death = 0 for that age-group if excess deaths considered. That is, Covid was not risk-additive at all.