Homework2

李青林(5110309074)

1 Q1

Q1. Scoreboarding vs. Tomasulo' s Algorithm

ADD.S F1, F2, F3
MUL.S F1, F2, F3

Scoreboarding would stop because of the WAW hazard.
But Tomasulo would continue would continue because the multiplier is free and it can avoid WAW hazard by register renaming.

2 Q2

Q2. Branch Prediction

a) m=2, n=2

b) not taken

B1: not taken
B2: taken

The predictor for B3 should be **01**
Since the state of predictor **01** is `00`, it predicte not taken.

c) **00** Because B3 is not taken

d) Each entry contains 8 bits.

$$10000/8 = 1250$$

The number of entry is $1024 = 2^{10}$

3 Q3

Q3. Code Scheduling

a) L.S F0, 0(R1)
L.S F1, 0(R2)
Stall
Stall
ADD.S F0, F0, F1
L.S F2, 0(R3)
L.S F3, 0(R4)
Stall
Stall
MUL.S F2, F2, F3
Stall
Stall
Stall
Stall
ADD.S F0, F0, F2
Stall
Stall
S.S F0, 0(R5)

18 cycles in all

b) L.S F2, 0(R3)
L.S F3, 0(R4)
L.S F0, 0(R1)
L.S F1, 0(R2)
MUL.S F2, F2, F3
ADD.S F0, F0, F1
ADD.S F0, F0, F2
S.S F0, 0(R5)

L.S F2, 0(R3)
L.S F3, 0(R4)
L.S F0, 0(R1)
L.S F1, 0(R2)
MUL.S F2, F2, F3
Stall
ADD.S F0, F0, F1

Stall
Stall
ADD.S F0, F0, F2
Stall
Stall
S.S F0, 0(R5)

13 cycles in all

4 Q4

Q4. VLIW

a)

ALU1	ALU2	MU1	MU2	FADD	FMUL
add r1, r1, -4	add r2, r2, -4	ld f1, 0(r1)	ld f1, 0(r1)		
add r3, r3, -4	add r3, r4, -1	ld f3, 0(r1)			
					fmul f4, f2, f1
			st f4, 4(r1)	fadd f5, f4, f3	
	benz r4, loop	st f5, 0(r3)			

b)

ALU1	ALU2	MU1	MU2	FADD	FMUL
add r4, r4, -2		ld f1, 0(r1)	ld f2, 0(r2)		
add r1, r1, -8	add r2, r2, -8	ld f6, -4(r1)	ld f7, -4(r2)		
add r3, r3, -8		ld f3, 0(r3)	ld f8, -4(r3)		
					fmul f4, f2, f1
					fmul f9, f8, f7
			st f4, 8(r1)	fadd f5, f4, f3	
			st f9, 4(r1)	fadd f10, f9, f8	
			st f5, 8(r3)		
	benz r4, loop		st f10, 4(r3)		

It do give us better performance.

5 Q5

Q5. Simple Cache

a)

reference	1	4	8	5	20	17
hit/miss	miss	miss	miss	miss	miss	miss
reference	19	56	9	11	4	43
hit/miss	miss	miss	miss	miss	miss	miss
reference	5	6	9	17		
hit/miss	hit	miss	hit	hit		

block NO.	0	1	2	3	4
content		17		19	4
block NO.	5	6	7	8	9
content	5	6		56	9
block NO.	10	11	12	13	14
content		43			
block NO.	15				
content					

b)

reference	1	4	8	5	20	17
hit/miss	miss	miss	miss	hit	miss	miss
reference	19	56	9	11	4	43
hit/miss	hit	miss	miss	hit	miss	miss
reference	5	6	9	17		
hit/miss	hit	hit	miss	hit		

block1	1	2	3	4
content	16	17	18	19
block2	1	2	3	4
content	4	5	6	7
block3	1	2	3	4
content	8	9	10	11
block4	1	2	3	4
content				

c)

reference	1	4	8	5	20	17
hit/miss	miss	miss	miss	miss	miss	miss
reference	19	56	9	11	4	43
hit/miss	miss	miss	miss	miss	hit	miss
reference	5	6	9	17		
hit/miss	hit	miss	hit	hit		

NO.	way0	way1
0	56	8
1	17	9
2		
3	43	11
4	4	20
5	5	
6	6	
7		

d)

reference	1	4	8	5	20	17
hit/miss	miss	miss	miss	miss	miss	miss
reference	19	56	9	11	4	43
hit/miss	miss	miss	miss	miss	hit	miss
reference	5	6	9	17		
hit/miss	hit	miss	hit	hit		

NO.	0	1	2	3
content	17	9	6	5
NO.	4	5	6	7
content	43	4	11	56
NO.	8	9	10	11
content	19	20	8	1
NO.	12	13	14	15
content				

6 Q6

Q6. SimpleScalar Assignment

anagram:

	hit rate
1bit-GAp	0.9671
1bit-PAg	0.9684
1bit-bimod	0.9335
2bit-GAp	0.9766
2bit-PAg	0.9750
2bit-bimod	0.9614

cc1:

	hit rate
1bit-GAp	0.8557
1bit-PAg	0.8210
1bit-bimod	0.8641
2bit-GAp	0.8454
2bit-PAg	0.8191
2bit-bimod	0.8823

go:

	hit rate
1bit-GAp	0.7419
1bit-PAg	0.7075
1bit-bimod	0.7361
2bit-GAp	0.7305
2bit-PAg	0.7156
2bit-bimod	0.7023