

Nome, MATRICOLA

Domanda 1

Considerando il processore MIPS64 e l'architettura descritta in seguito:

- Integer ALU: 1 clock cycle
- Data memory: 1 clock cycle
- FP multiplier unit: not pipelined unit that requires 6 stages
- FP divider unit: not pipelined unit that requires 6 clock cycles
- FP arithmetic unit: pipelined 4 stages
- branch delay slot: 1 clock cycle, and the branch delay slot disabled
- forwarding enabled
- it is possible to complete instruction EXE stage in an out-of-order fashion.

Usando il frammento di codice riportato, si calcoli il tempo di esecuzione dell'intero programma in colpi di clock e si completi la seguente tabella.

```

;    for (i = 0; i < 100; i++) {
;        v4[i] = v1[i]*v2[i]* v3[i];
;}

```

.data																																		Clock cycles			
V1:	.double "100 values"																																				
V2:	.double "100 values"																																				
V3:	.double "100 values"																																				
V4:	.double "100 values"																																				
V5:	.double "100 values"																																				
.text																																					
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		
	l.d f2,v2(r1)				F	D	E	M	W																											1	
mul.d f4,f1,f2						F	D	s	*	*	*	*	*	*	M	W																	7				
	l.d f3,v3(r1)						F	s	D	E	M	W																				0					
mul.d f4,f4,f3								F	D	s	s	s	s	*	*	*	*	*	*	M	W													6			
	s.d f4,v4(r1)								F	s	s	s	s	D	E	s	s	s	s	s	M	W												1			
daddi r2,r2,-1														F	D	s	s	s	s	s	E	M	W											1			
	daddui r1,r1,8														F	s	s	s	s	s	D	E	M	W											1		
bnez r2,loop																					F	D	E	M	W											1	
Halt																						F	-	-	-	-											1
Total						6 + (20*100)																										2006					

Nome, MATRICOLA

Domanda 2

Considerando il programma precedente, e in particolare la istruzione:

mul.d f4,f4,f3

quali potrebbero essere i cammini di forwarding che potrebbero partecipare alla sua esecuzione? Perché dovrebbero essere attivati? motivare la risposta.



Il percorso di forwarding utilizzato riguarda quello tra la fase di EX/MEM, ovvero quello riguardante i risultati della fase di esecuzione. Questo perché prima abbiamo l'istruzione mul.d f4,f1,f2 che calcola il valore di f4 (caso di WAW e RAW). Se non fosse attivo questo percorso di forwarding bisognerebbe attendere che il valore di f4 sia prima scritto, quindi attendere la fase di write di mul.d f4,f1,f2. Ciò costerebbe 2 c.c. in più (per via del RAW hazard).

Nome, MATRICOLA

Domanda 3

Considerando il programma precedente e l'architettura del processore superscalare descritto in seguito; completare la tabella relativa alle prime 3 iterazioni.

Processor architecture:

- Issue 2 instructions per clock cycle
- jump instructions require 1 issue
- handle 2 instructions commit per clock cycle
- timing facts for the following separate functional units:
 - 1 Memory address 1 clock cycle
 - 1 Integer ALU 1 clock cycle
 - 1 Jump unit 1 clock cycle
 - 1 FP multiplier unit, which is not pipelined: 6 stages
 - 1 FP divider unit, which is not pipelined: 6 clock cycles
 - 1 FP Arithmetic unit, which is pipelined: 4 stages
- Branch prediction is always correct
- There are no cache misses
- There are 2 CDB (Common Data Bus).

# iteration		Issue	EXE	MEM	CDB x2	COMMIT x2
1	l.d f1,v1(r1)	1	2m	3	4	5
1	l.d f2,v2(r1)	1	3m	4	5	6
1	mul.d f4,f1,f2	2	6x		12	13
1	l.d f3,v3(r1)	2	4m	5	6	13
1	mul.d f4,f4,f3	3	13x		19	20
1	s.d f4,v4(r1)	3	5m			20
1	daddi r2,r2,-1	4	5i		6	21
1	daddui r1,r1,8	4	6i		7	21
1	bnez r2,loop	5	6j			22
2	l.d f1,v1(r1)	6	8m	9	10	22
2	l.d f2,v2(r1)	6	9m	10	11	23
2	mul.d f4,f1,f2	7	20x		26	27
2	l.d f3,v3(r1)	7	10m	11	12	27
2	mul.d f4,f4,f3	8	27x		33	34
2	s.d f4,v4(r1)	8	11m			34
2	daddi r2,r2,-1	9	10i		11	35
2	daddui r1,r1,8	9	11i		12 13	35
2	bnez r2,loop	10	11j			36
3	l.d f1,v1(r1)	11	14m	15	16	36
3	l.d f2,v2(r1)	11	15m	16	17	37
3	mul.d f4,f1,f2	12	34x		40	41
3	l.d f3,v3(r1)	12	16m	17	18	41
3	mul.d f4,f4,f3	13	41x		47	48
3	s.d f4,v4(r1)	13	17m			48
3	daddi r2,r2,-1	14	15i		16	49
3	daddui r1,r1,8	14	16i		17	49
3	bnez r2,loop	15	16j			50

Nome, MATRICOLA

Domanda 4

Considerando il segmento di codice presentato nella tabella precedente, se assumessimo che il ROB abbia una dimensione di 16 elementi, quale sarebbe la prima istruzione che dovrebbe stallare durante la esecuzione del programma? motivare la risposta.

Stalla alla terza iterazione di l.d f1,v1(r1). Questo perché vengono estratte 18 istruzioni, mentre 2 sono state liberate dalla ROB per via del commit. Arrivati a questa istruzione non ci sarà più spazio nella rob e quindi non si potrà eseguire l'issue.