Problem 1) Consider the following code:

```
     addi R10, R0, 100  // initialize to 100
TOP:
     lw R1, 0(R2)
     lw R2, 0(R3)
     mul R5, R1, R2
     lw R6, 0(R7)
     add R8, R5, R6
     sw R8, 0(R9)
     addi R2, R2, 4
     addi R4, R4, 4
     addi R7, R7, 4
     addi R9, R9, 4
     addi R10, R10, -1
     BNE R10, R0, TOP
```

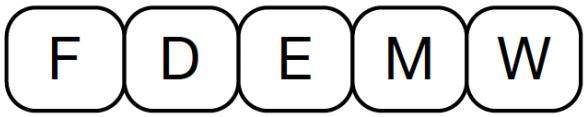
Assume the following latencies for a pipelined, in-order processor:

Instruction	Latency
add/addi	1
BNE (assume perfect prediction)	1
lw	4
mul	6

i. Show for one iteration of the loop (including the initial `addi`, in which cycle each instruction will be issued). Start numbers with cycle 1.

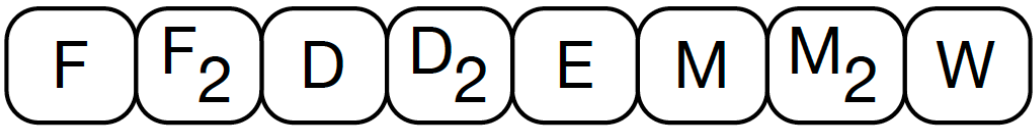
ii. How many cycles does the entire loop take to be issued?

iii. Rearrange the instructions to minimized the number of cycles required to execute the loop and give the new total number of cycles required.



Problem 2. i. Consider the RISC 5-stage pipeline above. Show a timeline that indicates which cycle each instruction will be in each of the stages as they move through the pipeline. Assume forwarding paths are available whenever forwarding is possible. Indicate forwarding whenever it occurs.

```
lw R1, 100(R2)
lw R3, 0(R1)
add R5, R3, R5
sub R6, R4, R1
or R7, R5, R6
```



ii. Consider the above pipeline. Assume that 3 additional states were added to 5 stage pipeline. Assume that branch outcomes are determined in the stage labeled M (and PC is updated in the M stage for mispredicted branches). Assume that memory accesses are not complete until the M_2 stage.

How many instructions have to be flushed when a branch is mispredicted?

iii. Assume that for the pipeline from part ii., 15% of all instructions are branches and 96% of those branches are correctly predicted. Assume there are no other hazards. Calculate average CPI.

iv.. Again for the pipeline for part ii., assume that 22% of all instructions are lw's (load word's). Assume the following distribution of instruction dependencies for load words:

lw dependencies	**percentage of load words**
next instruction depends on lw	33%
next instruction is independent of lw BUT 2nd instruction following lw depends on that lw	45%
next instruction and 2nd instruction are both independent of lw BUT 3rd instruction following lw depends on that lw	10%
no nearby instructions depend on load word	12%

Assume there are no other hazards. Calculate average CPI.

Problem 3. Consider a hypothetical single-cycle datapath that has a minimum clock period of 2500 ps. Assume that datapath can be broken into stages of exactly equal latency by pipelining using latches with an additional latency of 37 ps. Assume also that the mispredicted branch penalty a minimum of 1000 ps. Assume that 20% of instructions are branches. That 5% of branches are mispredicted. Ignore all other hazards

i. Fill in the table for the maximum frequency and CPI for the pipelined datapath with the following number of stages:

# stages	frequency	CPI
5		
18		
100		

(show your calculations)

ii. For the assumptions above, what is the speedup for a pipeline with 18 stages over the performance of a single-cycle datapath?

Pipeline review

Problem 4. Consider the following MIPS assembly language code fragment:

```
lw $1, 0($2)
addi $1, $1, 1
sw $1, 0($2)
addi $2, $2, 4
sub $4, $3, $2
```

Assume a 5 stage RISC pipeline. Assume all memory accesses are cache hits.

a. Show the timing of this instruction sequence for a pipeline *without* any forwarding or bypassing. Assume that a register write to and register read from the same register can occur within a clock cycle.

b. Show the timing of this instruction sequence for a pipeline with normal forwarding and bypassing.

Problem 5. Consider the following MIPS assembly language code fragment:

```
Loop:
   lw $10, 0($2)
   lw $4, 0($3)
   mult $10, $10, $4
   add $2, $10, $2
   addi $2, $2, 8
   addi $3, $3, 8
   sub $5, $4, $2
   bne $5, $0, Loop
```

Assume the 5 stage RISC pipeline. Assume the memory accesses are cache hits.

a. Show the timing of this instruction sequence for a pipeline *without* any forwarding or bypassing. Assume that a register write to and register read from the same register can occur within a clock cycle. Assume the `bne` is predicted not taken and is actually not taken.

b. Show the timing of this instruction sequence for a pipeline with normal forwarding and bypassing. Assume the `bne` is predicted not taken and is actually not taken.

c. Show the timing of this instruction sequence for a pipeline with normal forwarding and bypassing. Assume the `bne` is predicted taken but is actually not taken. Assume the branch misprediction is detected and the PC is corrected in MEM.