Please, configure the winMIPS64 simulator with the *Base Configuration* provided in the following:

- Code address bus: 12
- Data address bus: 12
- Pipelined FP arithmetic unit (latency): 4 stages
- Pipelined multiplier unit (latency): 8 stages
- divider unit (latency): not pipelined unit, 12 clock cycles
- Forwarding is enabled
- Branch prediction is disabled
- Branch delay slot is disabled
- *Integer ALU: 1 clock cycle*
- *Data memory: 1 clock cycle*
- *Branch delay slot: 1 clock cycle.*

1) Starting from the assembly program you created in the previous lab called **program_2.s**,:

```
for (i = 0; i < 40; i++){
        v5[i] = v1[i]+(v2[i] * v3[i]);
        v6[i] = v5[i]*v4[i];
        v7[i] = v6[i]/v2[i];
}
```

a. Detect manually the different data, structural and control hazards that provoke a pipeline stall

b. Optimize the program by re-scheduling the program instructions in order to eliminate as much hazards as possible. Compute manually the number of clock cycles the new program (<mark>**program_2_a.s**</mark>) requires to execute, and compare the obtained results with the ones obtained by the simulator.

c. Starting from **program_2_a.s**, enable the *branch delay slot* and re-schedule some instructions in order to improve the previous program execution time. Compute manually the number of clock cycles the new program (<mark>**program_2_b.s**</mark>) requires to execute, and compare the obtained results with the ones obtained by the simulator.

d. Unroll 4 times the program (**program_2_b.s**), if necessary re-schedule some instructions and renaming the used registers. Compute manually the number of clock cycles the new program (<mark>**program_2_c.s**</mark>) requires to

execute, and compare the obtained results with the ones obtained by the simulator.

Complete the following table with the obtained results:

Program / Clock cycle computation	program_2.s	program_2_a.s	program_2_b.s	program_2_c.s
By hand	1689	1489	1449	899
By simulation	1688	1568	1529	888

Compare the results obtained in the point 1, and provide some explanation in the case the results are different.

Eventual explanation:

Una possibile spiegazione dei diversi risultati ottenuti "by hand" e "by simulation" è che l'emulatore in alcuni casi anticipa l'inizio di alcune istruzioni, come nel caso dell'istruzione "add.d", per poi andare in stallo, permettendo alle istruzioni successive (come ad esempio la "mul.d") di procede senza entrare in stallo, cosa che nel calcolo "by hand" non succede.