COMPUTER ARCHITECTURES

02LSEOQ, 02LSEOV – A.Y. 2021/22

LAB 05 – WINMIPS

Considering a MIPS architecture with the following characteristics:

- Integer ALU: 1 clock cycle
- Data memory: 1 clock cycle
- FP arithmetic unit: pipelined, 2 clock cycles
- FP multiplier unit: pipelined, 6 clock cycles
- FP divider unit: not pipelined, 10 clock cycles

Assume also:

- branch delay slot corresponding to 1 clock cycle, branch delay slot not enabled
- data forwarding enabled
- EXE stage could be completed also in out-of-order fashion.

Given the codes provided, esteem the number of clock cycles needed for completion. Also, fill up the tables with the pipeline stages at each clock cycle (this is needed only for one iteration of the loop).

```
------------------------------------------------------------------------------------------------------------------
; ********************* C ***********************
for (i = 0; i < 30; i++){
     v5[i] = (v1[i]*v2[i]) + v3[i];
     v6[i] =(v3[i]*v4[i])/v5[i]:
}
; ********************* MIPS64 ***********************
```

.data	comments	Clock cycles
V1: .double “30 values”		
V2: .double “30 values”		
V3: .double “30 values”		
…		
V5: .space 240		
V6: .space 240		
.text		
main: daddui r2, r0, 0	r2 ← pointer	
daddui r1, r0, 30	r1 <= 30	
cycle: l.d f3, v1(r2)	f3 <= v1[i]	
l.d f4, v2(r2)	f4 <= v2[i]	
l.d f5, v3(r2)	f5 <= v3[i]	
l.d f6, v4(r2)	f6 <= v4[i]	
mul.d f3, f3, f4	f3 = f3 * f4	
mul.d f7, f5, f6	f7 = f5 * f6	
daddi r1, r1, -1	…	
add.d f4, f5, f3	…	
div.d f7, f7, f4	…	
s.d f4, v5(r2)	…	
s.d f7, v6(r2)	…	
daddui r2, r2, 8	…	

bnez r1, cycle
halt

	…	
Total:		

daddui r2, r0, 0																																		
daddui r1, r0, 30																																		
l.d f3, v1(r2)																																		
l.d f4, v2(r2)																																		
l.d f5, v3(r2)																																		
l.d f6, v4(r2)																																		
mul.d f3, f3, f4																																		
mul.d f7, f5, f6																																		
daddi r1, r1, -1																																		
add.d f4, f5, f3																																		
div.d f7, f7, f4																																		
s.d f4, v5(r2)																																		
s.d f7, v6(r2)																																		
daddui r2, r2, 8																																		
bnez r1, cycle																																		
halt																																		