CPU 设计文档

一、数据通路

（见附表）

二、控制器

1. 主控制器

func	10 0001	10 0011							001000	
op	000000	00 0000	00 1101	10 0011	10 1011	000100	00 1111	000011	000000	000010
	addu	subu	ori	lw	sw	beq	lui	jal	jr	j
RegDst[1:0]	01	01	00	00	X	X	00	10	X	X
ALUSrc	0	0	1	1	1	0	1	X	X	X
MemtoReg [1:0]	00	00	00	01	X	X	00	10	X	X
RegWrite	1	1	1	1	0	0	1	1	0	0
MemWrite	0	0	0	0	1	0	0	0	0	0
nPC_sel	0	0	0	0	0	1	0	0	0	0
ExtOp[1:0]	X	X	00	01	01	X	10	X	X	X
ALUctr[1:0]	00	01	10	00	00	01	00	X	X	X
j_instr	0	0	0	0	0	0	0	1	0	1
jr	0	0	0	0	0	0	0	0	1	0

2. 暂停机制

a. Tuse 和 Tnew

IF/ID 当前指令		
指令类型	源寄存器	Tuse
beq	rs/rt	0
cal_r	rs/rt	1
cal_i	rs	1
load	rs	1
store	rs	1
store	rt	2
jr	rs	0

ID/EX(Tnew)				EX/MEM(Tnew)				MEM/WB(Tnew)			
cal_r	cal_i	load	jal	cal_r	cal_i	load	jal	cal_r	cal_i	load	jal
1/rd	1/rt	2/rt	0/$31	0/rd	0/rt	1/rt	0/$31	0/rd	0/rt	0/rt	0/$31

b. 构造阻塞矩阵

IF/ID 当前指令			ID/EX(Tnew)			EX/MEM(Tnew)
指令类型	源寄存器	Tuse	cal_r 1/rd	cal_i 1/rt	load 2/rt	load 1/rt
beq	rs/rt	0	暂停	暂停	暂停	暂停
cal_r	rs/rt	1			暂停	
cal_i	rs	1			暂停	
load	rs	1			暂停	
store	rs	1			暂停	
store	rt	2				
jr	rs	0	暂停	暂停	暂停	暂停

3. 转发机制

	流水级	IF/ID		ID/EX		EX/MEM
	源寄存器	rs	rt	rs	rt	rt
	涉及指令	beq,jr	beq	cal_r,cal_i, lw,sw	cal_r	sw
	转发MUX	MFRSD	MFRTD	MFRSE	MFRTE	MFRTM
	控制信号	F_RS_D	F_RT_D	F_RS_E	F_RT_E	F_RT_M
	输入 0	RF.RD1	RF.RD2	RS@E	RT@E	RT@M
ID/EX	jal 0/$31	PC8@E	PC8@E			
EX/MEM	cal_r 0/rd	AO@M	AO@M	AO@M	AO@M	
	cal_i 0/rt	AO@M	AO@M	AO@M	AO@M	
	jal 0/$31	PC8@M	PC8@M	PC8@M	PC8@M	
MEM/WB	cal_r 0/rd	MUX_WD	MUX_WD	MUX_WD	MUX_WD	MUX_WD
	cal_i 0/rt	MUX_WD	MUX_WD	MUX_WD	MUX_WD	MUX_WD
	load 0/rt	MUX_WD	MUX_WD	MUX_WD	MUX_WD	MUX_WD
	jal 0/$31	MUX_WD	MUX_WD	MUX_WD	MUX_WD	MUX_WD

三、测试程序

1、测试代码

```
ori $at, $0,0x5678
lui $a0,0x1234
addu $at,$at,$a0
subu $s0,$at,$0
ori $a1,$0,1
addu $a2,$a0,$at
sw $a2,4($0)
```

```
ori $a3,$0,6
subu $t0,$a3,$a1
lw $t1,-1($t0)
addu $t2,$t1,$t1

loop3:
ori $s7,$ra,0x23
beq $a2,$t1,loop1
lui $t1,0x1256
ori $t2,$t1,0x1111
jr $ra
nop

loop1:
lw $t3,4($0)
lui $t4,20
addu $t5,$t3,$t1
ori $s2,$t5,0x1357
jal loop2
ori $t3,$ra,0x5678

j loop4
ori $s1,$0,1

loop2:
addu $s0,$ra,$t3
ori $s1,$0,20
addu $t4,$t2,$t3
sw $t4,-8($s1)
lw $t5,12($0)
addu $t6,$0,$ra
jal loop3
addu $t1,$t1,$ra
ori $ra,$t6,0
jr $ra
ori $t7,$t6,1

loop4:
nop
addu $0,$t1,$t2
ori $t3,$0,1
beq $t3,$s1,loop4
ori $s1,$s1,0x111
jal loop5
```

```
sw $t1,-0x30a4($ra)

loop6:beq $0,$0,loop6
nop

loop5:
lw $t2,-0x30ac($ra)
lw $t3,4($t2)
addu $0,$t2,$t3
sw $ra,4($0)
lw $ra,4($0)
jr $ra
nop
```

2、测试期望

```
 45@00003000: $ 1 <= 00005678
 55@00003004: $ 4 <= 12340000
 65@00003008: $ 1 <= 12345678
 75@0000300c: $16 <= 12345678
 85@00003010: $ 5 <= 00000001
 95@00003014: $ 6 <= 24685678
 95@00003018: *00000004 <= 24685678
115@0000301c: $ 7 <= 00000006
125@00003020: $ 8 <= 00000005
135@00003024: $ 9 <= 24685678
155@00003028: $10 <= 48d0acf0
165@0000302c: $23 <= 00000023
185@00003034: $ 9 <= 12560000
195@00003044: $11 <= 24685678
205@00003048: $12 <= 00140000
215@0000304c: $13 <= 36be5678
225@00003050: $18 <= 36be577f
235@00003054: $31 <= 0000305c
245@00003058: $11 <= 0000767c
255@00003064: $16 <= 0000a6d8
265@00003068: $17 <= 00000014
275@0000306c: $12 <= 48d1236c
275@00003070: *0000000c <= 48d1236c
295@00003074: $13 <= 48d1236c
305@00003078: $14 <= 0000305c
315@0000307c: $31 <= 00003084
325@00003080: $ 9 <= 12563084
335@0000302c: $23 <= 000030a7
```

```
355@00003034: $ 9 <= 12560000
365@00003038: $10 <= 12561111
395@00003084: $31 <= 0000305c
425@0000308c: $15 <= 0000305d
445@00003060: $17 <= 00000001
475@00003098: $11 <= 00000001
505@000030a0: $17 <= 00000111
535@00003098: $11 <= 00000001
565@000030a0: $17 <= 00000111
575@000030a4: $31 <= 000030ac
575@000030a8: *00000008 <= 12560000
595@000030b4: $10 <= 00000000
615@000030b8: $11 <= 24685678
635@000030c0: *00000004 <= 000030ac
655@000030c4: $31 <= 000030ac
```

思考题

1. 在本实验中你遇到了哪些不同指令组合产生的冲突？你又是如何解决的？相应的测试样例是什么样的？请有条理的罗列出来。(非常重要)

a.cal_r

冲突类型	解决办法	测试样例
R-M-RS	M 级转发	addu $t1,$t2,$t3 subu $t4,$t1,$t2
R-M-RT	M 级转发	addu $t1,$t2,$t3 subu $t4,$t2,$t1

R-W-RS	W 级转发	addu $t1,$t2,$t3
		instr 无关
		subu $t4,$t1,$t2

R-W-RT	W 级转发	addu $t1,$t2,$t3
		instr 无关
		subu $t4,$t2,$t1

I-M-RS	M 级转发	ori $t1,$t2,1000 subu $t4,$t1,$t2
I-M-RT	M 级转发	lui $t1,100 subu $t4,$t2,$t1
I-W-RS	W 级转发	lui $t1,100 instr 无关 subu $t4,$t1,$t2
I-W-RT	W 级转发	ori $t1,$t2,100 instr 无关

		subu $t4,$t2,$t1
LD-M-RS	暂停	lw $t1,0($t2) addu $t3,$t1,$t2
LD-M-RT	暂停	lw $t1,0($t2) addu $t3,$t2,$t1

LD-W-RS	W 级转发	lw $t1,0($t2)
		instr 无关
		addu $t3,$t1,$t2

LD-W-RT	W 级转发	lw $t1,0($t2)
		instr 无关
		addu $t3,$t2,$t1

JAL-M-RS	M 级转发	jal loop addu $t2,$31,$t1
JAL-M-RT	M 级转发	jal loop addu $t2,$t1,$31
JAL-W-RS	W 级转发	jal loop 延迟槽 loop:addu $t2,$31,$t1
JAL-W-RT	W 级转发	jal loop 延迟槽 loop:addu $t2,$t1,$31

b.cal_i

冲突类型	解决办法	测试样例
R-M-RS	M 级转发	addu $t2,$t1,$t3 ori $t4,$t2,100
R-W-RS	W 级转发	addu $t2,$t1,$t3 instr 无关 ori $t4,$t2,100
I-M-RS	M 级转发	lui $t1,100 ori $t2,$t1,200
I-W-RS	W 级转发	lui $t1,100 instr 无关 ori $t2,$t1,200
LD-M-RS	暂停	lw $t1,0($0) ori $t2,$t1,100
LD-W-RS	W 级转发	lw $t1,0($0) instr 无关 ori $t2,$t1,100
JAL-M-RS	M 级转发	jal loop ori $t1,$31,100
JAL-W-RS	W 级转发	jal loop 延迟槽 loop:ori $t1,$31,100

c.beq

冲突类型	解决办法	测试样例
R-E-RS	暂停	addu $t1,$t2,$t3 beq $t1,$t4,loop
R-E-RT	暂停	addu $t1,$t2,$t3 beq $t4,$t1,loop
R-M-RS	M 级转发	addu $t1,$t2,$t3 instr 无关 beq $t1,$t4,loop
R-M-RT	M 级转发	addu $t1,$t2,$t3 instr 无关 beq $t4,$t1,loop
R-W-RS	W 级转发	addu $t1,$t2,$t3 instr 无关 instr 无关 beq $t1,$t4,loop
R-W-RT	W 级转发	addu $t1,$t2,$t3 instr 无关 instr 无关 beq $t4,$t1,loop
I-E-RS	暂停	ori $t1,$t2,100 beq $t1,$t4,loop
I-E-RT	暂停	ori $t1,$t2,100 beq $t4,$t1,loop
I-M-RS	M 级转发	ori $t1,$t2,100 instr 无关 beq $t1,$t4,loop
I-M-RT	M 级转发	ori $t1,$t2,100 instr 无关 beq $t4,$t1,loop
I-W-RS	W 级转发	ori $t1,$t2,100 instr 无关 instr 无关 beq $t1,$t4,loop
I-W-RT	W 级转发	ori $t1,$t2,100 instr 无关 instr 无关 beq $t4,$t1,loop
LD-E-RS	暂停	lw $t1,0($t0) beq $t1,$t4,loop
LD-E-RT	暂停	lw $t1,0($t0) beq $t4,$t1,loop
LD-M-RS	暂停	lw $t1,0($t0)

		instr 无关 beq $t1,$t4,loop
LD-M-RT	暂停	lw $t1,0($t0) instr 无关 beq $t4,$t1,loop
LD-W-RS	W 级转发	lw $t1,0($t0) instr 无关 instr 无关 beq $t1,$t4,loop
LD-W-RT	W 级转发	lw $t1,0($t0) instr 无关 instr 无关 beq $t4,$t1,loop
JAL-M-RS	M 级转发	jal loop1 延迟槽 loop1:beq $31,$t1,loop2
JAL-M-RT	M 级转发	jal loop1 延迟槽 loop1:beq $t1,$31,loop2
JAL-W-RS	W 级转发	jal loop1 延迟槽 loop1: instr 无关 beq $31,$t1,loop2
JAL-W-RT	W 级转发	jal loop1 延迟槽 loop1: instr 无关 beq $t1,$31,loop2

d.load

冲突类型	解决办法	测试样例
R-M-RS	M 级转发	addu $t1,$t2,$t3 lw $s1,0($t1)

R-W-RS	W 级转发	addu $t1,$t2,$t3
		instr 无关
		lw $s1,0($t1)

I-M-RS	M 级转发	ori $t1,$t2,100 lw $s1,0($t1)
I-W-RS	W 级转发	ori $t1,$t2,100 instr 无关 lw $s1,0($t1)
LD-M-RS	暂停	lw $t1,0($t2) lw $s1,0($t1)

LD-W-RS	W 级转发	lw $t1,0($t2)
		instr 无关
		lw $s1,0($t1)

JAL-M-RS	M 级转发	jal loop lw $t1,4($31)
JAL-W-RS	W 级转发	jal loop 延迟槽 loop:lw $t1,4($31)

e.store

冲突类型	解决办法	测试样例
R-M-RS	M 级转发	addu $t1,$t2,$t3 sw $s1,0($t1)

R-W-RS	W 级转发	addu $t1,$t2,$t3
		instr 无关
		sw $s1,0($t1)

R-W-RT	W 级转发	addu $t1,$t2,$t3 sw $t1,0($t2)
I-M-RS	M 级转发	ori $t1,$t2,100 sw $s1,0($t1)
I-W-RS	W 级转发	ori $t1,$t2,100 instr 无关 sw $s1,0($t1)
I-W-RT	W 级转发	ori $t1,$t2,100 sw $t1,0($t2)
LD-M-RS	暂停	lw $t1,0($t2) sw $t3,0($t1)

LD-W-RS	W 级转发	lw $t1,0($t2)
		instr 无关
		sw $t3,0($t1)

LD-W-RT	W 级转发	lw $t1,0($t2) sw $t1,0($t5)
JAL-M-RS	M 级转发	jal loop sw $t1,0($31)
JAL-W-RS	W 级转发	jal loop 延迟槽 loop:sw $t1,0($31)
JAL-W-RT	W 级转发	jal loop sw $31,0($t1)

f.jr

冲突类型	解决办法	测试样例
R-E-RS	暂停	addu $t1,$t2,$t3 jr $t1

R-M-RS	M 级转发	addu $t1,$t2,$t3 instr 无关 jr $t1
R-W-RS	W 级转发	addu $t1,$t2,$t3 instr 无关 instr 无关 jr $t1
I-E-RS	暂停	ori $t1,$t2,100 jr $t1
I-M-RS	M 级转发	ori $t1,$t2,100 instr 无关 jr $t1
I-W-RS	W 级转发	ori $t1,$t2,100 instr 无关 instr 无关 jr $t1
LD-E-RS	暂停	lw $t1,0($t2) jr $t1
LD-M-RS	暂停	lw $t1,0($t2) instr 无关 jr $t1
LD-W-RS	W 级转发	lw $t1,0($t2) instr 无关 instr 无关 jr $t1
JAL-M-RS	M 级转发	jal loop 延迟槽 loop:jr $ra
JAL-W-RS	W 级转发	jal loop 延迟槽 loop: instr 无关 jr $ra