计算机组成原理 P5 实验报告

彭杰奇 15061169

一、 数据通路设计

F 级功能部件：

1. IFU 模块

（1）基本描述

IFU 主要功能是完成取指令功能。IFU 内部包括了 PC、IM(指令存储器)以及其他相关逻辑。

（2）模块接口

文件	模块接口定义
IFU.v	IFU(nextPC, Clk, Reset, PC_En, PC4, Instr); input [31:0] nextPC; input Clk; input Reset; input PC_En; output [31:0] Instr; output [31:0] PC4;

信号名	方向	描述
nextPC[31:0]	I	输入 PC 的下一条指令地址
Reset	I	复位信号，1:有效，0:无效
Clk	I	时钟信号
PC_En	I	IFU 内部 PC 的使能端，1:有效，0:无效
Instr[31:0]	O	当前指令输出
PC4[31:0]	O	当前 PC 下 PC + 4 的值

（3）功能定义

序号	功能名称	功能描述
1	复位	复位信号有效时，PC 设置为 0x00003000
2	取指令	根据 PC 当前值从 IM 中取指令输出
3	输出 PC+4	PC4 为当前 PC 下 PC + 4 的值

2. IM 模块

（1）基本描述

IM 是指令存储模块，由一个 32bit×1024 字的存储器组成，其功能是保存指令，并根据输入的 PC 输出相应指令。

（2）模块接口

文件	模块接口定义

IM.v	IM(Addr,Instr);
	input [11:2] Addr; // 输入的指令地址
	output [31:0] Instr; // 输出的指令

信号名	方向	描述
Addr[11:2]	I	输入的指令地址
Instr[31:0]	O	输出的指令

（3）功能定义

序号	功能名称	功能描述
1	输出指令	Instr ← im[Addr]

IF_ID 流水线寄存器：

(1)模块接口

文件	模块接口定义
IF_ID.v	IF_ID(IM, ADD4, Clk, Reset, IR_D_En, IR_D, PC4_D); input [31:0] IM; input [31:0] ADD4; input Clk; input Reset; input IR_D_En; output [31:0] IR_D; output [31:0] PC4_D;

（2）功能定义

序号	功能名称	功能描述
1	复位	清空寄存器
2	存数取数	将前一级的值存入寄存器中，将后一级的值输出
3	冻结	冻结 IR_D

D 级功能部件：

1.GRF 模块

（1）基本描述

GRF 模块为通用寄存器堆，主要由 32 个具有写使能端的 32 位寄存器组成，能够同时根据由 rs 和 rt 输入的地址从其中两个寄存器中读出数据，并根据 wr 中输入的地址向其中一个寄存器写入数据。

（2）模块接口

文件	模块接口定义
GRF.v	GRF(rs, rt, wr, WData, Clk, Reset, RegWrite, RData1, RData2); input [4:0] rs; input [4:0] rt; input [4:0] wr; input [31:0] WData; input Clk; input Reset; input RegWrite; output [31:0] RData1; output [31:0] RData2;

信号名	方向	描述
rs[4:0]	I	rs 寄存器的地址
rt[4:0]	I	rt 寄存器的地址
wr[4:0]	I	要写入的寄存器的地址
WData[31:0]	I	要写入的数据
Clk	I	时钟信号
Reset	I	复位信号
RegWrite	I	一般写使能信号，1:有效，0:无效
RData1[31:0]	O	rs 寄存器的值
RData2[31:0]	O	rt 寄存器的值

（3）功能定义

序号	功能名称	功能描述
1	读数据	RData1←(GRF[rs]) RData2←(GRF[rt])
2	写数据	RegWrite 有效时，(GPR[wr])←WData
3	清零	复位信号有效时，GRF 中所有寄存器都清零

2. EXT 模块

（1）基本描述

EXT 模块的作用是将 16 位立即数扩展为 32 位。

（2）模块接口

文件	模块接口定义
EXT.v	EXT(Imm_16, ExtOp, Imm_32); input [15:0] Imm_16; input [1:0] ExtOp; output [31:0] Imm_32;

信号名	方向	描述
Imm_16[15:0]	I	要扩展的 16 位立即数
ExtOp[1:0]	I	扩展方式选择信号 2’b00:符号扩展 2’b01:后接 16 位 0 2’b10:无符号扩展
Imm_32[31:0]	O	扩展后的 32 位立即数

（3）功能定义

序号	功能名称	功能描述
1	位数扩展	ExtOp 为 2’b00 时，16 位立即数正常符号扩展为 32 位 ExtOp 为 2’b01 时，16 为立即数后接 16 位 0 扩展为 32 位 ExtOp 为 2’b10 时，16 为立即数无符号扩展为 32 位

3. CMP 模块

（1）基本描述

CMP 模块用来比较输入的两个数据是否相等，用于 beq 指令是否跳转的判断

（2）模块接口

文件	模块接口定义
CMP.v	CMP(A1, A2, equal); input [31:0] A1; input [31:0] A2; output equal;

信号名	方向	描述
A1[31:0]	I	第一个数据
A2[31:0]	I	第二个数据
Equal	O	输入的数据是否相等 1:相等 0:不相等

（3）功能定义

序号	功能名称	功能描述
1	等于判断	equal ← (A1==A2)?1:0

4. NPC 模块

（1）基本描述

NPC 模块能够输出当前指令为 beq 或 J 类型等跳转指令时下一条指令地址

（2）模块接口

文件	模块接口定义
NPC.v	NPC(PC4, Instr, J_Sel, Branch, Zero, nPC); input [31:0] PC4; input [31:0] Instr; input [1:0] J_Sel; input Branch; input Zero; output [31:0] nPC;

信号名	方向	描述
PC4[31:0]	I	来自 PC4_D 寄存器
Instr[31:0]	I	来自 IR_D 寄存器
J_Sel[1:0]	I	当下 J 类型指令具体为: 2’b00:不是 j 类型指令 2’b01:指令为 J 2’b10:指令为 Jal 2’b11:指令为 Jr
Branch	I	当下指令是否为 beq 1:是，0:不是
Zero	I	若为 beq 指令，比较的两个数据是否相等 1:相等，0:不相等
nPC[31:0]	O	下一条指令地址 J_Sel 为 2’b01 或 2’b10: nPC ← PC[31:28]||index||0^2 Branch 为 1 且 Zero 为 1: nPC ← PC + 4 + Imm_32||0^2 Branch 为 1 而 Zero 为 0: nPC← PC4 + 4 其中: PC = PC4-4　Index = Instr[25:0]　Imm16 = Instr[15:0]

（3）功能定义

序号	功能名称	功能描述
1	输出下一条 PC 地址	nPC 输出当指令为 beq 或者 J 类型指令时下一条指令的地址

ID_EX 流水线寄存器：

(1)模块接口

文件	模块接口定义
ID_EX.v	ID_EX(IR_D, PC4_D, RD1, RD2, EXT, Clk, Reset, IR_E_Clr, IR_E, PC4_E, RS_E, RT_E, EXT_E); input [31:0] IR_D; input [31:0] PC4_D; input [31:0] RD1; input [31:0] RD2; input [31:0] EXT; input Clk; input Reset; input IR_E_Clr; output [31:0] IR_E; output [31:0] PC4_E; output [31:0] RS_E; output [31:0] RT_E; output [31:0] EXT_E;

（2）功能定义

序号	功能名称	功能描述
1	复位	清空寄存器
2	存数取数	将前一级的值存入寄存器中，将后一级的值输出
3	清除	清除 IR_E 的值

EX 级功能部件：

1. ALU 模块

（1）基本描述

ALU 为算数逻辑单元，可以对输入的两个数据进行加、减、按位与和按位或操作，并能够判断输入数据是否相等。

（2）模块接口

文件	模块接口定义

ALU.v	ALU(A1, A2, ALUCtr, ALUResult); input [31:0] A1; input [31:0] A2; input [2:0] ALUCtr; output [31:0] ALUResult;

信号名	方向	描述
A1[31:0]	I	第一个运算数
A2[31:0]	I	第二个运算数
ALUCtr[2:0]	I	ALU 控制信号 2'b000:加法运算 2'b001:减法运算 2'b010:按位与运算 2'b011:按位或运算
ALUResult[31:0]	O	ALU 运算结果

（3）功能定义

序号	功能名称	功能描述
1	加法运算	ALUResult ← A1+A2
2	减法运算	ALUResult ← A1-A2
3	按位与运算	ALUResult ← A1&A2
4	按位或运算	ALUResult ← A1|A2

EX_MEM 流水线寄存器：

(1)模块接口

文件	模块接口定义
EX_MEM.v	EX_MEM(IR_E, PC4_E, AO, RT_E, Clk, Reset, IR_M, PC4_M, AO_M, RT_M); input [31:0] IR_E; input [31:0] PC4_E; input [31:0] AO; input [31:0] RT_E; input Clk; input Reset; output [31:0] IR_M; output [31:0] PC4_M; output [31:0] AO_M; output [31:0] RT_M;

（2）功能定义

序号	功能名称	功能描述
1	复位	清空寄存器

2	存数取数	将前一级的值存入寄存器中，将后一级的值输出

MEM 级功能部件：

1.DM 模块

（1）基本描述

DM 模块为数据存储器，由一个 32bit * 32 字的存储器构成，起始地址为 0x00000000 用于存储数据。

（2）模块接口

文件	模块接口定义
DM.v	DM(Addr,Din,MemWrite,MemRead,Clk,Reset,Dout); input [31:0] Addr; input [31:0] Din; input MemWrite; input MemRead; input Clk; input Reset; output [31:0] Dout;

信号名	方向	描述
Addr[31:0]	I	读/写 DM 的地址
Din[31:0]	I	要写入 DM 的数据
MemWrite	I	写 DM 的控制信号
MemRead	I	读 DM 的控制信号
Clk	I	时钟信号
Reset	I	复位信号
Dout[31:0]	O	从 DM 读出的数据

（3）功能定义

序号	功能名称	功能描述
1	读数据	当 MemRead 为 1 时，ReadData ← RAM(Addr[11:2])
2	写数据	当 MemWrite 为 1 时，RAM(Addr) ← WriteData
3	清零	复位信号有效时，存储器清零

MEM_WB 流水线寄存器：

(1)模块接口

文件	模块接口定义
MEM_WB.v	MEM_WB(IR_M, PC4_M, AO_M, DR_M, Clk, Reset, IR_W, PC4_W, AO_W, DR_W); input [31:0] IR_M; input [31:0] PC4_M; input [31:0] AO_M; input [31:0] DR_M; input Clk; input Reset; output [31:0] IR_W; output [31:0] PC4_W; output [31:0] AO_W; output [31:0] DR_W;

（2）功能定义

序号	功能名称	功能描述
1	复位	清空寄存器
2	存数取数	将前一级的值存入寄存器中，将后一级的值输出

二、 控制器设计

1. Controller 模块定义

（1）基本描述

Controller 模块为 CPU 控制器，可以根据输入指令的 opcode 和 funct 值输出各种控制信号。

（2）模块接口

文件	模块接口定义
Controller.v	Controller(Instr, RegDst, ALUSrc, MemtoReg, RegWrite, MemWrite, MemRead, ExtOp, Branch, J_Sel, ALUCtr, PC_Sel); input [31:0] Instr; output [1:0] RegDst; output ALUSrc; output [1:0] MemtoReg; output RegWrite; output MemWrite; output MemRead; output [1:0] ExtOp; output Branch; output [1:0] J_Sel; output [1:0] PC_Sel;

	output [2:0] ALUCtr;

信号名	方向	描述
Instr[31:0]	I	指令
RegDst[1:0]	O	寄存器写入端地址控制 2’b00:选择 rt 字段 2’b01:选择 rd 字段 2’b10:选择 31 号寄存器
ALUSrc	O	ALU 输入端 A2 选择 0:选择 MSRTE 1:选择 EXT_E
MemtoReg[1:0]	O	寄存器堆写入端 WD 选择 2’b00:来自 ALU 输出 2’b01:来自 DM 输出 2’b10:来自 PC4_W+4
RegWrite	O	写寄存器控制信号
MemWrite	O	写 DM 控制信号
MemRead	O	读 DM 控制信号
ExtOp[1:0]	O	EXT 扩展方式控制信号
Branch	O	判断是否为 beq 指令
J_Sel[1:0]	O	2’b00:其他指令 2’b01:J 指令 2’b10:Jal 指令 2’b11:Jr 指令
PC_Sel[1:0]	O	2’b00:IFU 的 nextPC 选择 PC+4 2’b01:IFU 的 nextPC 选择 nPC 2’b10:IFU 的 nextPC 选择 RData1
ALUCtr[2:0]	O	ALU 控制信号

2. Controller 真值表

Instr	Subu	addu	Jr
opcode	000000	000000	000000
funct	100011	100001	001000
RegDst[1:0]	2’b01	2’b01	2’b01
ALUSrc	0	0	0
MemtoReg[1:0]	2’b00	2’b00	2’b00
RegWrite	1	1	0
Branch	0	0	0
J_Sel[1:0]	2’b00	2’b00	2’b11
ExtOp[1:0]	X	X	X

MemRead	0	0	0
MemWrite	0	0	0
PC_Sel[1:0]	2’b00	2’b00	2’b10
ALUCtr[2:0]	3’b000	3’b001	3’b000

Instr	J	ori	sw	lw	lui	beq	jal
opcode	000010	001101	100011	101011	000100	001111	000011
funct	N/A						
RegDst[1:0]	X	2’b00	2’b00	X	X	2’b00	2’b10
ALUSrc	0	1	1	1	0	1	0
MemtoReg[1:0]	X	0	1	X	X	0	2’b10
RegWrite	0	1	1	0	0	1	1
Branch	0	0	0	0	1	0	0
J_Sel[1:0]	2’b01	2’b00	2’b00	2’b00	2’b00	2’b00	2’b10
ExtOp[1:0]	2’b00	2’b10	2’b00	2’b00	2’b00	2’b01	2’b00
MemRead	0	0	1	0	0	0	0
MemWrite	0	0	0	1	0	0	0
PC_Sel[1:0]	2’b01	2’b00	2’b00	2’b00	2’b00	2’b01	2’b01
ALUOp[2:0]	3’b000	3’b010	3’b000	3’b000	3’b000	3’b000	3’b000

三、 冲突控制器

(1)模块接口

文件	模块接口定义
Conflict_manager.v	Conflict_manager(IR_D, IR_E, IR_M, IR_W, FRSD, FRTD, FRSE, FRTE, FRTM, stall); input [31:0] IR_D; input [31:0] IR_E; input [31:0] IR_M; input [31:0] IR_W; output [1:0] FRSD; output [1:0] FRTD; output [1:0] FRSE; output [1:0] FRTE; output [1:0] FRTM; output stall;

（2）功能定义

序号	功能名称	功能描述
1	产生暂停信号	暂停信号 stall
2	产生转发信号	5 个转发信号

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

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

转发机制

流水线寄存器	源寄存器	涉及指令									
IR@D	rs	beq, jr	MFRSD	FRSD	RF. RD1	AO@M	M4	PC4_E+4	PC4_M+4	M4
	rt	beq	MFRTD	FRTD	RF. RD2	AO@M	M4	PC4_E+4	PC4_M+4	M4
IR@E	rs	cal_r, cal_i, ld, st	MFRSE	FRSE	RS@E	AO@M	M4	PC4_M+4	/	/
	rt	cal_r, st	MFRTE	FRTE	RT@E	AO@M	M4	PC4_M+4	/	/
IR@M	rt	st	MFRTM	FRTM	RT@M	M4	/	/	/	/
			转发Mux	控制信号	输入0	输入1	输入2	输入3	输入4	输入5

							ID/EX	EX/MEM			MEM/WB			
							jal	jal	cal_r	cal_i	jal	cal_r	cal_i	load
							0	0	0	0	0	0	0	0
流水线寄存器		源寄存器	涉及指令				31	31	rd	rt	31	rd	rt	rt
IR@D		rs	beq, jr		MFRSD	FRSD	RF. RD1	PC4_E+4	PC4_M+4	AO	AO	M4	M4	M4	M4
		rt	beq		MFRTD	FRTD	RF. RD2	PC4_E+4	PC4_M+4	AO	AO	M4	M4	M4	M4
IR@E		rs	cal_r, cal_i, ld, st		MFRSE	FRSE	RS@E	/	PC4_M+4	AO	AO	M4	M4	M4	M4
		rt	cal_r, st		MFRTE	FRTE	RT@E	/	PC4_M+4	AO	AO	M4	M4	M4	M4
IR@M		rt	st		MFRTM	FRTM	RT@M	/	/	/	/	M4	M4	M4	M4
					转发Mux	控制信号	输入0								

暂停机制

IF/ID当前指令			ID/EX			EX/MEM
指令类型	源寄存器	Tuse	cal_r1/rd	cal_i1/rt	load2/rt	load1/rt
beq	rs/rt	0	pause	pause	pause	pause
cal_r	rs/rt	1			pause	
cal_i	rs	1			pause	
load	rs	1			pause	
store	rs	1			pause	
store	rt	2				
jr	rs	0	pause	pause	pause	pause

四、 测试程序

```
lui $3,1
ori $4,$3,2

start:
addu $5,$3,$4     #R=addu
beq $5,$3,label      #R_E_RS
subu $6,$5,$3     #R=subu
ori $7,$6,3    #R_M_RS
ori $8,$6,1    #R_W_RS
sll $5,$8,2    #test sll
srl $6,$5,1    #test srl
xor $6,$5,$6      #test xor
lui $9,2

j label        #test j
subu $13,$7,$8    #延迟槽
addu $10,$8,$9

label:
addi $10,$9,2     #test addi
subu $10,$7,$8    #R=subu
addu $11,$8,$10      #R_M_RT
subu $12,$11,$10  #R_W_RT
beq $10,$13,label2   #R_W_RS
nop

label3:
sw $3,4($0)
sw $4,8($0)
ori $5,$4,0xf1
lui $6,0xff
sw $6,12($0)
lw $7,12($0)      #Load
beq $7,$5,label4  #L_E_RS
nop
lw $8,4($0)    #Load
addu $9,$8,$7     #L_M_RS
ori $10,$8,5      #L_W_RS
beq $3,$8,label4  #L_W_RT
lui $11,2
```

```
nop
nop

label5:
jal label6
addu $10,$31,$0      #Jal_M_RS
label6:
beq $10,$31,label7   #Jal_M_RT
nop
label7:
jal label8
addu $11,$31,$31  #Jal_M_RS_RT
subu $12,$11,$31
beq $12,$12,label9
nop
label8:
addu $12,$31,$31  #Jal_W_RS_RT
jr $31         #Jal_W_RS
subu $11,$11,$11

label9:        #othertest
   lui $17,0xfe
   beq $17,$17,label10
   ori $18,0x30c8
   lui $20,18
   lui $21,0xfe11
   ori $21,1
   addu $23,$31,$20
   nop
   jal end
   nop
label2:
and $12,$3,$4     #test and
or $13,$3,$4      #test or
ori $5,$3,1    #I=ori
beq $5,$3,label      #I_E_RS
addu $6,$5,$4     #I_W_RS
addu $7,$5,$3     #内部转发
ori $8,$6,3    #I=ori
subu $9,$8,$7     #I_M_RS
addu $10,$9,$8    #I_W_RT
beq $8,$7,label3  #I_W_RS
nop
```

```
ori $12,$5,2
nop
lui $13,3      #I=lui
add $14,$12,$13      #I_M_RT
ori $13,$13,0xfe
ori $11,$5,2      #I=ori
bne $13,$11,label3   #I_E_RT
subu $12,$11,$5

label4:
lw $12,8($0)      #Load
sub $13,$5,$12    #L_M_RT
addu $14,$12,$13  #L_W_RS
addu $16,$5,$0
sw $5,16($0)
lw $15,16($0)     #Load
beq $16,$15,label5   #L_E_RT
addu $16,$16,$15

label10:
jr $18
nop
end:
nop
```

预期结果

```
$ 3 <= 00010000
$ 4 <= 00010002
$ 5 <= 00020002
$ 6 <= 00010002
$ 7 <= 00010003
$ 8 <= 00010003
$ 5 <= 0004000c
$ 6 <= 00020006
$ 6 <= 0006000a
$ 9 <= 00020000
$13 <= 00000000
$10 <= 00020002
$10 <= 00000000
$11 <= 00010003
$12 <= 00010003
$12 <= 00010000
$13 <= 00010002
```

$ 5 <= 00010001
$ 6 <= 00020003
$ 7 <= 00020001
$ 8 <= 00020003
$ 9 <= 00000002
$10 <= 00020005
$12 <= 00010003
$13 <= 00030000
$14 <= 00040003
$13 <= 000300fe
$11 <= 00010003
$12 <= 00000002
*00000004 <= 00010000
*00000008 <= 00010002
$ 5 <= 000100f3
$ 6 <= 00ff0000
*0000000c <= 00ff0000
$ 7 <= 00ff0000
$ 8 <= 00010000
$ 9 <= 01000000
$10 <= 00010005
$11 <= 00020000
$12 <= 00010002
$13 <= 000000f1
$14 <= 000100f3
$16 <= 000100f3
*00000010 <= 000100f3
$15 <= 000100f3
$16 <= 000201e6
$31 <= 00003094
$10 <= 00003094
$31 <= 000030a4
$11 <= 00006148
$12 <= 00006148
$11 <= 00000000
$12 <= ffffcf5c
$17 <= 00fe0000
$18 <= 000030c8
$20 <= 00120000
$21 <= fe110000
$21 <= fe110001
$23 <= 001230a4
$31 <= 000030e4

五、 思考题

用例编号	测试类型	前序指令	冲突位置	冲突寄存器	测试序列	测试序列
1	R_E_RS	addu	E	rs	addu $5,$3,$4	addu $5,$3,$4
					beq $5,$3,label	jr $5
2	R_E_RT	addu	E	rt	addu $5,$3,$4	
					beq $3,$5,label	
3	R_M_RS	subu	MEM	rs	subu $5,$3,$4	subu $5,$3,$4
					addu $6,$5,$3	ori $6,$5,1
					beq $5,$3,label	jr $5
4	R_M_RT	subu	MEM	rt	subu $5,$3,$4	subu $5,$3,$4
					addu $6,$3,$5	addu $6,$3,$5
						beq $3,$5,label
5	R_W_RS	addu	W	rs	addu $5,$3,$4	addu $5,$3,$4
					lui $6,1	lui $6,1
					subu $7,$5,$3	ori $7,$5,1
6	R_W_RT	addu	W	rt	addu $5,$3,$4	addu $5,$3,$4
					lui $6,1	A1,A2
					subu $7,$3,$5	beq $4,$5,label
7	I_E_RS	ori	E	rs	ori $4,$3,1	ori $4,$3,1
					beq $4,$3,label	jr $4
8	I_E_RT	ori	E	rt	ori $4,$3,1	
					beq $3,$4,label	
9	I_M_RS	ori	MEM	rs	ori $5,$4,1	ori $5,$4,1
					addu $6,$5,$3	ori $6,$5,1
					beq $5,$3,label	jr $5
10	I_M_RT	ori	MEM	rt	subu $5,$3,$4	subu $5,$3,$4
					addu $6,$3,$5	addu $6,$3,$5
						beq $3,$5,label
11	I_W_RS	ori	W	rs	ori $4,$3,1	ori $4,$3,1
					addu $6,$5,$3	lui $6,1
					subu $7,$4,$6	ori $5,$4,1
12	I_W_RT	ori	W	rt	ori $4,$3,1	
					lui $6,1	
					addu $5,$3,$4	

13	LD_E_RS	addu	E	rs	lw $5,4($0) beq $5,$3,label	lw $31,4($0) jr $31
14	LD_E_RT	addu	E	rt	lw $5,4($0) beq $3,$5,label	

15	LD_M_RS	load	MEM	rs	lw $5,4($0)
					addu $6,$5,$3
					jr $5
						lw $5,4($0)
						ori $6,$5,1
						beq $5,$4,label

16	LD_M_RT	load	MEM	rt	lw $5,4($0)
					subu $6,$3,$5
						lw $5,4($0)
						addu $6,$3,$5
						beq $4,$5,label

17	LD_W_RS	load	W	rs	lw $5,4($0)
					lui $7,1
					addu $6,$5,$3
					jr $5
						lw $5,4($0)
						lui $7,1
						ori $6,$5,1
						beq $3,$5,label

18	LD_W_RT	load	W	rt	lw $5,4($0)
					lui $7,1
					addu $6,$5,$3
						lw $5,4($0)
						lui $7,1
						ori $6,$5,1
						beq $3,$5,label

21	Jal_M_RS	load	MEM	rs	jal label addu $6,$31,$3 jr $31	jal label ori $6,$31,1 beq $31,$4,label2
22	Jal_M_RT	load	MEM	rt	jal label subu $6,$3,$31	jal label lui $6,1 beq $3,$31,label2
23	Jal_W_RS	jal	W	rs	jal label lui $7,1 addu $6,$31,$3 jr $31	jal label lui $7,1 ori $6,$31,1 beq $31,$5,label2
24	Jal_W_RT	jal	W	rt	jal label lui $7,1 addu $6,$3,$31 beq $5,$31,label2	

冲突处理

转发

							ID/EX	EX/MEM			MEM/WB			
							jal	jal	cal_r	cal_i	jal	cal_r	cal_i	load
							0	0	0	0	0	0	0	0
流水线寄存器		源寄存器	涉及指令				31	31	rd	rt	31	rd	rt	rt
IR@D		rs	beq, jr		MFRSD	FRSD	RF.RD1	PC4_E+4	PC4_M+4	A0	A0	M4	M4	M4	M4
		rt	beq		MFRTD	FRTD	RF.RD2	PC4_E+4	PC4_M+4	A0	A0	M4	M4	M4	M4
IR@E		rs	cal_r, cal_i, ld, st		MFRSE	FRSE	RS@E	/	PC4_M+4	A0	A0	M4	M4	M4	M4
		rt	cal_r, st		MFRTE	FRTE	RT@E	/	PC4_M+4	A0	A0	M4	M4	M4	M4
IR@M		rt	st		MFRTM	FRTM	RT@M	/	/	/	/	M4	M4	M4	M4
					转发Mux	控制信号	输入0								

流水线寄存器	源寄存器	涉及指令								
IR@D	rs	beq, jr	MFRSD	FRSD	RF. RD1	AO@M	M4	PC4_E+4	PC4_M+4	M4
	rt	beq	MFRTD	FRTD	RF. RD2	AO@M	M4	PC4_E+4	PC4_M+4	M4
IR@E	rs	cal_r, cal_i, ld, st	MFRSE	FRSE	RS@E	AO@M	M4	PC4_M+4	/	/
	rt	cal_r, st	MFRTE	FRTE	RT@E	AO@M	M4	PC4_M+4	/	/
IR@M	rt	st	MFRTM	FRTM	RT@M	M4	/	/	/	/
			转发Mux	控制信号	输入0	输入1	输入2	输入3	输入4	输入5

暂停

IF/ID当前指令			ID/EX			EX/MEM
指令类型	源寄存器	Tuse	cal_r1/rd	cal_i1/rt	load2/rt	load1/rt
beq	rs/rt	0	pause	pause	pause	pause
cal_r	rs/rt	1			pause	
cal_i	rs	1			pause	
load	rs	1			pause	
store	rs	1			pause	
store	rt	2				
jr	rs	0	pause	pause	pause	pause