HIGH PERFORMANCE COMPUTER ARCHITECTURE 23-06-2005 **MATRICULATION NO.**_____

(FORMER "CALCOLATORI ELETTRONICI 2") **SURNAME**_____

REVISED 22-10-2023 **FIRST NAME**_____

1) (POINTS 14/40) Consider the following snippet of code running a processor that uses the Tomasulo's algorithm to perform the dynamic scheduling of instructions. The program performs the operation Y=aX/Y on a vector of 100 elements. Initially, R1 = 0 and F0 contains the value of the constant 'a'.

```
etic:  L.D    F2, 0(R1)      ; read Xi
       MUL.D  F4, F2, F0     ; multiply a*Xi
       L.D    F6, 400(R1)    ; load Yi
       DIV.D  F6, F4, F6     ; a*Xi/Yi
       S.D    F6, 400(R1)    ; store Yi
       ADDI   R1, R1, 8      ; update R1
       SGTI   R3, R1, 800    ; R1 >? 800, result in R3
       BEQ    R3, R0, etic   ; continue to loop if false
```

Working hypothesis:

- the pipeline implements a single-dispatch policy
- the instructions after a branch are executed speculatively and predicted 'taken'
- the issue stage (I) calculates the address of the actual reads and writes;
- reads require 1 clock cycle; writes require 0 clock cycles (write buffer + split-cache);
- there's only one CDB
- dispatch stage (D) and complete stage (C) require 1 clock cycle
- there are separated integer units for the calculation of the actual address, for arithmetic and logical operations, for the evaluation of the branch condition
- the functional units do not take advantage of pipelining techniques internally (reservation stations are busy until the end of CDB-write, except for Stores)
- the load buffer has 5 slots
- the store buffer has 5 slots (writes wait for the operand in the store buffer, i.e. in the execution stage)
- the rest of the processor and has the following characteristics

Type of Functional Unit	No. of Functional Units	Cycles for stage I	No. of reservation stations
Integer (effective addr.)	1	1	2
Integer (op. A-L)	1	1	2
Integer (branch calc.)	1	1	2
FP Adder	1	4	3
FP Multiplier	1	8	3
FP Divider	1	15	2

Complete the following chart until the end of the third iteration of the code fragment above in the case of simple dynamic scheduling.

Iter.	Instruction	P disPatch (start-stop)	I Issue (start-stop)	M MEM ACCESS (clock)	W CDB-Write (Complete) (clock)	C Commit (clock)	Comments
1	`L.D F2,0(R1)`	1-4	2	3	4	5	
…	…						
…	…						

2) (POINTS 10/40) For the same fragment of code of exercise 1, let's assume a single-pipeline processor such that the branch condition is solved in the decode stage, so that we have only 1 cycle for the delay slot. Moreover, let's assume that:

- The dispatch and complete stage requires 1 cycle
- There are the following latencies between operations:

Producer Instruction	Consumer Instruction	Latency (clock cycles)
FP operation	FP operation	4
FP operation	Store double	2
Load double	FP operation	2
Load double	Store double	1

The pipeline is single-dispatch: calculate the execution time (in cycles) of a single loop and show where there are stalls with and without static scheduling of the instructions (without unrolling techniques).

3) (POINTS 8/40) Explain the operation and draw a diagram of a PAg branch 2-level predictor with a 12-bit BSHR and size 2^{12} x 2 bit for the PHT.

4) Given the sequence P1: R, P2: R, P3: R, P1: W, P2: W, P3: W (Px:R = read by the processor Px, Px:W write by the processor Px), with respect to a certain variable 'a ', show for each processor the sequence of states, and transactions on the bus that occur in a multiprocessor UMA with write-back caches for each processor and DRAGON coherence protocol.

EXERCIZE 1

Iter.	Instruction		P disPatch (start-stop)	I Issue (start-stop)	M MEM-ACCESS (clock)	W CDB-Write (Complete) (clock)	C Commit (clock)	Comments
1	L.D	F2,0(R1)	1-4	2	3	4	5	
1	MUL.D	F4,F2,F0	2-13	5-12		13	14	I waits F2 from 1/L.D
1	L.D	F6,400(R1)	3-6	4	5	6	15	
1	DIV.D	F6,F4,F6	4-29	14-28		29	30	I waits F4 from 1/MUL.D
1	S.D	F6,400(R1)	5-5	6	30		31	I waits F6 from 1/DIV.D
1	ADDI	R1,R1,8	6-8	7		8	32	
1	SGTI	R3,R1,800	7-10	9		10	33	I waits R1 from 1/ADDI
1	BEQ	R3,R0,etic	8-11	11			34	I waits R3 from 1/SGTI
2	L.D	F2,0(R1)	9-12	10	11	12	35	
2	MUL.D	F4,F2,F0	10-21	13-20		21	36	I waits F2 from 2/L.D
2	L.D	F6,400(R1)	11-14	12	13	14	37	
2	DIV.D	F6,F4,F6	12-44	29-43		44	45	I waits F4 from 1/MUL.D e free DIV-FU
2	S.D	F6,400(R1)	13-13	14	45		46	I waits F6 from 2/DIV.D
2	ADDI	R1,R1,8	14-16	15		16	47	
2	SGTI	R3,R1,800	15-18	17		18	48	I waits R1 from 2/ADDI
2	BEQ	R3,R0,etic	16-19	19			49	I waits R3 from 2/SGTI
3	L.D	F2,0(R1)	17-20	18	19	20	50	
3	MUL.D	F4,F2,F0	18-30	21-28		30	51	I waits F2 from 2/L.D e CDB waits bus free
3	L.D	F6,400(R1)	19-19	20	21	22	52	
3	DIV.D	F6,F4,F6	30-59	44-58		59	60	D waits available DIV-RS, I waits free DIV-FU
3	S.D	F6,400(R1)	31-31	32	60		61	I waits F6 from 3/DIV.D
3	ADDI	R1,R1,8	32-34	33		34	62	
3	SGTI	R3,R1,800	33-36	35		36	63	I waits R1 from 3/ADDI
3	BEQ	R3,R0,etic	34-37	37			64	I waits R3 from 3/SGTI