Homework 1

Question 1

Problems in this exercise assume that the logic blocks used to implement a processor's datapath have the following latencies:

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.

Please answer the following questions:

1. 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)?
2. What is the latency of **ld**? (Check your answer carefully. Many students place extra muxes on the critical path.)
3. What is the latency of **sd**? (Check your answer carefully. Many students place extra muxes on the critical path.)
4. What is the latency of **beq**?
5. What is the latency of an I-type instruction?
6. What is the minimum clock period for this CPU?

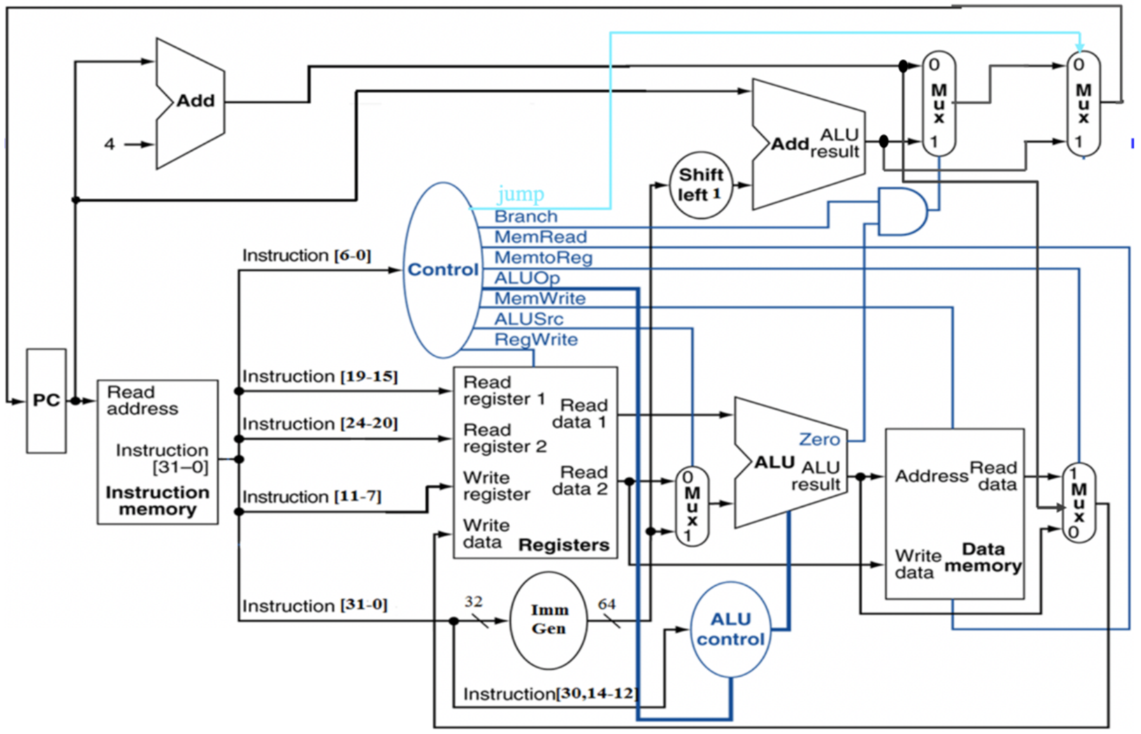
Hint

For example, what is the latency of **add** instruction?

1. First, we need to fetch the instruction from I-Mem with correct pc. This costs 30 + 250 ps.
2. Second, we need to decode the instruction. Because controller, register file and sign-extend module can work simultaneously, this costs max(50, 150, 50) ps.
3. Third, we need ALU to calculate. This costs 25 + 200 ps. (Here 25 represents a Mux deciding ALU_SrcB is immediate or not)
4. Last, we need to write the ALU result back to register file. This costs 25 + 20 ps. (Here 25 is also a Mux and 20 is Register Setup time)

So the latency of **add** instruction is 700 ps in total.

The CPU involved in the question can be considered with reference to the picture below, which is found in your Sys1 Lab10 experimental manual.



Question 2

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%

Please answer the following questions:

1. What is the clock cycle time in a pipelined and non-pipelined processor?
2. What is the total latency of an **ld** instruction in a pipelined and non-pipelined processor?
3. 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?
4. Assuming there are no stalls or hazards, what is the utilization of the data memory?
5. Assuming there are no stalls or hazards, what is the utilization of the write-register port of the "Registers" unit?

Hint

Utilization means quotient of clock cycles when this module is in use and clock cycles in total.

Question 3

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
```

Question 4

If we change load/store instructions to use a register (without an offset) as the address, these instructions no longer need to use the ALU. As a result, the MEM and EX stages can be overlapped and the pipeline has only four stages.

Please answer the following questions:

1. How will the reduction in pipeline depth affect the cycle time?
2. How might this change improve the performance of the pipeline?
3. How might this change degrade the performance of the pipeline?

Hint

Performance is related to both the number of instructions and the execution speed of each instruction.