# Computer Architectures
# Exam of 18.9.2023 - part I

*First name, Last name, ID*.....

## Question #2

Let's consider a superscalar MIPS64 architecture implementing dynamic scheduling, speculation, and multiple issues and composed of the following units:

- An issue unit able to process 2 instructions per clock period; in the case of a branch instruction, only one instruction is issued per clock period
- A commit unit able to process 1 instruction per clock period
- The following functional units (for each unit the number of clock periods to complete one instruction is reported):
  - 1 unit for memory access:1 clock period
  - 1 unit for integer arithmetic instructions: 1 clock period
  - 1 unit for branch instructions: 1 clock period
  - 1 unit for FP multiplication (pipelined): 5 clock periods
  - 1 unit for FP division (unpipelined): 6 clock periods
  - 1 unit for other FP instructions (pipelined): 3 clock periods
  - 2 Common Data Bus.
- Let's also assume that:
  - Branch predictions are always correct
  - All memory accesses never trigger a cache miss.

You should use the following table to describe the behavior of the processor during the execution of the first 2 iterations of the cycle, computing the total number of required clock cycles.
*Tip: For the EXE stage, it is recommended to add a lowercase letter to help distinguish between the different functional units available.*

| # iteration | Instruction | ISSUE x2 | EXE | MEM | CDB x2 | COMMIT | Notes |
|---|---|---|---|---|---|---|---|
| 1 | l.d f1,v1(r1) | 1 | 2m | 3 | 4 | 5 | |
| 1 | l.d f2,v2(r1) | 1 | 3m | 4 | 5 | 6 | |
| 1 | l.d f3,v3(r1) | 2 | 4m | 5 | 6 | 7 | |
| 1 | div f4, f1, f2 | 2 | 6d | | 12 | 13 | Wait for f1(4), f2(5) |
| 1 | add.d f5, f5, f3 | 3 | 7a | | 10 | 14 | Wait for f3(6), f5(0) |
| 1 | mul f6, f4, f5 | 3 | 13x | | 18 | 19 | Wait for f4(12), f5(10) |
| 1 | s.d f6,v4(r1) | 4 | 5s | | | 20 | Can calculate address immediately, but must wait f6(18) |
| 1 | daddui r1,r1,8 | 4 | 5i | | 6 | 21 | |
| 1 | daddi r2,r2,-1 | 5 | 6i | | 7 | 22 | |
| 1 | bnez r2,loop | 6 | 8b | | | 23 | Wait for r2(7) |
| 2 | l.d f1,v1(r1) | 7 | 8m | 9 | 10 | 24 | |
| 2 | l.d f2,v2(r1) | 7 | 9m | 10 | 11 | 25 | |
| 2 | l.d f3,v3(r1) | 8 | 10m | 11 | 12 | 26 | |
| 2 | div f4, f1, f2 | 8 | 12d | | 18 | 27 | Wait for f1(10), f2(11), div functional unit (11) |
| 2 | add.d f5, f5, f3 | 9 | 13a | | 16 | 28 | Wait for f3(12), f5(10) |
| 2 | mul f6, f4, f5 | 9 | 19x | | 24 | 29 | Wait for f4(18), f5(16) |
| 2 | s.d f6,v4(r1) | 10 | 11s | | | 30 | Can calculate address immediately, but must wait f6(24) |
| 2 | daddui r1,r1,8 | 10 | 11i | | 13 | 31 | FU stays busy one more cycle to wait for the CDB to be free |
| 2 | daddi r2,r2,-1 | 11 | 13i | | 14 | 32 | |
| 2 | bnez r2,loop | 12 | 15b | | | 33 | Wait for r2(14) |

Small variations with respect to this solution could be accepted as correct in a case by case manner.