Important Equations for Pipelining:

Speed Up:

$$S_n = \frac{Execution\ time_{old}}{Execution\ time_{new}} = \frac{nt_n}{(k+n-1)t_p}$$

Efficiency:

$$E = \frac{n}{(k+n-1)} = \frac{S_n}{(t_n / t_p)}$$

Where n is the number of tasks, t_n is the time to complete each task on a non-pipeline unit, k is the number of stages for each task, and t_p is the time period to complete a k-stage pipeline.

Problem 1:

Consider a machine where each task passes by 4 stages in order to be executed, each of length 20 ns, If 100 tasks to be executed:

a) On a non-pipelined machine, what is the execution time for the tasks on this machine?
b) On a 4-stage pipeline machine, what is the execution time for the tasks on this machine?
c) What is the speed up gained from pipelining?

Problem 2:

Consider a machine where each instruction passes by 5 stages, the 5 stages needs 10, 8, 10, 10, 7 ns respectively. If a pipeline is applied to the machine, it adds a latch of 1 ns between each stage and the other. If 50 tasks to be executed:

a) On a non-pipelined machine, what is the execution time?
b) On a 5-stage pipeline machine with a latch, what is the execution time?
c) What is the speed up gained from pipelining?

Problem 3:

Consider a non pipelined machine with 6 stages of lengths 50 ns, 50 ns, 60 ns, 60 ns, 50 ns, and 50 ns.

a) What is the execution time for an instruction on this machine?
b) How much time does it take to execute 100 instructions?

Suppose we introduce pipelining on this machine. Assume that when introducing pipelining, the clock adds 5ns latch to each execution stage.

c) What is the execution time for an instruction on this machine?
d) How much time does it take to execute 100 instructions?
e) What is the speedup obtained from pipelining for 100 instructions?

Problem 4:

Consider the following snippet of code:

```
FOR i = 1 TO 100 DO
{
        A[i] = (B[i]xC[i]) + D[i]
}
```

Assume that each operation, multiplication and addition, requires 10 ns to complete. Consider a pipelined unit that could break this computation into two stages, the first stage performs the multiplication and the second stage performs the addition.

a) What is the total execution time for the code before pipeline?
b) What is the total execution time for the code after pipeline if no latch is added?
c) What is the total execution time for the code after pipeline if a latch of 2ns is added between stages?
d) Calculate the speedup gained from pipeline with latch.

Problem 5:

Assume that the individual stages of on a machine takes the following execution time:

IF	ID	EX	MEM	WB
300ps	400ps	350ps	500ps	100ps

Assume that when pipelining, each pipeline stage costs 20ps extra between pipeline stages.

a) What is the clock cycle time in a pipelined and non-pipelined processor?
b) If you could split one of the pipeline stages into 2 equal halves, which one would you choose? What is the new cycle time?

Problem 6:

Consider a machine where each instruction passes by 4 stages, fetch (IF), decode(ID), execute(EX), and write back (WB).

a) Fill table 1 to show 3 instructions execution on a non-pipelined machine.
b) Fill table 2 to show 3 instructions execution on a pipelined machine.

Table 1

Inst1															
Instr2															
Instr3															

Table 2

Inst1															
Instr2															
Instr3															