<u>12.14.5</u>
In the exercise, we examine in detail how an instruction is executed in a single-cycle datapath. Problems in this exercise refer to a clock cycle in which the processor fetches the following instruction word: 0xadac0014.

For context: The encoded instruction is `sw $t4,20($t5)`

a) What are the values of the ALU control unit's inputs for this instruction?

0xadac0014 in binary: 101011 01101 01100 0000000000010100
Alu Op (Last 2 bits): 00
Instructions (Last 6 bits): 010100

b) What is the new PC address after this instruction is executed? Highlight the path through which this value is determined.

PC + 4

c) For each mux, show the values of its inputs and outputs during the execution of this instruction. List values that are register outputs at `Reg [xn]`

101011	01101	01100	0000000000010100
Instruction	rs	rt	address

Register Mux:
Input/Output: **01100**(rt)
Alu Mux:
Input/Output: sign extension of (15:0) **00000000000000000000000000010100**
Read Data 2: $t4
Mem to Reg Mux:
Input: **read data Reg[$t5]**
Output: **undefined**
PCSrc Mux:
Input/Output: **PC + 4**

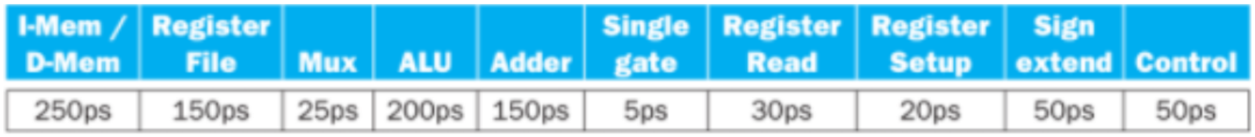
d) What are the input values for the ALU and the two add units?

ALU:
input: base address of $t5
Add PC:
input: PC + 4
after instruction [15:0] is shifted left by 2 -> 20 sll 2 = 80

e) What are the values of all inputs for the registered unit?

Read register 1: 01101
Read register 2: 01100
Write register: 0 don't use
Write data: don't use
RegWrite: 0 don't use

<u>**12.14.7**</u>



I-Mem / D-Mem	Register File	Mux	ALU	Adder	Single gate	Register Read	Register Setup	Sign extend	Control
250ps	150ps	25ps	200ps	150ps	5ps	30ps	20ps	50ps	50ps

"Register read" is the time needed after the rising clock edge for the new register value to appear on the output. This value applies to the PC only. "Register setup" is the amount of time a register's data input must be stable before the rising edge of the clock. This value applies to both the PC and Register File.

a) What is the latency of an R-type instruction(i.e., how long must the clock period be to ensure that this instruction works correctly)?

R-type latency: I-mem + Register File + Alu + Add + Mux * 3 + Register read + Single gate + Register set up + Control
= 250 + 150 + 200 + 150 + 25*3 + 30 + 5 + 20 + 50 = **930**

b) What is the latency of lw? (Check your answer carefully. Many students place extra muxes on the critical path.

LW Latency: I-mem + Register File + Alu + D-mem + Mux *2 + Register read + Register set up
= 250 + 150 + 200 + 250 + 50 + 30 + 20 = **950**

c) What is the latency of sw? (Check your answer carefully. Many students place extra muxes on the critical path.)

SW Latency: I-mem + D-mem + Register read + Register File + Alu + Mux
= 250 + 250 + 30 + 150 + 200 + 25 = **905**

d) What is the latency of beq?

Beq Latency: I-mem + Register read + Register File + Alu + Mux*2 + single gate + Register set up
= 250 + 150 + 30 + 200 + 50 + 5 + 20 = **705**

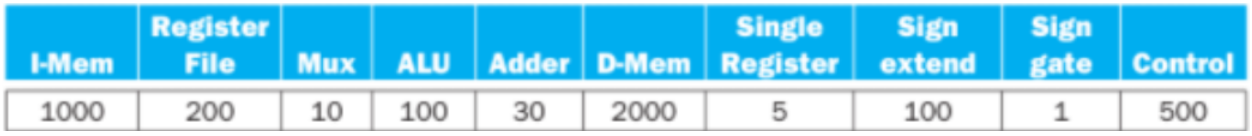
e) What is the latency of an arithmetic, logical, or shift I-type (non-load) instruction?

Arithmetic/logic/shift I-type Latency: I-mem + Register read + Register File + Alu + Mux*2 + Register set up
= 250 + 150 + 30 + 200 + 50 + 20 = **700**

f) What is the minimum clock period for this CPU?

Max latency = minimum clock period = **950 (LW latency)**

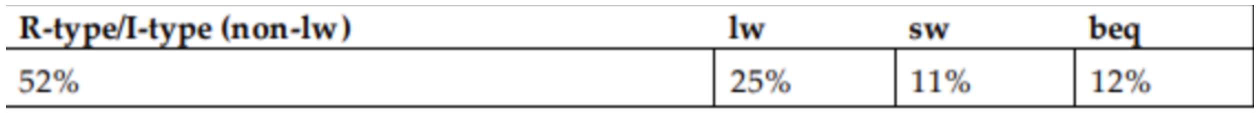
<u>**12.14.10**</u>
When the processor designers consider a possible improvement to the processor datapath, the decision usually depends on the cost/performance trade-off. In the following three problems, assume that we are beginning with the datapath from COD figure 4.21, the latencies from Exercise 4.7, and the following costs:



I-Mem	Register File	Mux	ALU	Adder	D-Mem	Single Register	Sign extend	Sign 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 `lw` and `sw` instructions 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.)

a) What is the speedup achieved by adding this improvement?



R-type/I-type (non-lw)	lw	sw	beq
52%	25%	11%	12%

Reduction: (.25 + .11) * .12 = 4.32%
Increase latency from 150 to 160 so new performance is 950 + 10 = 960
After Reduction Performance: 960 * (1 - .0432) = 918.528
Speedup: 950 . 918.528 = 1.034 = **3.4%** faster

b) Compare the change in performance to the change in cost.

OG cost: I-mem + PC + Register file + D-mem + Control * 2+ ALU + D-mem + Sign Extend + Adder*2 + Mux*3 + gate
= 1000 + 200 + 5 + 1000 + 100 + 2000 + 100 + 60 + 30 + 1 = 4496

Increase cost from 200 to 400: 4496 + 200 = 4696
4696/4496 = 1.044 = 4.4% change in cost

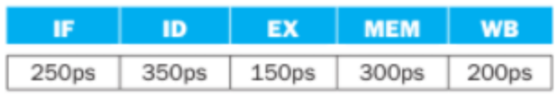
change in performance is 3.4% and increase in cost is 4.4%

c) 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 our calculations. by adding more registers we increased our performance by 3.4% but increased our cost by 4.4%. In a scenario where we want more performance and we don't care about our cost, adding more registers is the way to go. Vice versa.

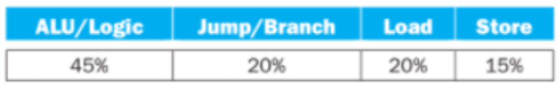
12.14.16

In this exercise, we examine how pipelining affects the clock cycle time of the processor. Problems in this exercise assume that individual stages of the datapath have the following latencies:



IF	ID	EX	MEM	WB
250ps	350ps	150ps	300ps	200ps

Also, assume that instructions executed by the processor are broken down as follows:



ALU/Logic	Jump/Branch	Load	Store
45%	20%	20%	15%

a) What is the clock cycle time in a pipelined and non-pipelined processor?

Pipelined (highest latency): 350ps
Non-Pipelined: IF + ID + EX + MEM + WB
= 250 + 350 + 150 + 300 + 200 = **1250ps**

b) What is the total latency of an lw instruction in a pipelined and non-pipelined processor?

Pipelined (highest latency * 5) = 350 * 5 = **1750ps**
Non-Pipelined: IF + ID + EX + MEM + WB
= 250 + 350 + 150 + 300 + 200 = **1250ps**

c) If we can split one stage of the pipelined datapath into two new stages, each with half the latency of the original stage, which stage would you split and what is the new clock cycle time of the processor?

I would split the longest stage which will be the ID stage. The new split of the pipelined datapath will now be MEM or 300ps.

d) Assuming there are no stalls or hazards, what is the utilization of the data memory?

Load + Store = 35%

e) Assuming there are no stalls or hazards, what is the utilization of the write-register port of the "Registers" unit?

ALU/Logic + Jump/Branch = 65%