Essentials of Computer Systems - Exercises #4

1 Performance Equations

Execution time $= T = IC \times t_{\text{CLK}} \times CPI = \frac{\#insn}{program} \cdot \frac{\#seconds}{cycle} \cdot \frac{\#cycpes}{insn}$
Performance $= \frac{1}{T}$

Speedup= $\frac{Told}{Tnew}$

Amdahl's law: the total speedup depends on

- fraction of X being used in original system $\Rightarrow f$
- enhancement/speedup of X $\Rightarrow S$

$S_{\text{total}} = \frac{1}{\frac{f}{S}+(1-f)}$

Exercise 1.1 *Average CPI*

When running a given benchmark we found the frequency and CPIs of instructions given in table 1. Afterwards, two separate modifications were made and benchmarking repeated:

a) hardware modifications resulting in changed CPIs of the instructions (see table 2)

b) modifications to the compiler resulting in changed frequencies if the instructions (see table 3)

Compare all three options and identify the best one!

Table 1

insn	f_i	CPI_i
ALU	*50%*	*1*
load	*20%*	*5*
store	*10%*	*3*
other	*20%*	*2*

Table 2

insn	f_i	CPI_i
ALU	*50%*	*1*
load	*20%*	*4*
store	*10%*	*4*
other	*20%*	*2*

Table 3

insn	f_i	CPI_i
ALU	*55%*	*1*
load	*10%*	*5*
store	*10%*	*3*
other	*25%*	*2*

Exercise 1.2 *Amdahl's law*

The old CPU takes 5 seconds to execute a given program. A new CPU with four times faster adder[1] *is available. The adder is used in 25% of the instructions in the given program. Compute the new execution time for the given program when using the new CPU.*

[1] adder is a part of the ALU

2 Execution on the Pipelined CPU

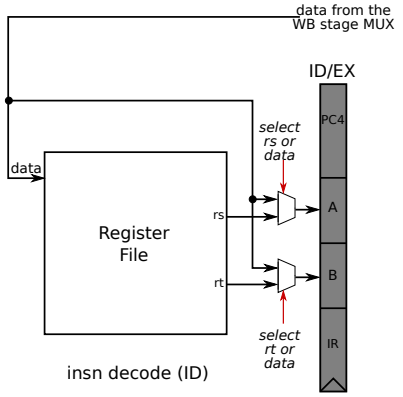
Exercise 2.1 *For the following code snippet show:*

a) analysis of data dependencies. Identify the data hazards.

b) show the timing diagram of execution on pipelined CPU without bypassing logic (see lecture slides)

c) show the timing diagram of execution on pipelined CPU with bypassing logic from EX/MEM and MEM/WB interstage registers

d) show the timing diagram of execution on pipelined CPU with bypassing logic from EX/MEM and MEM/WB interstage registers and with RegFile bypassing. RegFile bypassing allows the "overlap" of ID and WB stage by adding MUXes on the inputs to the interstage registers ID/EX.A and ID/EX.B, allowing to select either the RegFile outputs rs and rt or the data value from WB stage - please see schematic below.



recap:

insn syntax	*operation*
and Rd, Rs, Rt	*Rd ← Rs AND Rt*
or Rd, Rs, Rt	*Rd ← Rs OR Rt*
lw Rt, offset(Rs)	*Rt ← MEM[Rs+offset]*

code snippet:

```
lw R3, 0(R1)
and R4, R3, R2
or R5, R3, R2
and R3, R4, R5
```