Written Homework Assignment #1

1. Identify the control (or branch) hazard and modify the code to eliminate the branch hazard assuming the branch decision is made in ID stage.

```
...
beq   $12, $0, L1
add   $5, $0, 1
beq   $12, $5, L2
add   $15, $14, $2

L1:    add $12, $12, -1
       add $5, $6, $5
L2:    or $3, $4, $12
...
```

Solution:

```
…
beq   $12, $0, L1
nop
add   $5, $0, 1
beq   $12, $5, L2
nop
add   $15, $14, $2

L1:    add $12, $12, -1
       add $5, $6, $5
L2:    or $3, $4, $12
...
```

2. Calculate how many clock cycles are needed to execute the instructions below with and without pipeline? You can assume the following:

1) Instruction Fetch take 2 ns
2) Instruction Decode 1ns
3) ALU 2ns
4) Memory Access 2ns
5) Write Register 1ns

```
add $1, $6, $11
lw   $2,  8($12)
lw   $3,  8($13)
and $4,  $9, $14
or   $5, $10, $15
```

Solution:

Without Pipeline:
3 R-type instructions, therefore need to use all stages except memory access stage:
(2 + 1 + 2 + 1) x 3 = 18 ns
2 I-type instructions, only need to use all stages:
(2 + 1 + 2 + 2 + 1) x 2 = 16 ns

Total of **34 ns**

With Pipeline:
The longest time taking to execute a stage among all the instruction stages are 2n. Therefore, the clock period will be 2 ns.

Below shows how each clock cycle looks like:

	1	2	3	4	5	6	7	8	9
add $1, $6, $11	IF	ID	EX	MA	WB				
lw $2, 8($12)		IF	ID	EX	MA	WB			
lw $3, 8($13)			IF	ID	EX	MA	WB		
and $4, $9, $14				IF	ID	EX	MA	WB	
or $5, $10, $15					IF	ID	EX	MA	WB

Total of 9 cycles, therefore the total time needed to execute all 5 instructions is:

$$\frac{2ns}{cycle} \times 9cycles = 18ns$$

3. How can the structural hazard on register file be eliminated? What advantage can we take from the clock?

Solution:

Structural hazard on register file means that there is a conflict in register file access in a certain clock cycle.
An example of it could be: An instruction is trying to write back to the register file during its write back stage while another instruction is in its instruction decode stage trying to read from the register file.

There are 2 ways to eliminate the structural hazard:
1. Instruction scheduling: you can optimally rearrange your instructions to avoid having structural hazard.
2. Hardware hazard control unit: To prevent the conflict we can perform write back during the first half of the clock cycle and read during the second half of the clock cycle.