

P5 Verilog 流水线处理器

一、模块定义

使用架构 F-IFID-D-IDEXE-E-EXEMEM-M-MEMWB-W(D)，按阶段标记为 0-01-1-12-2-23-3-34-4，对每个阶段端口进行阶段标示性命名。

主模块：mips.v

文件	模块接口定义
mips.v	input clk; //clock input reset; //reset

1. PC.v

表 1-1 PC.v 文件信号定义

文件	模块接口定义
PC.v	input clk; //clock input reset; //reset input [31:0] npc_0; output [31:0] pc_0;

2. IM.v

表 2-1 IM.v 文件信号定义

文件	模块接口定义
IM.v	input [31:0] imaddr; output [31:0] instr_0;

3. ADD4.v

表 3-1 IM.v 文件信号定义

文件	模块接口定义
IM.v	input [31:0] pc_0; output [31:0] pc4_0;

4. D.v

表 4-1 D.v 文件信号定义

文件	模块接口定义
D.v	input clk; input [31:0] IR_0; input [31:0] pc4_0; input [31:0] pc_0; output [31:0] IR_01;

	output [31:0] pc4_01; output [31:0] pc_01;
--	---

5. GRF. v

表 5-1 GRF. v 文件接口定义

文件	模块接口定义
GRF.v	input [31:0] WPC; input clk; //clock input reset; //reset input [4:0] A1_1; input [4:0] A2_1; input [4:0] A3_4; //write back stage input RegWr_4; input [31:0] WD_4; //write back stage output [31:0] RD1_1; output [31:0] RD2_1;

6. EXT. v

表 6-1 EXT. v 文件接口定义

文件	模块接口定义
EXT.v	input ExtOp_1; input [15:0] Imm_1; output [31:0] E32_1;

7. NPC. v

表 7-1 NPC. v 文件信号定义

文件	模块接口定义
NPC.v	input [31:0] pc_1; input [31:0] pc4_1; //already+4 input zero; //for beq judgement input [2:0] npc_sel; input [25:0] Imm26_1; output [31:0] npc_1;

8. CMP. v

表 8-1 CMP. v 文件接口定义

文件	模块接口定义
CMP.v	input [31:0] D1_1; input [31:0] D2_1; output zero; //for beq

9. E. v

表 9-1 E. v 文件信号定义

文件	模块接口定义
E.v	<pre> input clk; input [31:0] V1_1; input [31:0] V2_1; input [5:0] A1_1; input [5:0] A2_1; input [5:0] A3_1; input [31:0] E32_1; input [31:0] pc4_1; input [31:0] pc_1; output [31:0] V1_12; output [31:0] V2_12; output [5:0] A1_12; output [5:0] A2_12; output [5:0] A3_12; output [31:0] E32_12; output [31:0] pc4_12; output [31:0] pc_12; </pre>

10. ALU.v

表 10-1 ALU.v 文件接口定义

文件	模块接口定义
ALU.v	<pre> input [2:0] ALUctr_2; input [31:0] A_2; input [31:0] B_2; output [31:0] C_2; </pre>

11. M.v

表 11-1 M.v 文件接口定义

文件	模块接口定义
M.v	<pre> input clk; input [31:0] V2_2; input [5:0] A2_2; input [5:0] A3_2; input [31:0] AO_2; input [31:0] pc4_2; input [31:0] pc_2; output [31:0] V2_23; output [5:0] A2_23; output [5:0] A3_23; output [31:0] AO_23; output [31:0] pc4_23; output [31:0] pc_23; </pre>

12. DM. v

表 12-1 DM. v 文件接口定义

文件	模块接口定义
DM.v	input [31:0] DPC; input clk; input reset; input MemWr_3; input [31:0] dmAd_3; input [31:0] Din_3; output [31:0] DO_3;

13. W. v

表 13-1 W. v 文件接口定义

文件	模块接口定义
W.v	input clk; input [5:0] A3_3; input [31:0] AO_3; input [31:0] DR_3; input [31:0] pc4_3; input [31:0] pc_3; output [5:0] A3_34; output [31:0] AO_34; output [31:0] DR_34; output [31:0] pc4_34; output [31:0] pc_34;

14. Ctr_D. v

表 14-1 Ctr_D. v 文件接口定义

文件	模块接口定义
Ctr_D.v	input clk; input [5:0] op_01; input [5:0] func_01; output [1:0] MA3E;//choose the right value output [1:0] MPCF; // output ExtOp_1; output [2:0] npc_sel; output [5:0] op_12; output [5:0] func_12;

16. Ctr_E. v

表 16-1 Ctr_E. v 文件接口定义

文件	模块接口定义
Ctr_E.v	<pre> input clk; input [5:0] op_12; input [5:0] func_12; output MALUBE; output [2:0] ALUctr; output [5:0] op_23; output [5:0] func_23; </pre>

17. Ctr_M.v

表 17-1 Ctr_M.v 文件接口定义

文件	模块接口定义
Ctr_M.v	<pre> input clk; input [5:0] op_23; input [5:0] func_23; output MemWr_3; output [1:0] MRFWDW; output [5:0] op_34; output [5:0] func_34; </pre>

18. Ctr_W.v

表 18-1 Ctr_W.v 文件接口定义

文件	模块接口定义
Ctr_W.v	<pre> input clk; input [5:0] op_34; input [5:0] func_34; output [1:0] MRFWDW; output RegWr_4; </pre>

19. decoder.v

表 19-1 decoder.v 文件接口定义

文件	模块接口定义
decoder.v	<pre> input [5:0] op, input [5:0] func, output addu, output subu, output ori, output lui, output lw, output sw, </pre>

	output beq, output j, output jal, output jr;
--	---

20. STALL. v

文件	模块接口定义
STALL.v	<pre> module STALL(input [31:0] instr_D, input [4:0] A3_E, input [4:0] A3_M, input RegWr_E, input RegWr_M, input [1:0] Tnew_E, input [1:0] Tnew_M, input [1:0] Tnew_W, output stall); </pre>

21. ForwardUnit. v

文件	模块接口定义
ForwardUnit.v	<pre> module ForwardUnit(input [31:0] instr_D, input [4:0] A3_E, input [4:0] A3_M, input [4:0] A3_W, input [4:0] A1_E, input [4:0] A2_E, input [4:0] A2_M, input MemWr_D, //没用 input RegWr_E, input RegWr_M, input RegWr_W, input [1:0] Tnew_E, input [1:0] Tnew_M, output [1:0] FRD1D, output [1:0] FRD2D, output [1:0] FALUAE, output [1:0] FALUBE, output FWDM); </pre>

22. MUX. v

文件	模块接口定义
----	--------

MUX.v	<p>将所有 MUX 归结到一个文件中调用</p> <pre> module MUX_Four2one(sel, option_0, option_1, option_2, option_3, result); parameter WIDTH = 32; input [1:0] sel; input [WIDTH:1] option_0; input [WIDTH:1] option_1; input [WIDTH:1] option_2; input [WIDTH:1] option_3; output reg [WIDTH:1] result; module MUX_Three2one(sel, option_0, option_1, option_2, result); parameter WIDTH = 32; input [1:0] sel; input [WIDTH:1] option_0; input [WIDTH:1] option_1; input [WIDTH:1] option_2; output reg [WIDTH:1] result; module MUX_Two2one (sel, option_0, option_1, result); parameter WIDTH = 32; input sel; input [WIDTH:1] option_0; input [WIDTH:1] option_1; output reg [WIDTH:1] result; </pre>
-------	---

二、数据通路参数定义

1. 功能 MUX 信号

信号	定义
PC_sel	PC 值选择 0: ADD4 1: NPC
A3_sel[1:0]	RF 回写地址选择 00: rd (IR[rd]@D) 01: rt (IR[rt]@D) 10: 0x1f
ALUBE	ALU 操作数选择 0: RF.RD2 (V2@E) 1: Ext (E32@E)
RFWDW_sel	RF 回写数据选择

[1:0]	00: ALU (AO@W) 01: DM.DO (DR@W) 10: pc4 (PC4@W)
ExtOp	0: 零扩展 1: 符号扩展
npc_sel [2:0]	000: 非跳转 001: BEQ 010: JAL 011: JR 100: J
ALUctr [2:0]	000: ADDU 001: SUBU 010: OR 011: LUI

2. 功能信号真值表

信号名称	指令 1	指令 2	指令 3	指令 4	指令 5	指令 6	指令 7	指令 8	指令 9	指令 10	指令 11
(func)	10 0001	10 0011								00 1000	
(op)	00 0000	00 0000	00 1101	10 0011	10 1011	00 0100	00 1111	00 0000	00 0011	00 0000	00 0010
	ADDU	SUBU	ORI	LW	SW	BEQ	LUI	NOP	JAL	JR	J
PC_sel	0	0	0	0	0	x	0	x	1	1	1
A3_sel[1:0]	01	01	00	00	x	x	00	x	10	x	x
ALUBE_sel	0	0	1	1	1	0	1	x	x	x	X
RFWDW_sel 1 [1:0]	00	00	00	01	x	x	00	x	10	x	x
RegWr	1	1	1	1	0	0	1	0	1	0	0
MemWr	0	0	0	0	1	0	0	0	0	0	0
nPC_sel[2:0]	000	000	000	000	000	001	000	000	010	011	100
ExtOp	x	x	0	1	1	x	0	x	x	x	x
ALUctr[2:0]	ADD 000	SUB 001	OR 010	ADD 000	ADD 000	x	LUI 011	x	x	x	x

三、策略矩阵

Tuse			Tnew				
指令	rs	rt	指令	部件	E	M	W
addu	1	1	addu	ALU	1	0	0
subu	1	1	subu				
ori	1		ori				
lui		1	lui	DM	2	1	0
lw	1		lw				
sw	1	2	sw				
beq			beq	PC	0	0	0
jr			jr				
jal			jal				
j			j				

RS	Tuse \ Tnew	E			M			W		
		ALU	DM	PC	ALU	DM	PC	ALU	DM	PC
		1	2	0	0	1	0	0	0	0
	0	S	S	F	F	S	F	F	F	F
	1	F	S	F	F	F	F	F	F	F

RT	Tuse \ Tnew	E			M			W		
		ALU	DM	PC	ALU	DM	PC	ALU	DM	PC
		1	2	0	0	1	0	0	0	0
	0	S	S	F	F	S	F	F	F	F
	1	F	S	F	F	F	F	F	F	F
	2	F	F	F	F	F	F	F	F	F

四、转发构造

1. 需求

D 级对 rs, rt 寄存器的需求, E 级对 rs, rt 寄存器的需求, 以及 M 级对 rt 寄存器的需求。在需求点添加转发 MUX, 每个需求点后的各级流水线寄存器均向其转发数据。注意跳转指令在 D 级的转发。

2. 供给

一般为 M 级的 ALU 结果, W 级的 dm 输出结果以及还未写入 RF 的结果。

信号名	控制信号	D	E	M	W	需求
MFRD1D	FRD1D	RF.RD1 0	V1@E 3	AOM_PC8@M 2	WD@W 1	CMP.D1@D V1@E
MFRD2D	FRD2D	RF.RD2 0	V2@E 3	AOM_PC8@M 2	WD@W 1	CMP.D2@D V2@E
MFALUAE	FALUAE		V1@E 0	AOM_PC8@M 2	WD@W 1	ALU.A@E
MFALUBE	FALUBE		V2@E 0	AOM_PC8@M 2	WD@W 1	MALUB.0@E V2@M

MFWDM	FWDM			V2@M 0	WD@W 1	WD@M
-------	------	--	--	-----------	-----------	------

3. 转发条件

读写寄存器编号相同，数据为最新生成。

注意不对写入地址为\$0 的指令转发。

4. 优先级

端口数字越大优先级越高，越是新产生的数据。

五、暂停构造

1. 暂停条件

在 D 级进行判断， $T_{new} > T_{use}$ ，新数据将写入 RF，写入寄存器编号与读取寄存器编号相同。

2. 暂停处理

冻结 PC 值。

冻结 D 级流水寄存器。

清零 E 级流水寄存器。

六、测试程序

1. 暂停

类型	测试类型	前序指令	冲突位置	冲突寄存器	测试序列
R 型	LD-E-RS	lw	D	RS	lw \$4, 0(\$5) addu \$4, \$4, \$5
	LD-E-RT	lw	D	RT	lw \$4, 0(\$5) addu \$4, \$5, \$4
I 型	LD-E-RS	lw	D	RS	lw \$4, 0(\$5) ori \$5, \$4, 0xffff
LD 型	LD-E-RS	lw	D	RS	lw \$4, 0(\$5) lw \$3, 4(\$4)
ST 型	LD-E-RS	lw	D	RS	lw \$4, 0(\$5) sw \$3, 4(\$4)
JR	R-E-RS	addu	D	RS	addu \$4, \$4, \$5 jr \$4
	I-E-RS	ori	D	RS	ori \$4, 0xffff jr \$4
	LD-E-RS	lw	D	RS	lw \$4, 0(\$5) jr \$4
	LD-M-RS	lw	D	RS	lw \$4, 0(\$5) nop jr \$4
B 型	R-E-RS	addu	D	RS	addu \$4, \$4, \$5 beq \$4, \$5, loop
	I-E-RS	ori	D	RS	ori \$4, 0xffff beq \$4, \$5, loop
	LD-E-RS	lw	D	RS	lw \$4, 0(\$5) beq \$4, \$5, loop
	LD-M-RS	lw	D	RS	lw \$4, 0(\$5) nop beq \$4, \$5, loop
	R-E-RT	addu	D	RT	addu \$4, \$4, \$5 beq \$5, \$4, loop
	I-E-RT	ori	D	RT	ori \$4, 0xffff beq \$5, \$4, loop
	LD-E-RT	lw	D	RT	lw \$4, 0(\$5) beq \$5, \$4, loop
	LD-M-RT	lw	D	RT	lw \$4, 0(\$5) nop beq \$5, \$4, loop

2. 暂停测试代码

3. 转发

类型	测试类型	前序指令	冲突位置	冲突寄存器	测试序列
----	------	------	------	-------	------

R 型 (以 addu 为例)	R-W-RS	addu	D	RS	addu \$4, \$4, \$5 nop nop addu \$4, \$4, \$5
	I-W-RS	ori	D	RS	ori \$4, \$5, 0xffff nop nop addu \$4, \$4, \$5
	LD-W-RS	lw	D	RS	lw \$4, 0(\$5) nop nop addu \$4, \$4, \$5
	JAL-W-RS	jal	D	RS	jal loop nop nop addu \$1, \$31, \$1
	R-W-RT	addu	D	RT	addu \$4, \$4, \$5 nop nop addu \$4, \$5, \$4
	I-W-RT	ori	D	RT	ori \$4, \$5, 0xffff nop nop addu \$4, \$5, \$4
	LD-W-RT	lw	D	RT	lw \$4, 0(\$5) nop nop addu \$4, \$5, \$4
	JAL-W-RT	jal	D	RT	jal loop nop nop addu \$1, \$1, \$31
	R-W-RS	addu	E	RS	addu \$4, \$4, \$5 nop addu \$4, \$4, \$5 nop
	I-W-RS	ori	E	RS	ori \$4, \$5, 0xffff nop addu \$4, \$4, \$5 nop
	LD-W-RS	lw	E	RS	lw \$4, 0(\$5) nop addu \$4, \$4, \$5 nop

	JAL-W-RS	jal	E	RS	jal loop nop addu \$1, \$31, \$1 nop
	R-M-RS	addu	E	RS	addu \$4, \$4, \$5 addu \$4, \$4, \$5 nop
	I-M-RS	ori	E	RS	ori \$4, \$5, 0xffff addu \$4, \$4, \$5 nop
	JAL-M-RS	jal	E	RS	jal loop addu \$1, \$31, \$1 nop
	R-W-RT	addu	E	RT	addu \$4, \$4, \$5 nop addu \$4, \$5, \$4 nop
	I-W-RT	ori	E	RT	ori \$4, \$5, 0xffff nop addu \$4, \$5, \$4 nop
	LD-W-RT	lw	E	RT	lw \$4, 0(\$5) nop addu \$4, \$5, \$4 nop
	JAL-W-RT	jal	E	RT	jal loop nop addu \$1, \$1, \$31 nop
	R-M-RT	addu	E	RT	addu \$4, \$4, \$5 addu \$4, \$5, \$4 nop
	I-M-RT	ori	E	RT	ori \$4, \$5, 0xffff addu \$4, \$5, \$4 nop
	JAL-M-RT	jal	E	RT	jal loop addu \$1, \$1, \$31 nop
I 型 (以 ori 为 例)	R-W-RS	addu	D	RS	addu \$4, \$4, \$5 nop nop ori \$4, \$4, 0xffff
	I-W-RS	ori	D	RS	ori \$4, \$5, 0xffff nop

					nop ori \$4, \$4, 0x0000
	LD-W-RS	lw	D	RS	lw \$4, 0(\$5) nop nop ori \$4, \$4, 0xffff
	JAL-W-RS	jal	D	RS	jal loop nop nop ori \$1, \$31, 0xffff
	R-W-RS	addu	E	RS	addu \$4, \$4, \$5 nop ori \$4, \$4, 0xffff nop
	I-W-RS	ori	E	RS	ori \$4, \$5, 0xffff nop ori \$4, \$4, 0x0f0f nop
	LD-W-RS	lw	E	RS	lw \$4, 0(\$5) nop ori \$4, \$4, 0xffff nop
	JAL-W-RS	jal	E	RS	jal loop nop ori \$1, \$31, 0xffff nop
	R-M-RS	addu	E	RS	addu \$4, \$4, \$5 ori \$4, \$4, 0xffff nop
	I-M-RS	ori	E	RS	ori \$4, \$5, 0xffff ori \$4, \$4, 0xf0f0 nop
	JAL-M-RS	jal	E	RS	jal loop ori \$1, \$31, 0xffff0 nop
LD 型	R-W-RS	addu	D	RS	addu \$4, \$4, \$5 nop nop lw \$5, 0(\$4)
	I-W-RS	ori	D	RS	ori \$4, \$5, 0xffff nop nop lw \$5, 0(\$4)
	LD-W-RS	lw	D	RS	lw \$4, 0(\$5)

					nop nop lw \$5, 0(\$4)
	JAL-W-RS	jal	D	RS	jal loop nop nop lw \$5, 0(\$31)
	R-W-RS	addu	E	RS	addu \$4, \$4, \$5 nop lw \$5, 0(\$4) nop
	I-W-RS	ori	E	RS	ori \$4, \$5, 0xffff nop lw \$5, 0(\$4) nop
	LD-W-RS	lw	E	RS	lw \$4, 0(\$5) nop lw \$5, 0(\$4) nop
	JAL-W-RS	jal	E	RS	jal loop nop lw \$5, 0(\$31) nop
	R-M-RS	addu	E	RS	addu \$4, \$4, \$5 lw \$5, 0(\$4) nop
	I-M-RS	ori	E	RS	ori \$4, \$5, 0xffff lw \$5, 0(\$4) nop
	JAL-M-RS	jal	E	RS	jal loop lw \$5, 0(\$31) nop
ST 型	R-W-RS	addu	D	RS	addu \$4, \$4, \$5 nop nop sw \$5, 0(\$4)
	I-W-RS	ori	D	RS	ori \$4, \$5, 0xffff nop nop sw \$5, 0(\$4)
	LD-W-RS	lw	D	RS	lw \$4, 0(\$5) nop nop sw \$5, 0(\$4)

	JAL-W-RS	jal	D	RS	jal loop nop nop sw \$5, 0(\$31)
	R-W-RT	addu	D	RT	addu \$4, \$4, \$5 nop nop sw \$4, 0(\$5)
	I-W-RT	ori	D	RT	ori \$4, \$5, 0xffff nop nop sw \$4, 0(\$5)
	LD-W-RT	lw	D	RT	lw \$4, 0(\$5) nop nop sw \$4, 0(\$6)
	JAL-W-RT	jal	D	RT	jal loop nop nop sw \$31, 0(\$5)
	R-M-RS	addu	E	RS	addu \$4, \$4, \$5 sw \$5, 0(\$4) nop
	I-M-RS	ori	E	RS	ori \$4, \$4, \$5 sw \$5, 0(\$4) nop
	JAL-M-RS	jal	E	RS	jal loop sw \$5, 0(\$31) nop
	R-W-RS	addu	E	RS	addu \$4, \$4, \$5 nop sw \$5, 0(\$4) nop
	I-W-RS	ori	E	RS	ori \$4, \$5, 0xffff nop sw \$5, 0(\$4) nop
	LD-W-RS	lw	E	RS	lw \$4, 0(\$5) nop sw \$5, 0(\$4) nop
	JAL-W-RS	jal	E	RS	jal loop nop sw \$5, 0(\$31)

					nop
	R-W-RT	addu	E	RT	addu \$4, \$4, \$5 nop sw \$4, 0(\$5) nop
	I-W-RT	ori	E	RT	ori \$4, \$5, 0xffff nop sw \$4, 0(\$5) nop
	LD-W-RT	lw	E	RT	lw \$4, 0(\$5) nop sw \$4, 0(\$6) nop
	JAL-W-RT	jal	E	RT	jal loop nop sw \$31, 0(\$5) nop
	R-W-RT	addu	M	RT	addu \$4, \$4, \$5 sw \$4, 0(\$5) nop nop
	I-W-RT	ori	M	RT	ori \$4, \$5, 0xffff sw \$4, 0(\$5) nop nop
	LD-W-RT	lw	M	RT	lw \$4, 0(\$5) sw \$4, 0(\$6) nop nop
	JAL-W-RT	jal	M	RT	jal loop sw \$31, 0(\$5) nop nop
JR	R-W-RS	addu	D	RS	addu \$4, \$4, \$5 nop nop jr \$4
	I-W-RS	ori	D	RS	ori \$4, \$5, 0xffff nop nop jr \$4
	LD-W-RS	lw	D	RS	lw \$4, 0(\$5) nop nop

					jr \$4
	JAL-W-RS	jal	D	RS	jal loop nop nop jr \$31
	R-M-RS	addu	D	RS	addu \$4, \$4, \$5 nop jr \$4
	I-M-RS	ori	D	RS	ori \$4, \$5, 0xffff nop jr \$4
	JAL-M-RS	jal	D	RS	jal loop nop jr \$31
	JAL-E-RS	jal	D	RS	jal loop jr \$31
B 型	R-W-RS	addu	D	RS	addu \$4, \$4, \$5 nop nop beq \$4, \$3, loop
	I-W-RS	ori	D	RS	ori \$4, \$5, 0xffff nop nop beq \$4, \$3, loop
	LD-W-RS	lw	D	RS	lw \$4, 0(\$5) nop nop beq \$4, \$3, loop
	JAL-W-RS	jal	D	RS	jal loop nop nop beq \$31, \$3, loop
	R-M-RS	addu	D	RS	addu \$4, \$4, \$5 nop beq \$4, \$3, loop
	I-M-RS	ori	D	RS	ori \$4, \$5, 0xffff nop beq \$4, \$3, loop
	JAL-M-RS	jal	D	RS	jal loop nop beq \$31, \$3, loop
	JAL-E-RS	jal	D	RS	jal loop beq \$31, \$3, loop
	R-W-RT	addu	D	RT	addu \$4, \$4, \$5

					nop nop beq \$3, \$4, loop
	I-W-RT	ori	D	RT	ori \$4, \$5, 0xffff nop nop beq \$3, \$4, loop
	LD-W-RT	lw	D	RT	lw \$4, 0(\$5) nop nop beq \$3, \$4, loop
	JAL-W-RT	jal	D	RT	jal loop nop nop beq \$3, \$31, loop
	R-M-RT	addu	D	RT	addu \$4, \$4, \$5 nop beq \$3, \$4, loop
	I-M-RT	ori	D	RT	ori \$4, \$5, 0xffff nop beq \$3, \$4, loop
	JAL-M-RT	jal	D	RT	jal loop nop beq \$3, \$31, loop
	JAL-E-RT	jal	D	RT	jal loop beq \$3, \$31, loop

4. 指令集全覆盖测试代码

4.1 测试代码

```

ori $1, $0, 0x1234
ori $2, $1, 0x1
ori $3, $2, 0x0
addu $4, $4, $1
subu $5, $4, $4
addu $6, $5, $4
subu $7, $6, $1
lui $7, 0x1
lui $6, 0x2
ori $10, $0, 0x24

```

```

sw $6, 0($0)
sw $7, 0($10)
lw $7, 0($10)
lw $6, 0($0)
j jump
addu $10, $6, $6
subu $10, $10, $10
jump:
ori $15, $0, 0
beq $15, $0, out
addu $20, $15, $6
nop
lui $20, 0x20
nop
nop
out:
jal target
sw $31, 0($0)
nop
ori $15, 0x307c
jr $15
nop
target:
jr $31
addu $31, $31, $1
nop
nop
nop
subu $31, $31, $31

```

4.2 测试结果

```
45@00003000: $ 1 <= 00001234
55@00003004: $ 2 <= 00001235
65@00003008: $ 3 <= 00001235
75@0000300c: $ 4 <= 00001234
85@00003010: $ 5 <= 00000000
95@00003014: $ 6 <= 00001234
105@00003018: $ 7 <= 00000000
115@0000301c: $ 7 <= 00010000
125@00003020: $ 6 <= 00020000
135@00003024: $10 <= 00000024
135@00003028: *00000000 <= 00020000
145@0000302c: *00000024 <= 00010000
165@00003030: $ 7 <= 00010000
175@00003034: $ 6 <= 00020000
195@0000303c: $10 <= 00040000
205@00003044: $15 <= 00000000
235@0000304c: $20 <= 00020000
245@00003060: $31 <= 00003068
245@00003064: *00000000 <= 00003068
275@0000307c: $31 <= 0000429c
295@0000306c: $15 <= 0000307c
335@0000307c: $31 <= 000054d0
375@0000308c: $31 <= 00000000
```

5. 覆盖转发暂停指令测试块

内容如上表，进行覆盖全指令测试后，基本正确。

七、思考题

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。

遇到了数据冲突，使用暂停和转发解决，详细列表如第六大点。