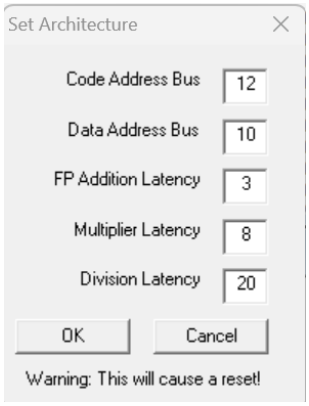
Architetture dei Sistemi di Elaborazione 02GOLOV	Delivery date: **9th November 2023**
Laboratory 3	Expected delivery of lab_03.zip must include: - `program_1_a.s`, `program_1_b.s` and `program_1_c.s` - this file compiled and if possible in pdf format.

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	6346	5578	5546	5611
By simulation	6388	6132	5900	5857

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	0.38	0.397	0.402	0.411

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

Eventual explanation:
Program_1_a.s: I cicli di clock si sono ridotti sia nel calcolo a mano sia nel calcolo dal simulatore, L'ottimizzazione è avvenuta riordinando le istruzioni per ridurre stalli e sfruttare al meglio la pipeline della cpu.

Program_1_b.s: Ulteriore riduzione dei colpi di clock, che si notano sia a meno che dal simulatore. L'utilizzo del delay slot consente in modo efficace l'esecuzione delle istruzioni subito dopo i salti, risparmiando ulteriori colpi di clock.

Program_1_c.s; Lo srotolamento x2 del ciclo ha ridotto in modo non troppo efficiente i cicli clock dal punto di vista del simulatore, sicuramente rispetto al tempo che ha impiegato la sua scrittura non lo rifarei, siccome il guadagno non è troppo elevato rispetto al tempo impiegato.

Tutti questi miglioramenti apportati al programma iniziale si riflettono nei risultati ottenuti, sia per quanto riguarda i colpi di clock che per l'IPC, quindi confermano che le ottimizzazioni effettuate hanno migliorato le prestazioni del programma sul Winmips64