Consider the following code segment, with the values MAX = 10^6 and WSIZE = 5 and the following data:

- Read from x and w takes 20 cycles
- Multiply takes 10 cycles
- t+= takes 5 cycles
- Write to y takes 10 cycles
- Each i++ and j++ take 5 cycles
- Each comparison takes 5 cycles
- Write to a register takes 1 cycle.
- L1 cache hits require 5 cycles
- Cache miss penalty is 10 cycles

```
for( i = 0; i < MAX; ++i ){
        t = 0;
        for( j = 0; j < WSIZE; ++j ){
                t += x[i+j]*w[j];
                y[i] = t;
        }
}
```

a) Without any modifications, how many cycles does the code segment take?
Total loops = WSIZE* MAX = $5*10^6$, i++ cycles = (5+5) * $1*10^6$ = $10*10^6$, j++ cycles = (5+5) * $5*10^6$ = $50*10^6$

b) Discuss the opportunities for reduction of cycle times and determine the improvement in performance when implemented in SIMD with 8 processors.

Improvement: $3.35*10^8$ / $1.3125*10^7$ = ~25.52x

Given the following RISC-V Code and the following set of latencies with dependencies, and a branch delay slot of 1 cycle

Type	Instruction producing result	Instruction using result	Stalls
1	FP ALU operation	Another FP Operation	3
2	FP ALU operation	Store double	2
3	Load double	FP ALU operation	1
4	Load double	Store double	1
5	Integer operation	Integer operation	1

```
Loop:
        LD    F0, 0(R1)
        ADDD  F4,F0,F2
        SD    0(R1), F4
        ADDI  R1,R1,-8
        BNEZ  R1,Loop
        NOP   ;delayed branch slot
```

First, determine the stalls require to implement the code. Then, perform out-or-order execution to improve the run time.

Consider the following RISC-V code

Show the execution of a single issue Tomasulo algorithm in the form of status tables for the above code sequence for one iterations of the above loop.

```
LD      F6,   40(R2)
LD      F2,   48(R3)
MULD    F0,   F2,   F4
SUBD    F8,   F6,   F2
DIVD    F10,  F0,   F6
ADDD    F6,   F8,   F2
```

Assume the following

- There are 1 FP ADD/SUB unit, 1 FP MULT units, 2 load buffers and 2 store buffers
- FP ADD/SUB take 2 cycles, multiplication takes 10 cycles, divides take 40 cycles and load/stores take 2 cycles for execution
- Function units are not pipelined and there is no forwarding between function units; results are communicated by the CDB
- If two instructions complete at the same time one of them will wait to write result on CDB

(Note, on an exam, the number of empty slots do not necessarily have to match the result

Iter #	Instructions	Issues	Execute	Write Result	Comment

Consider the following RISC-V code

```
LD      F2,   0(R1)
MULTD   F4,   F0,   F2
SD      F4,   0(R1)
SUBI    R1,   R1,   #8
BNE     R1,   R0,   Loop
```

Show the execution of a single issue Tomasulo algorithm in the form of status tables for the above code sequence for two iterations of the above loop.

Assume the following

- There are 1 FP ADD/SUB unit, 2 FP MULT units, 2 load buffers and 2 store buffers
- FP ADD/SUB take 2 cycles, multiplication takes 4 cycles, divides take 16 cycles and load/stores take 1 cycle for execution
- First load takes 8 clocks (L1 cache miss), all other lw/sw takes 1 clock (hit)
- Function units are not pipelined and there is no forwarding between function units; results are communicated by the CDB
- SUBI, BEQZ are integer instructions that each take 1 clock cycle to execute
- If two instructions complete at the same time one of them will wait to write result on CDB

Iter #	Instructions	Issues	Execute	Write Result	Comment