



北京航空航天大学
BEIHANG UNIVERSITY

计算机组成原理 Project5 实验报告

Verilog – 支持 11 指令流水线 CPU

{addu,subu,ori,lui,beq,sw,lw,jal,jr,j,nop}

北京航空航天大学

计算机学院

陈麒先

16061160

二〇一七年十二月

郑重声明

关于诚实守信公约：

本实验报告由本人独立完成，全部内容均为本人通过查找互联网资料、翻阅课件、课堂笔记和教材后独立思考的结果。特此声明。

16061160

陈麒先

原创性声明

作业中出现的公式、图片、代码段以及图片的文字注释信息，均为作者原创。抄袭行为在任何情况下都被严格禁止 (COPY is strictly prohibited under any circumstances)！转载或引用须征得作者本人同意，并注明出处！

16061160

陈麒先

目录

第一章 设计架构	2
第一节 流水线CPU顶层架构视图	2
第二节 流水线CPU模块定义说明	2
第三节 流水线CPU控制单元设计	6
第四节 流水线CPU数据通路构造	6
第五节 流水线CPU转发暂停机制	6
第二章 测试验证	10



第一章 设计架构

第一节 流水线 CPU 顶层架构视图

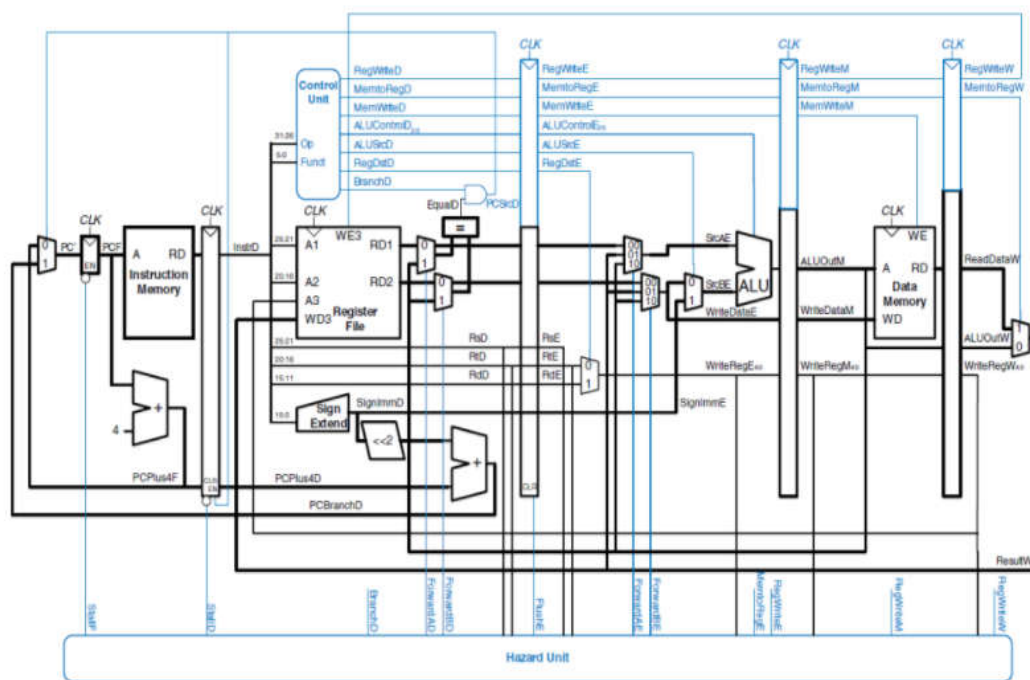


Figure 7.58 Pipelined processor with full hazard handling

第二节 流水线 CPU 模块定义说明

1、IFU

(1) 基本要求

- 起始地址：0x00003000。
- 在实现设计中，为了保持模块的独立性，将 IFU 模块分成 IM、PC、NPC 三个模块的组合。其中 IM 模块单独负责取指令，其输入为 32 位 PC，输出为 32 位指令码；PC 单独建模寄存器，用于在 clk 上升沿更新为 NPC 的值；NPC 通过控制信号的输入，实现对 NPC 的计算。

(2) 端口定义

端口名	方向	描述
clk	In	时钟信号
Reset	In	异步复位信号
if_beq	In	B 类跳转使能信号
if_jr	In	jr 跳转使能信号
if_j	In	j 跳转使能信号
If_jal	In	jal 跳转使能信号
zero	In	相等条件信号
Offset	In	地址偏移量
Instr	Out	输出 32 位指令码

(3) 功能说明

序号	功能	描述
1	同步复位	Reset 信号有效时, PC 复位
2	b 类跳转	b 类信号和条件信号同时有效时, 执行跳转
3	j 类跳转	j 类信号有效时, 执行跳转
4	jr 类跳转	jr 信号有效时, 执行跳转至 jr_PC
5	PC4	PC 决定所取指令的地址, 每个周期 PC+4

2、GPR

(1) 基本要求

- 用具有写使能的寄存器实现, 寄存器总数为 32 个。
- 0 号寄存器的值保持为 0。

(2) 端口定义

端口名	方向	描述
clk	In	时钟信号
reset	In	异步复位信号
RA	[4:0] In	读总线 A 地址
RB	[4:0] In	读总线 B 地址
RW	[4:0] In	写寄存器地址
WD	[31:0] In	写数据输入

WE	In	写使能
BusA	[31:0] Out	总线 A 输出
BusB	[31:0] Out	总线 B 输出

(3) 功能说明

序号	功能	描述
1	同步复位	Reset 信号有效时, GRF 复位
2	读寄存器	根据 RA, RB 所指示的地址, 读出对应寄存器的值
3	写寄存器	根据 RW 所指示的地址, 将 WD 的数据写入对应寄存器
4	0 号寄存器	0 号寄存器不连接数据写入端口, 输出接地

3、ALU

(1) 基本要求

- 提供 32 位加、减、或运算及大小比较功能。
- 可以不支持溢出（不检测溢出）。

(2) 端口定义

端口名	方向	描述
A	[31:0] In	ALU 第一个操作数
B	[31:1] In	ALU 第二个操作数
ALUOp	[1:0] In	ALU 控制信号
ALUOut	[31:0] Out	ALU 计算结果

(3) 功能说明

序号	功能	描述
1	加	ALUOp = 00, 两个操作数相加
2	减	ALUOp = 01, 两个操作数相减
3	或	ALUOp = 10, 两个操作数按位或

4、DM

(1) 基本要求

- 容量为 32bit * 1024。
- 起始地址：0x00000000。

(2) 端口定义

端口名	方向	描述
clk	In	时钟信号
reset	In	异步复位信号
WE	In	写使能信号
WD	In	写入数据
address	In	写入地址
RD	Out	读内存数据

(3) 功能说明

序号	功能	描述
1	同步复位	Reset 信号有效时，DM 复位
2	读内存数据	WE 信号为 0 时，读内存中 Address 指示的地址数据
3	写内存数据	WE 信号为 1 时，向内存中 Address 指示的地址写数据

5、EXT:

(1) 基本要求

- 注意扩展方式

(2) 端口定义

端口名	方向	描述
Imm16	[15:0] In	输入待扩展的 16 位立即数
ExtOp	[1:0] In	扩展信号
Ext32	[31:0] Out	扩展结果输出

(3) 功能说明

序号	功能	描述
1	零扩展	当 ExtOp==2'b00 时，执行零扩展
2	符号扩展	当 ExtOp==2'b01 时，执行符号扩展

3	高位扩展	当 ExtOp==2'b10 时, 执行高位扩展
---	------	--------------------------

6、CMP:

(1) 基本要求

- 作为跳转类指令的条件判断信号, 需前移至 F 级。

(2) 端口定义

端口名	方向	描述
A	[31:0] In	比较数 A
B	[31:0] In	比较数 B
equal	Out	相等比较结果

(3) 功能说明

序号	功能	描述
1	相等比较	当 equal 输出高电平时, 说明两输入操作数相等

第三节 流水线 CPU 控制单元设计

(1) 端口定义

端口名	方向	描述
OpCode	[5:0] In	指令高六位 Ins [31:26]
Func	[5:1] In	指令低六位 Ins [5:0]
RegDst	Out	若为高电平, 选择 Rd, 否则选择 Rt 作为 GPR 的 RW 输入
ALUSrc	Out	若为高电平, 选择扩展数字, 否则选择 BusB 作为 ALU 的第二位输入
MemToReg	Out	若为高电平, 选择 DM, 否则选择 ALUOut 作为 GPR 的 WD 输入
RegWrite	Out	若为高电平, 则对 GPR 进行写操作, 否则写使能无效
MemWrite	Out	若为高电平, 则对 DM 进行写操作, 否则对 DM 进行读操作
If_beq	Out	若为高电平且 zero 为高电平, 则跳转
ExtOp[0]	Out	共同决定 EXT 的行为
ExtOp[1]	Out	
ALUOp[0]	Out	共同决定 ALU 的行为
ALUOp[1]	Out	

(2) 控制器真值表

	addu	subu	ori	lui	lw	sw	beq	jal	j	jr
OpCode	000000	000000	001101	001111	100011	101011	000100	000011	000010	000000
Func	100001	000000	N / A							001000
RegDst	1	1	0	0	0	X	X	X	X	X
RegWrite	1	1	1	1	1	0	0	1	0	0
ALUSrc	0	0	1	1	1	1	0	0	X	X
MemWrite	0	0	0	0	0	1	0	0	0	0
MemToReg	0	0	0	0	1	X	X	X	X	X
ExtOp[1]	X	X	0	1	0	0	0	X	X	X
ExtOp[0]	X	X	0	0	1	1	1	X	X	X
ALUOp[1]	0	0	1	0	0	0	X	X	X	X
ALUOp[0]	0	1	0	0	0	0	X	X	X	X

第四节 数据通路构造表

指令		addu	subu	ori	lui	lw	sw
F级	pc	pc4	pc4	pc4	pc4	pc4	pc4
	im	pc	pc	pc	pc	pc	pc
	pc4	pc	pc	pc	pc	pc	pc
F-D	D-Reg	IR@D	im	im	im	im	im
		PC4@D	pc4	pc4	pc4	pc4	pc4
D级	GPR	RA	IR@D[Rs]	IR@D[Rs]	IR@D[Rs]	IR@D[Rs]	IR@D[Rs]
		RB	IR@D[Rt]	IR@D[Rt]			IR@D[Rt]
	EXT				IR@D[IMM]	IR@D[IMM]	IR@D[IMM]
	NPC	pc4					
		jr_pc					
		imm					
		IR					
	CMP	D1					
		D2					
	pc8						
D-E	E-Reg	IR@E	IR@D	IR@D	IR@D	IR@D	IR@D
		PC8@E					
		RA@E	GPR. RA	GPR. RA	GPR. RA	GPR. RA	GPR. RA
		RB@E	GPR. RB	GPR. RB			GPR. RB
		ext@E			EXT	EXT	EXT
E级	ALU	A	RA@E	RA@E	RA@E	RA@E	RA@E
		B	RB@E	RB@E	ext@E	ext@E	ext@E
			N/A				
E-M	M-Reg	IR@M	IR@E	IR@E	IR@E	IR@E	IR@E
		PC8@M					
		AO@M	ALU	ALU	ALU	ALU	ALU
		WM@M					RB@E
M级	DM	add				AO@M	AO@M
		WM					WM@M
M-W	W-Reg	IR@W	IR@E	IR@E	IR@E	IR@E	IR@E
		PC8@W					
		AO@W	AO@E	AO@E	AO@E	AO@E	
		DO@W				DM	
W级	GPR	RW	IR@W[Rd]	IR@W[Rd]	IR@W[Rt]	IR@W[Rt]	
		WD	AO@W	AO@W	AO@W	DO@W	

指令	beq	jal	jr	j	MUX	0	1	2
F 级	pc4 npc	npc	npc	npc	pc_in	pc4@F	NPC	
	pc	pc	pc	pc	pc			
	pc	pc	pc	pc	pc			
F-D	im	im	im	im	im			
	pc4	pc4	pc4	pc4	pc4			
D 级	IR@D[Rs]		IR@D[Rs]		IR@D[Rs]			
	IR@D[Rt]				IR@D[Rt]			
	IR@D[IMM]				IR@D[IMM]			
	PC4@D	PC4@D		PC4@D	PC4@D			
			GPR. RA		MFAD	GPR. RA	F1	F2
	EXT				EXT			
		IR@D		IR@D	IR@D			
	IR@D[Rs]				MFAD			
	IR@D[Rt]				MFBD			
		PC4@D			PC4@D			
D-E	IR@D	IR@D	IR@D	IR@D	IR@D			
		pc8			pc8			
					GPR. RA			
					GPR. RB			
					EXT			
E 级					MFAE	RA@E	WD@W	AO@M
					ALUB	MFBE	ext@E	
	N/A				MFBE	RB@E	WD@W	AO@M
E-M	IR@E	IR@E	IR@E	IR@E	IR@E			
		PC8@E			PC8@E			
					ALU			
					RB@E			
M 级					AO@M			
					MFBM	WM@M	WD@W	
M-W	IR@E	IR@E	IR@E	IR@E	IR@E			
		PC8@M			PC8@M			
					AO@E			
					DM			
W 级		31			regwrite	IR@W[Rd]	IR@W[Rt]	31
		PC8@W			writeback	AO@W	DO@W	PC8@W

第五节 转发暂停控制机制

1. 概述

流水线各流水级间冲突主要体现在寄存器的占用冲突，因此结合分布式译码特性，我们容易获知各流水级所执行指令对寄存器的存取特征。根据这些特征，我们可以分析出转发暂停的构造机制。

2. 转发暂停构造表

		E				M				W			
		RegWrite		MemToReg		RegWrite		MemToReg		RegWrite		MemToReg	
		Rs	Rt	Rs	Rt	Rs	Rt	Rs	Rt	Rs	Rt	Rs	Rt
D	Rs	S	—	S	—	F	—	S	—	F	—	F	—
	Rt	—	S	—	S	—	F	—	S	—	F	—	F
E	Rs	—	—	—	—	F	—	S	—	F	—	F	—
	Rt	—	—	—	—	—	F	—	S	—	F	—	F
M	Rt	—	—	—	—	—	—	—	—	—	F	—	F

第二章 测试验证

流水线覆盖性分析测试用例构造表

1. addu 测试样例表

用例编号	测试类型	前序指令	冲突位置	冲突寄存器	测试样例
1	R-M-RS	subu	MEM	rs	subu \$1, \$2, \$3 addu \$4, \$1, \$2
2	R-M-RT	subu	MEM	rt	subu \$1, \$2, \$3 addu \$4, \$2, \$1
3	R-W-RS	subu	WB	rs	subu \$1, \$2, \$3 nop addu \$4, \$1, \$2
4	R-W-RT	subu	WB	rt	subu \$1, \$2, \$3

					nop addu \$4, \$2, \$1
5	R-W-RS	subu	WB	rs	subu \$1, \$2, \$3 nop nop addu \$4, \$2, \$1
6	R-W-RT	subu	WB	rt	subu \$1, \$2, \$3 nop nop addu \$4, \$2, \$1
7	I-M-RS	ori	MEM	rs	ori \$1, 20 addu \$4, \$1, \$2
8	I-M-RT	ori	MEM	rt	ori \$1, 20 addu \$4, \$2, \$1
9	I-W-RS	ori	WB	rs	ori \$1, 20 nop addu \$4, \$1, \$2
10	I-W-RT	ori	WB	rt	ori \$1, 20 nop addu \$4, \$2, \$1
11	I-W-RS	ori	WB	rs	ori \$1, 20 nop nop addu \$4, \$1, \$2
12	I-W-RT	ori	WB	rt	ori \$1, 20 nop nop addu \$4, \$2, \$1
13	LW-M-RS	lw	MEM	rs	lw \$1, 0(\$0)

					addu \$4, \$1, \$2
14	LW-M-RT	lw	MEM	rt	lw \$1, 0(\$0) addu \$4, \$2, \$1
15	LW-WB-RS	lw	WB	rs	lw \$1, 0(\$0) nop addu \$4, \$1, \$2
16	LW-WB-RT	lw	WB	rt	lw \$1, 0(\$0) nop addu \$4, \$2, \$1
17	LW-W-RS	lw	WB	rs	lw \$1, 0(\$0) nop nop addu \$4, \$1, \$2
18	LW-W-RT	lw	WB	rt	lw \$1, 0(\$0) nop nop addu \$4, \$2, \$1
19	J-M-RS	jal	MEM	rs	jal loop addu \$1, \$31, \$2 loop:
20	J-M-RT	jal	MEM	rt	jal loop addu \$1, \$2, \$31 loop:
21	J-W-RS	jal	WB	rs	jal loop ori \$4, 1 loop: addu \$1, \$31, \$2
22	J-W-RT	jal	WB	rt	jal loop ori \$4, 1

					loop: addu \$1, \$2, \$31
--	--	--	--	--	------------------------------

2. subu 测试样例表

用例编号	测试类型	前序指令	冲突位置	冲突寄存器	测试样例
1	R-M-RS	addu	MEM	rs	addu \$1, \$2, \$3 subu \$4, \$1, \$2
2	R-M-RT	subu	MEM	rt	subu \$1, \$2, \$3 subu \$4, \$2, \$1
3	R-W-RS	addu	WB	rs	addu \$1, \$2, \$3 nop addu \$4, \$1, \$2
4	R-W-RT	subu	WB	rt	subu \$1, \$2, \$3 nop subu \$4, \$2, \$1
5	R-W-RS	addu	WB	rs	addu \$1, \$2, \$3 nop nop addu \$4, \$1, \$2
6	R-W-RT	subu	WB	rt	subu \$1, \$2, \$3 nop nop subu \$4, \$2, \$1
7	I-M-RS	ori	MEM	rs	ori \$1, 10 subu \$4, \$1, \$2
8	I-M-RT	ori	MEM	rt	ori \$1, 2 subu \$4, \$2, \$1
9	I-W-RS	ori	WB	rs	ori \$1, 10 nop subu \$4, \$1, \$2

10	I-W-RT	ori	WB	rt	ori \$1, 2 nop subu \$4, \$2, \$1
11	I-W-RS	ori	WB	rs	ori \$1, 10 nop nop subu \$4, \$1, \$2
12	I-W-RT	ori	WB	rt	ori \$1, 2 nop nop subu \$4, \$2, \$1
13	LW-M-RS	lw	MEM	rs	lw \$1, 0(\$0) subu \$4, \$1, \$2
14	LW-M-RT	lw	MEM	rt	lw \$1, 0(\$0) subu \$4, \$2, \$1
15	LW-W-RS	lw	WB	rs	lw \$1, 0(\$0) nop subu \$4, \$1, \$2
16	LW-W-RT	lw	WB	rt	lw \$1, 0(\$0) nop subu \$4, \$2, \$1
17	LW-W-RS	lw	WB	rs	lw \$1, 0(\$0) nop nop subu \$4, \$1, \$2
18	LW-W-RT	lw	WB	rt	lw \$1, 0(\$0) nop nop subu \$4, \$2, \$1

19	J-M-RS	jal	MEM	rs	jal loop subu \$1, \$31, \$2 loop:
20	J-M-RT	jal	MEM	rt	jal loop subu \$1, \$2, \$31 loop:
21	J-W-RS	jal	WB	rs	jal loop ori \$4, 1 loop: subu \$1, \$31, \$2
22	J-W-RT	jal	WB	rt	jal loop ori \$4, 1 loop: subu \$1, \$2, \$31

3. ori 测试样例表

用例编号	测试类型	前序指令	冲突位置	冲突寄存器	测试样例
1	R-M-RS	addu	MEM	rs	addu \$1, \$2, \$3 ori \$4, \$1, 7
3	R-W-RS	addu	WB	rs	addu \$1, \$2, \$3 nop ori \$4, \$1, 7
5	I-M-RS	ori	MEM	rs	ori \$1, \$2, 2 ori \$3, \$1, 8
6	I-W-RS	ori	WB	rs	ori \$1, \$2, 2 nop ori \$3, \$1, 8
7	LW-M-RS	lw	MEM	rs	lw \$1, 0(\$0)

					ori \$3, \$1, 2
8	LW-W-RS	lw	WB	rs	lw \$1, 0(\$0) nop ori \$3, \$1, 2
9	LW-W-RS	lw	WB	rs	lw \$1, 0(\$0) nop nop ori \$3, \$1, 2
10	J-M-RS	jal	MEM	rs	jal loop ori \$31, \$31, 1 loop:
11	J-W-RS	jal	WB	rs	jal loop ori \$1, \$1, 1 loop: ori \$31, \$31, 1

4. j 测试样例表

用例编号	测试类型	测试样例
1	J	ori \$2, 5 ori \$1, 1 addu \$1, \$1, \$2 j exit ori \$3, 1 exit:

5. lw 测试样例表

用例编号	测试类型	前序指令	冲突位置	冲突寄存器	测试样例
1	R-M-RS	addu	MEM	rs	addu \$1, \$2, \$3 lw \$4, 0(\$1)

2	R-W-RS	addu	WB	rs	addu \$1, \$2, \$3 nop lw \$4, 0(\$1)
3	R-W-RS	addu	WB	rs	addu \$1, \$2, \$3 nop nop lw \$4, 0(\$1)
4	I-M-RS	ori	MEM	rs	ori \$1, 8 lw \$4, 0(\$1)
5	I-W-RS	ori	WB	rs	ori \$1, 8 nop lw \$4, 0(\$1)
6	I-W-RS	ori	WB	rs	ori \$1, 8 nop nop lw \$4, 0(\$1)
7	LW-M-RS	lw	MEM	rs	lw \$1, 4(\$0) lw \$4, 0(\$1)
8	LW-W-RS	lw	WB	rs	lw \$1, 4(\$0)

					nop lw \$4, 0(\$1)
9	LW-W-RS	lw	WB	rs	lw \$1, 4(\$0) nop nop lw \$4, 0(\$1)
10	J-M-RS	jal	MEM	rs	jal loop lw \$1, 0(\$31) loop:
11	J-M-RT	jal	WB	rs	jal loop loop: lw \$1, 0(\$31)

6. sw 测试样例表

用例编号	测试类型	前序指令	冲突位置	冲突寄存器	测试样例
1	R-M-RS	addu	MEM	rs	addu \$1, \$2, \$3 sw \$2, 0(\$1)
2	R-W-RS	addu	WB	rs	addu \$1, \$2, \$3 nop sw \$2, 0(\$1)
3	R-W-RS	addu	WB	rs	addu \$1, \$2, \$3 nop

					nop sw \$2, 0(\$1)
4	I-M-RS	ori	MEM	rs	ori \$1, 8 sw \$2, 0(\$1)
5	I-W-RS	ori	WB	rs	ori \$1, 8 nop sw \$2, 0(\$1)
6	I-W-RS	ori	WB	rs	ori \$1, 8 nop nop sw \$2, 0(\$1)
7	LW-M-RS	lw	MEM	rs	lw \$1, 4(\$0) sw \$2, 0(\$1)
8	LW-W-RS	lw	WB	rs	lw \$1, 4(\$0) nop sw \$2, 0(\$1)
9	LW-W-RS	lw	WB	rs	lw \$1, 4(\$0) nop nop sw \$2, 0(\$1)
10	J-M-RS	jal	MEM	rs	jal loop sw \$1, 0(\$31) loop:
11	J-M-RT	jal	WB	rs	jal loop loop: sw \$1, 0(\$31)

7. beq 测试样列表

用例编号	测试类型	前序指令	冲突位置	冲突寄存器	测试样例
1	R-M-RS	addu	MEM	rs	addu \$1, \$3, \$0 beq \$1, \$3, exit ori \$4, 1 ori \$4, 2 exit:
2	R-M-RT	addu	MEM	rt	addu \$1, \$3, \$0 beq \$3, \$1, exit ori \$4, 1 ori \$4, 2 exit:
3	R-W-RS	addu	WB	rs	addu \$1, \$3, \$0 nop beq \$1, \$3, exit ori \$4, 1 ori \$4, 2 exit:
4	R-W-RT	addu	WB	rt	addu \$1, \$3, \$0 nop beq \$3, \$1, exit ori \$4, 1

					ori \$4, 2 exit:
5	R-W-RS	addu	WB	rs	addu \$1, \$3, \$0 nop nop beq \$1, \$3, exit ori \$4, 1 ori \$4, 2 exit:
6	R-W-RT	addu	WB	rt	addu \$1, \$3, \$0 nop nop beq \$3, \$1, exit ori \$4, 1 ori \$4, 2 exit:
7	I-M-RS	ori	MEM	rs	ori \$1, 3 beq \$1, \$3, exit ori \$4, 1 ori \$4, 2 exit:
8	I-M-RT	ori	MEM	rt	ori \$1, 3 beq \$3, \$1, exit

					ori \$4, 1 ori \$4, 2 exit:
9	I-W-RS	ori	WB	rs	ori \$1, 3 nop beq \$1, \$3, exit ori \$4, 1 ori \$4, 2 exit:
10	I-W-RT	ori	WB	rt	ori \$1, 3 nop beq \$3, \$1, exit ori \$4, 1 ori \$4, 2 exit:
11	I-W-RS	ori	WB	rs	ori \$1, 3 nop nop beq \$1, \$3, exit ori \$4, 1 ori \$4, 2 exit:
12	I-W-RT	ori	WB	rt	ori \$1, 3 nop nop beq

					\$3, \$1, exit ori \$4, 1 ori \$4, 2 exit:
13	LW-M-RS	lw	MEM	rs	addu \$4, \$2, \$3 lw \$1, 8(\$0) beq \$1, \$4, exit ori \$5, 1 ori \$5, 2 exit:
14	LW-M-RT	lw	MEM	rt	addu \$4, \$2, \$3 lw \$1, 8(\$0) beq \$4, \$1, exit ori \$5, 1 ori \$5, 2 exit:
15	LW-WB-RS	lw	WB	rs	addu \$4, \$2, \$3 lw \$1, 8(\$0) nop beq \$1, \$4, exit ori \$5, 1 ori \$5, 2 exit:

16	LW-WB-RT	lw	WB	rt	addu \$4, \$2, \$3 lw \$1, 8(\$0) nop beq \$4, \$1, exit ori \$5, 1 ori \$5, 2 exit:
17	LW-W-RS	lw	WB	rs	addu \$4, \$2, \$3 lw \$1, 8(\$0) nop nop beq \$1, \$4, exit ori \$5, 1 ori \$5, 2 exit:
18	LW-W-RT	lw	WB	rt	addu \$4, \$2, \$3 lw \$1, 8(\$0) nop nop beq \$4, \$1, exit ori \$5, 1 ori \$5, 2 exit:

8. jr 测试样例表

用例编号	测试类型	前序指令	冲突位置	冲突寄存器	测试样例
1	R-M-RS	addu	MEM	rs	ori \$1, 4 jal loop addu \$31, \$31, \$1 ori \$2, 1 ori \$3, 1 j exit ori \$8, 1 loop: jr \$31 ori \$4, 1 exit:
2	R-W-RS	addu	WB	rs	ori \$1, 4 jal loop addu \$31, \$31, \$1 ori \$2, 1 ori \$3, 1 j exit ori \$8, 1 loop: ori \$5, 1 jr \$31 ori \$4, 1 exit:
3	R-W-RS	subu	WB	rs	ori \$1, 4 jal loop

					<pre> addu \$31, \$31, \$1 ori \$2, 1 ori \$3, 1 j exit ori \$8, 1 loop: ori \$5, 1 ori \$6, 1 jr \$31 ori \$4, 1 exit: </pre>
4	I-M-RS	ori	MEM	rs	<pre> ori \$1, 4 ori \$1, 4 ori \$1, 4 jal loop ori \$31, 8 ori \$2, 1 ori \$3, 1 j exit ori \$8, 1 loop: jr \$31 ori \$4, 1 exit: </pre>
5	I-W-RS	ori	WB	rs	<pre> ori \$1, 4 ori \$1, 4 ori \$1, 4 jal loop </pre>

					ori \$31, 8 ori \$2, 1 ori \$3, 1 j exit ori \$8, 1 loop: ori \$5, 1 jr \$31 ori \$4, 1 exit:
6	I-W-RS	ori	WB	rs	ori \$1, 4 ori \$1, 4 ori \$1, 4 jal loop ori \$31, 8 ori \$2, 1 ori \$3, 1 j exit ori \$8, 1 loop: ori \$5, 1 ori \$6, 1 jr \$31 ori \$4, 1 exit:
7	LW-M-RS	lw	MEM	rs	ori \$1, 4 ori \$2, 0x301c sw \$2, 0(\$0) jal loop

					lw \$31, 0(\$0) ori \$2, 1 ori \$3, 1 j exit ori \$8, 1 loop: jr \$31 ori \$4, 1 exit:
8	LW-WB-RS	lw	WB	rs	ori \$1, 4 ori \$2, 0x301c sw \$2, 0(\$0) jal loop lw \$31, 0(\$0) ori \$2, 1 ori \$3, 1 j exit ori \$8, 1 loop: ori \$5, 1 jr \$31 ori \$4, 1 exit:
9	LW-W-RS	lw	WB	rs	ori \$1, 4 ori \$2, 0x301c sw \$2, 0(\$0) jal loop lw \$31, 0(\$0) ori \$2, 1

					ori \$3, 1 j exit ori \$8, 1 loop: ori \$5, 1 ori \$6, 1 jr \$31 ori \$4, 1 exit:
10	J-M-RS	jal	WB	rs	ori \$1, 4 jal loop addu \$31, \$31, \$1 ori \$2, 1 ori \$3, 1 j exit ori \$8, 1 loop: jr \$31 ori \$4, 1 exit:
11	J-W-RS	jal	WB	rs	ori \$1, 4 jal loop addu \$31, \$31, \$1 ori \$2, 1 ori \$3, 1 j exit ori \$8, 1

					loop: ori \$5,1 jr \$31 ori \$4,1 exit:
--	--	--	--	--	---

