Ahram Kim 114055134

Homework #2-3

Problem 1

4.18 Assume that x11 is initialized to 11 and x12 is initialized to 22. Suppose you executed the code below on a version of the pipeline from Section 4.5 that does not handle data hazards (i.e., the programmer is responsible for addressing data hazards by inserting NOP instructions where necessary). What would the final values of registers x13 and x14 be?

```
addi x11, x12, 5   // x11 = 27
add x13, x11, x12  // x13 = 11+22=33
addi x14, x11, 15  // x14 =11+15=26
```

x13=33
x14=26

Problem 2

4.20 Add NOP instructions to the code below so that it will run correctly on a pipeline that does not handle data hazards.

```
addi x11, x12, 5
add  x13, x11, x12
addi x14, x11, 15
add  x15, x13, x12
```

```
addi x11, x12, 5
NOP
NOP
add x13, x11, x12
addi x14, x11, 15
NOP
add x15, x13, x12
```

Problem 3

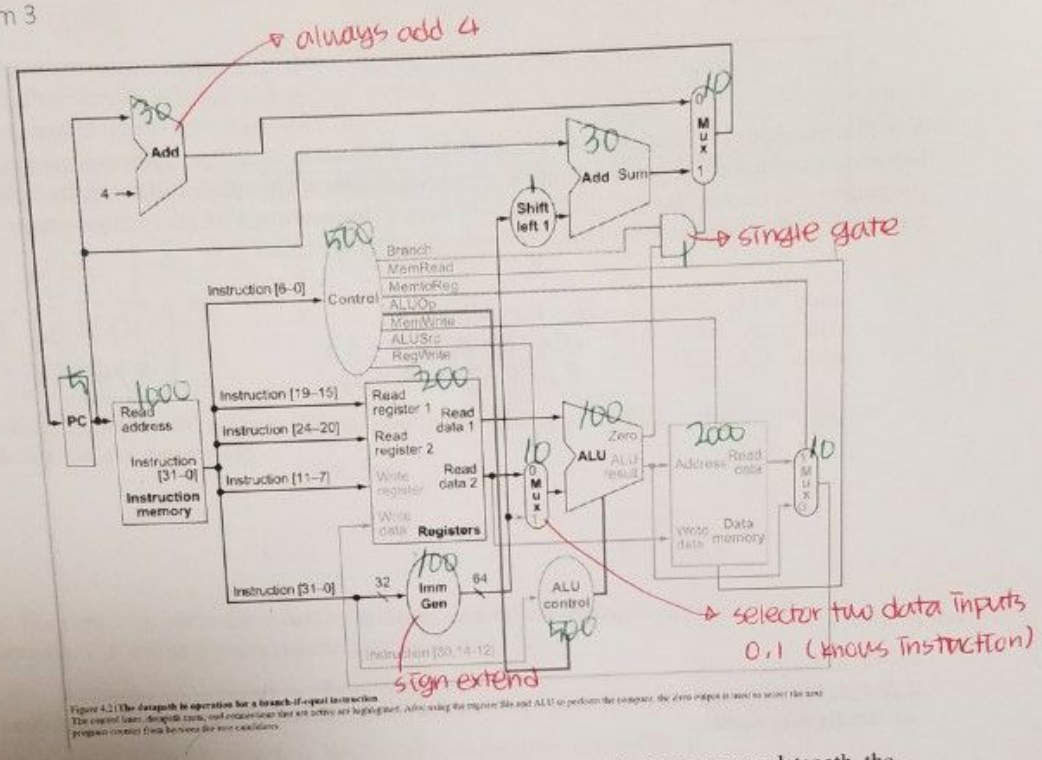

Figure 4.21 The datapath in operation for a branch-if-equal instruction. The control lines, the logic unit, and the new adder are highlighted. After using the register file and ALU to perform the compare, the Zero output is used to select the next program counter from between the two candidates.

4.10 When processor designers consider a possible improvement to the processor datapath, the decision usually depends on the cost/performance trade-off. In the following three problems, assume that we are beginning with the datapath from Figure 4.21, the latencies from Exercise 4.7, and the following costs:

R-type/I-type (non-ld)	ld	sd	beq
52%	25%	11%	12%

I-Mem	Register File	Mux	ALU	Adder	D-Mem	Single Register	Sign extend	Single gate	Control
1000	200	10	100	30	2000	5	100	1	500

Suppose doubling the number of general purpose registers from 32 to 64 would reduce the number of ld and sd instruction by 12% but increase the latency of the register file from 150 ps to 160 ps and double the cost from 200 to 400. (Use the instruction mix from Exercise 4.8 and ignore the other effects on the ISA discussed in Exercise 2.18.)

4.10.1 What is the speedup achieved by adding this improvement?

ld_instr = 0.25 , sd_instr = 0.11

- reduce 12% , reduction = 0.12 × (0.25+0.11) = 0.0432
- origin time = 930
- new time = 930 + 10 = 940) (930+10)n → 940n

1 − Reduction = 1 − 0.0432 = 0.9568

940 × (1 − 0.0432) = 899.392

$\frac{940}{899.392} = 1.045$ → increase 4.5%

- speedup = $\frac{930}{899.392} = 1.034$ → performance increase in 3.4%

4.10.2 Compare the change in performance to the change in cost.

The cost of original CPU is

The cost of improved CPU is

↳ original CPU + CPU

∴ 3.4% increase in performance.

the cost of CPU increase by about 4.4%

$\frac{450n}{450n} = 1.0443$ → cost increase in 4.4%

4.10.3 Given the cost/performance ratios you just calculated, describe a situation where it makes sense to add more registers and describe a situation where it doesn't make sense to add more registers.

From a strictly mathematical standpoint it does not make sense to add more registers because the new CPU costs more per unit of performance. However, that simple calculation does not account for the utility of the performance. For example. In a real time system, a 3% performance may make the difference between meeting or missing deadlines. In which case, the important would be well worth the 4.4% additional cost.