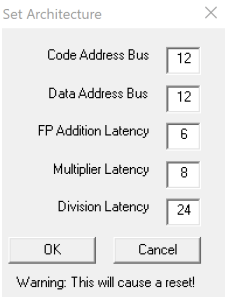
Architetture dei Sistemi di Elaborazione 02GOLOV [GRB-ZZZ]	Delivery date: 1st November 2022
Laboratory 3	Expected delivery of lab_03.zip must include: - `program_1_a.s`, `program_1_b.s` and `program_1_c.s` - this file compiled and if possible in pdf format.

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): 6 stages
- Pipelined multiplier unit (latency): 8 stages
- Divider unit (latency): not pipelined unit, 24 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_1.s`**:

```
for (i = 0; i < 60; i++){
        v5[i] = ((v1[i]+v2[i]) * v3[i])+v4[i];
        v6[i] = v5[i]/(v4[i]*v1[i]);
        v7[i] = v6[i]*(v2[i]+v3[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 many hazards as possible. Compute manually the number of clock cycles the new program (**`program_1_a.s`**) requires to execute, and compare the obtained results with the ones obtained by the simulator.

   c. Starting from **`program_1_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 (**`program_1_b.s`**) requires to execute, and compare the obtained results with the ones obtained by the simulator.

   d. Unroll 3 times the program (**`program_1_b.s`**), if necessary re-schedule some instructions and increase the number of used registers. Compute

manually the number of clock cycles the new program (**`program_1_c.s`**) 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_1.s`	`program_1_a.s`	`program_1_b.s`	`program_1_c.s`
`By hand`	4156	3855	3797	3738
`By simulation`	4087	3791	3736	3696

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

Eventual explanation:

Osservando la pipeline prodotta da WinMips64, è possibile notare come, nel caso di hazard di dato, WinMips64 passi comunque alla successiva fase di processamento dell'istruzione per poi andare in stallo in attesa del dato. Analizzando più approfonditamente questa soluzione, sebbene sia più rischiosa dal punto di vista dell'accesso al dato, consente effettivamente di risparmiare un clock cycle per tutte le istruzioni a seguire in ogni iterazione del loop. Per questo motivo il calcolo dei clock cycle "by hand" fornisce un risultato maggiore rispetto a quello "by simulation" per ognuno dei tre casi.

Formatted: No underline, Font colour: Background 1
Formatted: No underline, Font colour: Background 1
Formatted: No underline, Font colour: Background 1
Formatted: No underline, Font colour: Background 1
Formatted: Centred
Formatted: No underline, Font colour: Background 1
Formatted: No underline, Font colour: Background 1
Formatted: No underline, Font colour: Background 1
Formatted: No underline, Font colour: Background 1
Formatted: Centred
Formatted: Italian
Formatted: Italian
Formatted: No underline
Formatted: No underline
Formatted: No underline
Formatted: No underline
Formatted: No underline
Formatted: No underline, Italian
Formatted: Italian