Architetture dei Sistemi di Elaborazione	Delivery date: October 29th 2021
Laboratory 3	Expected delivery of lab_03.zip must include: - `program_2_a.s`, `program_2_b.s` and `program_2_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): 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 (**`program_2_a.s`**) 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 (**`program_2_b.s`**) 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 (**`program_2_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_2.s`	`program_2_a.s`	`program_2_b.s`	`program_2_c.s`
`By hand`	1728	1608	1488	1208
`By simulation`	1647	1607	1488	938

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

Eventual explanation:

program_2.s ha risultati diversi by hand e by simulation perché le istruzioni non erano schedulate bene e a mano si erano 2 stalli che il compilatore non vede perché reschedula quello che può.
Si nota che con una reschedulizzazione semplice, il numero di cicli di program_2_a.s si riduce già di un po'. Se utilizziamo il delay slot inserendo un'istruzione sotto l'istruzione di salto si riduce ancora di più il numero di cicli perché sostituisco un'istruzione NOP con un istruzione utile. Reschedulando nel punto c dopo aver fatto unroll il numero di cicli diminuisce drasticamente perché si è ridotta di minimo la dipendenza tra i dati.