1  Activity 1 - Data Hazards

Consider the following instruction sequence:

```
add  $t0, $s1,$s2
beq  $t0,$zero, LABEL
sw   $t4, 0($t0)
```

Determine what each instruction is doing on each clock cycle. For each instruction, determine:

- When it reads its input
- When it calculates a value
- When it stores the result back
- When it determines what the next instruction will be

2  Activity 2 - Forwarding

Simulate the following sequence of instructions. Do **NOT** insert any stalls. For the EX phase of each instruction, determine:

- Were the values read in ID correct?
- If not, what pipeline register currently contains the not-yet-written values?

Draw a pipeline diagram, showing the dependencies:

```
sub   $2,  $1,$3
and   $12, $2,$5
add   $13, $2,$2
or    $14, $5,$2
sw    $15, 100($2)
```
3 Activity 3 - Data Forwarding Logic

Suppose that you can read the following control fields:

- **id_ex.rs** - The rs register # (not value) of the instruction currently running the EX phase.
- **ex_mem.regWrite** - The regWrite control bit of the instruction currently running the MEM phase.
- **ex_mem.writeReg** - The register that will be written to, by the instruction currently running the MEM phase.
- **mem_wb.regWrite** - The regWrite control bit of the instruction currently running the WB phase.
- **mem_wb.writeReg** - The register that will be written to, by the instruction currently running the WB phase.
- **mem_wb.memToReg** - The memToReg control bit of the instruction currently running the WB phase.

(You may assume that the instruction in MEM is **not** a LW instruction. But note that the instruction in WB might have been LW.)

Write pseudocode to determine what value should be given to input 1 of the ALU. There are three options:

- **id_ex.rsVal** - the rs register value that was read back in the ID phase.
- **ex_mem.aluResult** - the result of the ALU, for the instruction currently running the MEM phase.
- **mem_wb.aluResult** - the result of the ALU, for the instruction currently running the WB phase.
- **mem_wb.memResult** - the result of the memory read, for the instruction currently running the WB phase.