Computer Architectures
Exam of 3.2.2021 – part I

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

Question #1

Let focus on the Reorder Buffer (ROB) existing in the architecture of some superscalar processors. You are requested to

1. Explain the role of the ROB (when an entry is allocated in it, when it is written, when it is read, when it is de-allocated)
2. Describe the ROB architecture, detailing the fields composing each entry
3. Summarizing the advantages stemming from the adoption of the ROB.

Answers

1. The ROB is crucial to allow in-order instruction commitment while execution is performed out-of-order. An entry in the ROB is allocated to an instruction when this is successfully issued. Its content is written via the CDB when the execution finishes its execution. It is read when other instructions use the result produced by the instruction while this is not yet committed. The ROB is a FIFO (i.e., a circular buffer). When an instruction becomes the oldest in the ROB, it is committed (i.e., its result is written in the destination) and the entry is de-allocated. If the instruction to be committed is a mispredicted branch, the whole buffer is flushed.
2. The ROB is a buffer, whose elements are composed of the following fields:
    - *Instruction type*
    - *Destination*
    - *Value*
    - *Ready*
3. The ROB allows the processor to implement dynamic scheduling with speculation. It also allows precise exception management.

Computer Architectures
Exam of 3.2.2021 - 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: 6 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 values of the constants k1 and k2 are written in f10 and f11 before the beginning of the code fragment.

```
; ********************* MIPS64 ***********************
;  for (i = 0; i < 10; i++) {
;       v4[i] = (v1[i]+v2[i])/(v3[i]*k);
;  }
```

```
            .data
v1:     .double "10 values"
v2:     .double "10 values"
V3:     .double "10 values"


            .text
main:  daddui r1,r0,0     I1
       daddui r2,r0,10    I2
loop:  l.d  f1,v1(r1)     I3
       l.d  f2,v2(r1)     I4
       add.d  f5, f1, f2  I5
       l.d  f3,v3(r1)     I6
       mul.d  f6, f3, f10 I7
       div.d f4, f5, f6   I8
       s.d  f4,v4(r1)     I9
       daddui  r1,r1,8    I10
       daddi  r2,r2,-1    I11
       bnez  r2,loop      I12
       halt               I13
```

	Comments	Clock cycles
	r1← pointer	5
	r2 <= 10	1
	f1 <= v1[i]	1
	f2 <= v2[i]	1
	f5 <= v1[i] +v2[i]	3
	f3 <= v2[i]	1
	f6 <= f3*k	7
	f4 <= (v1[i]+v2[i]) /(v3[i]*k)	8
	v4[i] <= f4	1
	r1 <= r1 + 8	1
	r2 <= r2 - 1	1
		2
		1
total		6+27*10=276

Computer Architectures
Exam of 11.9.2020 - part I

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

main:	daddui r1,r0,0	F	D	E	M	W																																5
	daddui r2,r0,10		F	D	E	M	W																															1
loop:	l.d f1,v1(r1)			F	D	E	M	W																														1
	l.d f2,v2(r1)				F	D	E	M	W																													1
	add.d f5, f1, f2					F	D		E	E	M	W																									3
	l.d f3,v3(r1)						F		D	E		M	W																								1
	mul.d f6, f3, f10								F	D			E	E	E	E	E	E	M	W																7
	div.d f4, f5, f6									F			D						E	E	E	E	E	E	E	E	M	W								8
	s.d f4,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																																				1