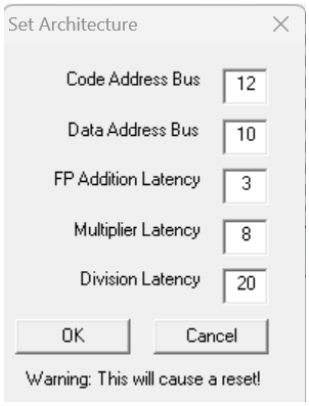
Architetture dei Sistemi di Elaborazione 02GOLOV	Delivery date: **2st November 2023**
Laboratory 3	Expected delivery of lab_03.zip must include: - `program_1_a.s`, `program_1_b.s` and `program_1_c.s` - this file compiled and if possible in pdf format.

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): 3 stages
- Pipelined multiplier unit (latency): 8 stages
- divider unit (latency): not pipelined unit, 20 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) Enhance the assembly program you created in the previous lab called **`program_1.s`**:

```
int m=1 /* 64  bit */
double k,p
for (i = 0; i < 64; i++){
        if (i is even) {
           p= v1[i] * ((double)( m<< i)) /*logic shift */
           m = (int)p
        } else {
        /*  i is odd */
            p= v1[i] / ((double)m* i))
            k =  ((float)((int)v4[i]/ 2^i)
        }

        v5[i] = ((p * v2[i]) + v3[i])+v4[i];

        v6[i] = v5[i]/(k+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 2 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	5196	4432	4496	3147
By simulation	5612	5487	5551	2203

Collect the IPC (from the simulator) for different programs.

	program_1.s	program_1_a.s	program_1_b.s	program_1_c.s
IPC	0.3310	0.3386	0.3462	0.3658

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

Eventual explanation:

In the case in which the computation by hand is higher, it could be linked to the fact that in the simulator the architecture enters the E fase and then stops to wait for an operand, while by we don't do that. By not doing that, the following instructions will have stalls given by structural hazards.

In the other case i must have not seen some stalls, but i really can't seem to find the points in which I made this mistake.