

体系结构第四次作业

PB19051183 吴承泽

1.

a)

调度与未调度指令流水如下：

时钟周期	未调度指令	调度指令
1	DADDIU R4,R1,#800	DADDIU R4,R1,#800
2	L.D F2,0(R1)	L.D F2,0(R1)
3	Stall	L.D F6,0(R2)
4	MUL.D F4,F2,F0	MUL.D F4,F2,F0
5	L.D F6,0(R2)	DADDIU R1,R1,#8
6	Stall	DADDIU R2,R2,#8
7	Stall	DSLTU R3,R1,R4
8	Stall	Stall
9	Stall	Stall
10	ADD.D F6,F4,F6	ADD.D F6,F4,F6
11	Stall	Stall
12	Stall	Stall
13	Stall	BNEZ R3,foo
14	S.D F6,0(R2)	S.D F6,-8(R2)
15	DADDIU R1,R1,#8	
16	DADDIU R2,R2,#8	
17	Stall	
18	DSLTU R3,R1,R4	
19	BNEZ R3,foo	
20	Stall	

未调度循环需要19个周期；调度后循环需要13个周期。

该调度实现的加速比为 $19/13 = 1.4615$ ，若通过硬件匹配编译的性能改进，则时钟频率应是原来的1.4615倍。

b)

展开3次后的循环如下（再多进行展开循环也是可行的，取最小次数）：

时钟周期	调度指令
1	L.D F2,0(R1)
2	L.D F6,0(R2)
3	MUL.D F4,F2,F0
4	L.D F8,8(R1)
5	L.D F12,8(R2)
6	MUL.D F10,F8,F0
7	L.D F14,16(R1)
8	L.D F18,16(R2)
9	MUL.D F16,F14,F0
10	ADD.D F6,F4,F6
11	DADDIU R1,R1,#24
12	DADDIU R2,R2,#24
13	ADD.D F12,F10,F12
14	DSLTU R3,R1,R4
15	ADD.D F18,F16,F18
16	BNEZ R3,foo
17	S.D F6,-24(R2)
18	S.D F12,-16(R2)
19	S.D F18,-8(R2)

展开**3次**，每个循环执行时间为 $19/3 = 6.333$ 个Cycle。

2.

a)

循环次数	指令	Issue	EX/MEM Access	WR (CDB)	注释	停顿周期数 (Cycle)
1	L.D F2,0(R1)	1	2	3		/
1	MUL.D F4,F2,F0	2	4~18	19	等待F2写回	1
1	L.D F6,0(R2)	3	4	5		/
1	ADD.D F6,F4,F6	4	20~29	30	等待F6、F4写回	15
1	S.D F6,0(R2)	5	31	/	等待F6写回	27
1	DADDIU R1,R1,#8	6	7	8		/
1	DADDIU R2,R2,#8	7	8	9		/
1	DSLTU R3,R1,R4	8	9	10		/
1	BNEZ R3,foo	9	11	/	等待R3写回	1
2	L.D F2,0(R1)	10	12	13	等待跳转后执行	/
2	MUL.D F4,F2,F0	11	19~33	34	等待MUL释放资源	7
2	L.D F6,0(R2)	12	13	14		/
2	ADD.D F6,F4,F6	13	35~44	45	等待F4写回	21
2	S.D F6,0(R2)	14	46	/	等待F6写回	31
2	DADDIU R1,R1,#8	15	16	17		/
2	DADDIU R2,R2,#8	16	17	18		/
2	DSLTU R3,R1,R4	17	18	19		/
2	BNEZ R3,foo	18	20	/	等待R3写回	1
3	L.D F2,0(R1)	19	20	21		/
3	MUL.D F4,F2,F0	20	34~48	49	等待MUL释放资源	13
3	L.D F6,0(R2)	21	22	23		/
3	ADD.D F6,F4,F6	22	50~59	60	等待F4写回	27

循环次数	指令	Issue	EX/MEM Access	WR (CDB)	注释	停顿周期数 (Cycle)
3	S.D F6,0(R2)	23	61	/	等待F6中写回	37
3	DADDIU R1,R1,#8	24	25	26		/
3	DADDIU R2,R2,#8	25	26	27		/
3	DSLTU R3,R1,R4	26	27	28		/
3	BNEZ R3,foo	27	29	/	等待R3写回	1

每个循环迭代的时钟周期数：

Iteration 1: $31 - 1 + 1 = 31$

Iteration 2: $46 - 10 + 1 = 37$

Iteration 3: $61 - 19 + 1 = 43$

3.

(1)

即有占有所有指令15%的分支指令，其中有90% * 90%预测正确且执行，其中有90% * 10%的预测错误且执行，有10%的缓冲不命中直接执行。

CPI:

$$CPI = 1 + 0.15 * 0.9 * 0.9 * 0 + 0.15 * 0.9 * 0.1 * 4 + 0.15 * 0.1 * 3 = 1.099$$

(2)

采用固定2个时钟周期延迟分支策略的CPI如下：

$$CPI = 1 + 0.15 * 2 = 1.3$$

显然 $1.099 < 1.3$ ，因此第一种方法执行速度更快。

4.

因为无条件转移指令占5%，而无条件转移指令的额外开销为 x 个时钟周期，因此：

$$1 + 0.05 * x = 1.1$$

因此 $x = 2$

当增设分支目标缓冲时**CPI**如下：

$$CPI = 1 + (2 * 0.5 * 0.1 + 0 * 0.5 + 0.9) = 1.01$$

因此 $CPI = 1.01$

