Computer Architectures
Exam of 23.6.2021 - part I

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

Question #1

The Tomasulo architecture for superscalar processors with dynamic scheduling and speculation uses one or more Common Data Busses (CDBs).
You are requested to

1. Explain what the CDB is and where it is placed in the Tomasulo architecture, listing the modules able to write on it, and those reading from it
2. Summarize when data are written in the CDB
3. Explain the advantages possibly stemming from the introduction of multiple instances of the CDB.

Guidelines for answers:

1. The CDB is a bus where the different functional units write their result. The data carried by the CDB are read by
    a. All the reservation stations possibly waiting for operands for their operation
    b. The Register File
    c. The Reorder Buffer (ROB)
2. Data are written on the CDB by the functional units as soon as they are available, i.e., each time an instruction is completely processed
3. Having a single CDB means that it may become a bottleneck: if 2 or more functional units finish processing their instruction at the same time, only one of them can write the result on the CDB, while the other(s) must wait. Hence, having multiple CDBs allow improving the performance, clearly at the cost of an increased HW cost.

Computer Architectures
Exam of 23.6.2021 - part I

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

Question #2

Let consider a MIPS64 architecture including the following functional units (for each unit the number of clock periods to complete one instruction is reported):

- Integer ALU: 1 clock period
- Data memory: 1 clock period
- FP arithmetic unit: 2 clock periods (pipelined)
- FP multiplier unit: 4 clock periods (pipelined)
- FP divider unit: 8 clock periods (unpipelined)

You should also assume that

- The branch delay slot corresponds to 1 clock cycle, and the branch delay slot is not enabled
- Data forwarding is enabled
- The EXE phase can be completed out-of-order.

You should consider the following code fragment and, filling the following tables, determine the pipeline behavior in each clock period, as well as the total number of clock periods required to execute the fragment. The value of the constant k is written in f10 before the beginning of the code fragment.

```
; ********************* MIPS64 ***********************
;  for (i = 0; i < 100; i++) {
;
      v4[i] = ((v1[i]/k1) + (v2[i]/k2))*v3[i];
;  }
```

```
                .data
v1:     .double "100 values"
v2:     .double "100 values"
v3:     .double "100 values"
v4:     .double "100 values"


                .text
main:  daddui r1,r0,0      I1
       daddui r2,r0,100    I2
loop:  l.d  f1,v1(r1)      I3
       div.d  f2, f1, f10  I4
       l.d  f3,v2(r1)      I5
       div.d  f4, f3, f11  I6
       add.d  f5,f2,f4     I7
       l.d  f6,v3(r1)      I8
       mul.d  f7,f5,f6     I9
       s.d  f7,v4(r1)      I10
       daddui  r1,r1,8     I11
       daddi  r2,r2,-1     I12
       bnez  r2,loop       I13
       halt                I14
```

	Comments	Clock cycles
I1	r1← pointer	5
I2	r2 <= 100	1
I3	f1 <= v1[i]	1
I4	f2 <= v1[i]/k1	9
I5	f3 <= v2[i]	0
I6	f4 <= v2[i]/k2	8
I7	f6 <= v1[i]/k1*v2[i]/k2	2
I8	f5 <= v3[i]	1
I9	f7 <= v3[i]* (v1[i]/k1*v2[i]/k2)	5
I10	v4[i] <= f7	1
I11	r1 <= r1 + 8	1
I12	r2 <= r2 - 1	1
I13		2
I14		1
total		3206

Computer Architectures
Exam of 18.6.2020 - part I

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

main:	daddui r1,r0,0	F	D	E	M	W																																			5
	daddui r2,r0,100		F	D	E	M	W																																		1
loop:	l.d f1,v1(r1)			F	D	E	M	W																																	1
	div.d f2, f1, f10				F	D		E	E	E	E	E	E	E	E	M	W																								9
	l.d f3,v2(r1)					F		D	E	M	W																														0
	div.d f4, f3, f11							F	D							E	E	E	E	E	E	E	E	M	W																8
	add.d f5,f2,f4								F							D								E	E	M	W														2
	l.d f6,v3(r1)															F								D	E		M	W													1
	mul.d f7,f5,f6																							F	D			E	E	E	E	M	W							5
	s.d f7,v4(r1)																								F			D				E	M	W						1
	daddui r1,r1,8																											F				D	E	M	W					1
	daddi r2,r2,-1																															F	D	E	M	W				1
	bnez r2,loop																																F		D	E	M	W		2
	halt																																		F	D	E	M	W	1