

MIPS 流水线处理器设计文档

MIPS 单周期处理器设计文档由顶层模块、数据通路、控制器、冲突模块、测试程序组成。

一、顶层模块 mips.v

序号	信号	位数	方向	功能描述
1	clk	1	I	pc、grf、dm 的时钟信号
2	reset	1	I	当 reset=1 且时钟为上跳沿时，pc 被设置为 0x00003000，gpr 所有寄存器置零，dm 所有数据置零

顶层模块将数据通路模块 datapath.v 与控制器模块 control.v,hazard.v 相连。

二、数据通路 datapath.v

信号端口

序号	信号	位数	方向	描述
1	reset	1	I	复位信号
2	clk	1	I	时钟信号
3	PCOP	2	I	muxPCOP 模块控制信号
4	enable	1	I	PC,IF_ID 使能端
5	clear	1	I	ID_EX 清零信号
6	RegDst	2	I	muxRegDst 模块控制信号
7	ExtOP	1	I	Ext 模块控制信号
8	RegWrite	1	I	寄存器文件写使能
9	RegWData	1	I	muxRegWData 模块控制信号
10	ALUSrc	1	I	muxALUSrc 模块控制信号
11	ALUOP	4	I	选择 ALU 运算类型的信号
12	MemWrite	1	I	dm 写入控制信号
13	Instr_original	32	O	IM 输出指令
14	Instr_D	32	O	D 级指令
15	Instr_E	32	O	E 级指令
16	Instr_W	32	O	W 级指令
17	MF_RS_D_OP	2	O	MF_RS_D 模块控制信号

18	MF_RT_D_OP	2	O	MF_RT_D 模块控制信号
19	MF_RS_E_OP	2	O	MF_RS_E 模块控制信号
20	MF_RT_E_OP	2	O	MF_RT_E 模块控制信号
21	MF_RT_M_OP	2	O	MF_RT_M 模块控制信号
22	Equal_out	1	O	equal 模块输出信号

datapath.v 将 pc.v , dm.v , im.v , add4.v,npc.v,alu.v , grf.v , ext.v , muxPCOP.v , muxRegWData.v , muxRegDst.v , muxALUSrc.v 等模块连接在一起。下面将介绍这些模块。

1、grf.v

信号端口

序号	信号	位数	方向	描述
1	RA1	5	I	读寄存器文件时第一个寄存器的下标
2	RA2	5	I	读寄存器文件时第二个寄存器的下标
3	Waddr	5	I	写入文件的寄存器下标
4	WData	32	I	寄存器文件写入数据
5	clk	1	I	时钟信号
6	reset	1	I	复位信号
7	RegWrite	1	I	寄存器文件写使能
8	RData1	32	O	读寄存器文件时第一个寄存器的输出
9	RData2	32	O	读寄存器文件时第二个寄存器的输出

模块功能

序号	功能名称	功能描述
1	读寄存器	RData1 输出 RA1 所寻址的寄存器的文件数据，RData2 输出 RA2 所寻址的寄存器的文件数据
2	写寄存器	当时钟上升沿到来时，并且 reset=0 且 RegWrite 有效时，WD 被写入 WA 所寻址的寄存器
3	复位	reset=1 且 clk 为上升沿时 gpr 所有寄存器置零

2、alu.v

信号端口

序号	信号	位数	方向	描述
1	ALU_A	32	I	ALU 的第一个操作数
2	ALU_B	32	I	ALU 的第二个操作数
3	ALUOP	4	I	选择 ALU 运算类型的信号
4	ALU_C	32	O	ALU 的运算结果
5	equal	1	O	判断 ALU_A 与 ALU_B 是否相等，若相等输出 1，否则输出 0
6	s	5	I	逻辑左移、右移位数

模块功能

序号	功能名称	功能描述
1	加法	输出端 ALU_C=ALU_A+ALU_B
2	减法	输出端 ALU_C=ALU_A-ALU_B
3	或	输出端 ALU_C=ALU_A ALU_B
4	判断相等	若 ALU_A=ALU_B, equal 信号输出 1，否则输出 0
5	置高位	输出端 ALU_C={{ALU_B[15:0]},{16{1'b0}}}
6	逻辑左移	输出端 ALU_C=ALU_A<<s
7	逻辑右移	输出端 ALU_C=ALU_A>>s
8	与	输出端 ALU_C=ALU_A&ALU_B
9	异或	输出端 ALU_C=ALU_A^ALU_B

3、dm.v

信号端口

序号	信号	位数	方向	描述
1	addr	32	I	写入数据储存器的数据的地址/读取数据储存器的数据的地址
2	din	32	I	写入数据储存器的数据
3	MemWrite	1	I	写入控制信号
4	clk	1	I	时钟信号
5	reset	1	I	复位信号
6	dout	32	O	从数据存储器读取的数据

模块功能

序号	功能名称	功能描述
1	读数据存储器	dout 输出数据存储器在 addr[11:2]地址的数据
2	写数据存储器	当 MemWrite 有效,reset=0,并且时钟为上升沿时, din 被写入数据

	寄存器	存储器地址为 addr[11:2]的区域
3	复位	reset=1 并且时钟为上升沿时,DM 所有数据置零

4、ext.v

信号端口

序号	信号	位数	方向	描述
1	ext_in	16	I	位扩展输入信号
2	EXTOP	1	I	信号为 1 时，进行符号扩展，信号为 0 时，进行 0 扩展
3	ext_out	32	O	位扩展输出信号

模块功能

序号	功能名称	功能描述
1	符号扩展	当 EXTOP 信号为 1 时，ext_in 符号扩展后由 ext_out 输出
2	0 扩展	当 EXTOP 信号为 0 时，在 ext_in 前置 16 个 0 后由 ext_out 输出

5、im.v

信号端口

序号	信号	位数	方向	描述
1	address	32	I	指令地址
2	instr	32	O	指令

模块功能

序号	功能名称	功能描述
1	取指令	取出位于(address-0x00003000)[11:2]处指令

6、pc.v

信号端口

序号	信号	位数	方向	描述
1	PC_in	32	I	下一条指令 PC 值
2	clk	1	I	时钟信号
3	reset	1	I	复位信号
4	PC_out	32	O	当前 PC 值
5	Enable	1	I	使能端

模块功能

序号	功能名称	功能描述
----	------	------

1	输出指令地址	时钟上升沿且 reset=0, enable=1, PC 输出指令地址
2	复位	时钟上升沿且 reset=1, 将 PC 置为 0x00003000

7、muxPCOP.v

信号端口

序号	信号	位数	方向	描述
1	PCOP	2	I	muxPCOP 控制信号
2	PC4	32	I	PC+4
3	NPC	32	I	NPC 结果
4	newPC	32	O	下条指令地址
5	RD1	32	I	rs 值

模块功能

序号	功能名称	功能描述
1	输出下条指令地址	PCOP==2'b00:newPC<=PC4; PCOP==2'b01:newPC<=NPC; PCOP==2'b10:newPC<=RD1;

8、muxRegDst.v

信号端口

序号	信号	位数	方向	描述
1	RegDst	2	I	muxRegDst 控制信号
2	IR_W	32	I	W 级指令
3	WAddr	5	O	写入文件的寄存器下标

模块功能

序号	功能名称	功能描述
1	输出写入文件的寄存器下标	RegDst=2'b00, WAddr=rd RegDst=2'b01, WAddr=rt RegDst=2'b10, WAddr=5'd31

9、muxRegWData.v

信号端口

序号	信号	位数	方向	描述
1	ALUC_W	32	I	W 级 ALU 结果
2	DM_W	32	I	W 级从数据存储器读取的数据
3	RegWData	2	I	muxRegWData 的控制信号
4	WData	32	O	寄存器文件写入数据

5	IR_W	32	I	W 级指令
6	PC4_W	32	I	PC+4

模块功能

序号	功能名称	功能描述
1	输出寄存器文件写入数据	RegWData=2'b00:WData<=ALUC_W; RegWData=2'b01:WData<=DM_W; RegWData=2'b10:WData<=PC4_W+4;

10、muxALUSrc.v

信号端口

序号	信号	位数	方向	描述
1	ALUSrc	1	I	muxAIUSrc 的控制信号
2	RD2	32	I	读寄存器文件时第二个寄存器的下标
3	EXT_out	32	I	位扩展输出信号
4	ALU_B	32	O	ALU 的第二个操作数

模块功能

序号	功能名称	功能描述
1	选择 ALU 的第二个操作数	ALUSrc=0 时: ALU_B=RD2 ALUSrc=1 时: ALU_B=EXT_out

11、add4.v

信号端口

序号	信号	位数	方向	描述
1	PC	32	I	PC
3	PC4	32	O	PC+4

模块功能

序号	功能名称	功能描述
1	PC+4	PC4=PC+4

12、IF_ID.v

信号端口

序号	信号	位数	方向	描述
1	clk	1	I	时钟信号
2	reset	1	I	复位信号
3	IR_IM	32	I	IM 输出指令
4	PC4_ADD 4	32	I	PC+4
5	PC4_D	32	O	PC+4
6	IR_D	32	O	D 级指令
7	enable	1	I	使能端

模块功能

序号	功能名称	功能描述
1	D 级流水线寄存器	传递 IR 与数据
2	使能端置零	暂停时 enable=0

13、ID_EX.v

信号端口

序号	信号	位数	方向	描述
1	clk	1	I	时钟信号
2	reset	1	I	复位信号
3	IR_D	32	I	D 级指令
4	RD1	32	I	读寄存器文件时第一个寄存器的输出
5	RD2	32	O	读寄存器文件时二个寄存器的输出
6	EXT_out	32	O	D 级指令
7	PC4_D	32	O	PC+4
8	IR_E	32	O	E 级指令
9	RD1_E	32	O	读寄存器文件时第一个寄存器的输出
10	RD2_E	32	O	读寄存器文件时第二个寄存器的输出
11	EXT_E	32	O	EXT 结果
12	PC4_E	32	O	PC+4
13	Clear	1	I	清零信号

模块功能

序号	功能名称	功能描述
1	E 级流水线寄存器	传递 IR 与数据
2	清零	ID_EX 清零

14、EX_MEM. v

信号端口

序号	信号	位数	方向	描述
1	clk	1	I	时钟信号
2	reset	1	I	复位信号
3	IR_E	32	I	E 级指令
4	RD2_E	32	I	读寄存器文件时二个寄存器的输出
5	ALU_out	32	I	ALU 结果
6	PC4_E	32	I	PC+4
7	IR_M	32	O	M 级指令
8	ALUC_M	32	O	ALU 结果
9	PC4_M	32	O	PC+4
10	RD2_M	32	O	读寄存器文件时二个寄存器的输出

模块功能

序号	功能名称	功能描述
1	M 级流水线寄存器	传递 IR 与数据

15、MEM_WB. v

信号端口

序号	信号	位数	方向	描述
1	clk	1	I	时钟信号
2	reset	1	I	复位信号
3	IR_M	32	I	M 级指令
4	DM_Data	32	I	从数据存储器读取的数据
5	ALU_M	32	I	ALU 结果
6	PC4_M	32	I	PC+4
7	IR_W	32	O	W 级指令
8	ALUC_W	32	O	ALU 结果
9	PC4_W	32	O	PC+4
10	DM_W	32	O	从数据存储器读取的数据

模块功能

序号	功能名称	功能描述
1	W 级流水线寄存器	传递 IR 与数据

16、MF_RS_E. v

信号端口

序号	信号	位数	方向	描述
1	MF_RS_E_OP	2	I	MF_RS_E 控制信号
2	RD1_E	32	I	读寄存器文件时第二个寄存器的值
3	ALUC_M	32	I	ALU 结果
4	WData	32	I	muxRegWData 结果
5	MF_RS_E_out	32	O	转发结果
6	PC4_M	32	I	PC+4

模块功能

序号	功能名称	功能描述
1	转发	控制信号为 01,10,11 时转发

17、MF_RT_E.v

信号端口

序号	信号	位数	方向	描述
1	MF_RT_E_OP	2	I	MF_RT_E 控制信号
2	RD2_E	32	I	读寄存器文件时第二个寄存器的值
3	ALUC_M	32	I	ALU 结果
4	WData	32	I	muxRegWData 结果
5	RD2	32	O	转发结果
6	PC4+M	32	I	PC+4

模块功能

序号	功能名称	功能描述
1	转发	控制信号为 01,10,11 时转发

18、MF_RS_D.v

信号端口

序号	信号	位数	方向	描述
1	MF_RS_D_OP	2	I	MF_RS_D 控制信号
2	RData1	32	I	读寄存器文件时第 1 个寄存器的值

3	ALUC_M	32	I	ALU 结果
4	WData	32	O	muxRegWData 结果
5	MF_RS_D_out	32	I	转发结果
6	PC4_M	32	I	PC+4

模块功能

序号	功能名称	功能描述
1	转发	控制信号为 01,10,11 时转发

19、MF_RT_D.v

信号端口

序号	信号	位数	方向	描述
1	MF_RT_D_OP	2	I	MF_RS_D 控制信号
2	RData2	32	I	读寄存器文件时第二个寄存器的值
3	ALUC_M	32	I	ALU 结果
4	WData	32	O	muxRegWData 结果
5	MF_RT_D_out	32	I	转发结果
6	PC4_M	32	I	PC+4

模块功能

序号	功能名称	功能描述
1	转发	控制信号为 01,10,11 时转发

20、MF_RT_M.v

信号端口

序号	信号	位数	方向	描述
1	MF_RT_M_OP	2	I	MF_RT_M 控制信号
2	RD2_M	32	I	读寄存器文件时第二个寄存器的值
3	WData	32	O	muxRegWData 结果
4	MF_RT_M_out	32	I	转发结果

模块功能

序号	功能名称	功能描述
1	转发	控制信号为 01 时转发

21、compare.v

21、信号端口

序号	信号	位数	方向	描述
1	RD1	32	I	读寄存器文件时第一个寄存器的值
2	RD2	32	I	读寄存器文件时第二个寄存器的值
3	Equal_out	1	O	RD1=RD2, 则为 1, 否则为 0

三、控制器 control.v

以下为控制信号产生的真值表：

func	100001	100011	无						001000	无
opcode	000000	000000	001101	100011	101011	000100	001111	000011	000000	
控制信号	addu	subu	ori	lw	sw	beq	lui	jal	jr	nop
PCOP	00	00	00	00	00	01	00	01	10	00
EXTOP	x	x	0	1	1	1	x	x	x	x
RegDst[1:0]	00	00	01	01	x	x	01	10	x	x
RegWrite	1	1	1	1	0	0	1	1	0	0
RegWData[1:0]	00	00	00	01	x	x	10	11	x	x
ALUSrc	0	0	1	1	1	x	x	x	x	x
ALUOP[3:0]	0000	0001	0010	0000	0000	x	0011	x	x	x
MemWrite	0	0	0	0	1	0	0	0	0	0

func	100000	100010	100101	000000	000010					100100	100101	100110
opcode	000000	000000	000000	000000	000000	000101	000010	001000	001001	000000	000001	000000
控制信号	add	sub	or	sll	srl	bne	j	addi	addiu	and	or	xor
PCOP	00	00	00	00	00	01	01	00	00	00	00	00
EXTOP	x	x	x	x	x	1	x	1	1	x	x	x

RegDst[1:0]	00	00	00	00	00	x	x	01	01	00	00	00
RegWrite	1	1	1	1	1	0	0	1	1	1	1	1
RegWData[1:0]	00	00	00	00	00	x	x	00	00	00	00	00
ALUSrc	00	00	00	00	00	x	x	01	01	00	00	00
ALUOP[3:0]	0000	0001	0010	0100	0101	x	x	0000	0000	0110	0010	0111
MemWrite	0	0	0	0	0	0	0	0	0	0	0	0

control.v 信号端口

序号	信号	位数	方向	描述
1	instr	32	I	指令
3	PCOP	2	O	muxPCOP 控制信号
4	RegDst	2	O	muxRegDst 控制信号
5	ExtOP	1	O	ext 控制信号
6	RegWrite	1	O	gpr 写入控制信号
7	RegWData	2	O	muxRegWData 的控制信号
8	ALUSrc	1	O	muxAIUSrc 的控制信号
9	ALUOP	4	O	alu 控制信号
10	MemWrite	1	O	dm 写入控制信号

四、冲突模块

信号端口

序号	信号	位数	方向	描述
1	IR_D	32	I	D 级指令
2	IR_E	32	I	E 级指令
3	IR_M	32	I	M 级指令
4	IR_W	32	I	W 级指令
5	MF_RT_E_OP	2	O	MF_RT_E 控制信号
6	MF_RS_E_OP	2	O	MF_RS_E 控制信号
7	MF_RS_D_OP	2	O	MF_RS_D 控制信号
8	MF_RT_D_OP	2	O	MF_RT_D 控制信号
9	MF_RT_M_OP	2	O	MF_RT_M 控制信号
10	enable	1	O	PC,IF_ID 使能信号

11	clear	1	O	ID_EX 清零信号
----	-------	---	---	------------

五、测试代码

```

lui $1,0x1010
lui $2,0x1010
beq $1,$2,label
lui $3,0x1111
label1:lw $9,4($0)
jr $9
lui $4,0x1111
lui $5,0x1111
label:
lui $6,0x1111
jal label1
sw $31,4($0)
lui $10,0x1111
lui $0,0x1111
ori $1,$0,0x0101
sw $1,0($0)
lw $1,0($0)
addi $2,$1,4
sw $2,4($0)
sub $3,$2,$1
sw $3,8($0)
lw $2,4($0)
or $4,$3,$2
sw $4,12($0)
lw $3,4($0)
lw $4,12($0)
xor $5,$4,$3
sw $5,16($0)
lw $6,16($0)
ori $6,$5,0x0010
lw $5,4($0)
and $7,$5,$6
subu $8,$6,$7
addiu $9,$8,0x2030
add $10,$8,$9
ori $11,$10,0x3241
sll $12,$10,9
srl $13,$12,6
xor $14,$12,$13
sw $14,0($0)
lw $15,0($0)
addi $16,$15,0x4564
bne $16,$15,end
sub $16,$16,$15
add $17,$16,$15
end:
ori $16,$16,0x1100

```

```

$ 1 <= 10100000
$ 2 <= 10100000
$ 3 <= 11110000
$ 6 <= 11110000
$31 <= 0000302c
*00000004 <= 0000302c
$ 9 <= 0000302c
$ 4 <= 11110000
$10 <= 11110000
$ 1 <= 00000101
*00000000 <= 00000101
$ 1 <= 00000101
$ 2 <= 00000105
*00000004 <= 00000105
$ 3 <= 00000004
*00000008 <= 00000004
$ 2 <= 00000105
$ 4 <= 00000105
*0000000c <= 00000105
$ 3 <= 00000105
$ 4 <= 00000105
$ 5 <= 00000000
*00000010 <= 00000000
$ 6 <= 00000000
$ 6 <= 00000010
$ 5 <= 00000105
$ 7 <= 00000000
$ 8 <= 00000010
$ 9 <= 00002040
$10 <= 00002050
$11 <= 00003251
$12 <= 0040a000
$13 <= 00010280
$14 <= 0041a280
*00000000 <= 0041a280
$15 <= 0041a280
$16 <= 0041e7e4
$16 <= 00004564
$16 <= 00005564

```

六、思考题

令

CAL_R={add,addu,sub,subu,sll,srl,and,or,xor},CAL_I={lui,ori,addi,addiu},B={beq,bne};

转发

1、D 级为 B(rs,rt), M 级为 CAL_R(rd).

```
lui $1,0x1111
```

```
lui $0,0x1111
```

```
xor $2,$1,$0
```

```
addi $2,$2,5
```

```
bne $0,$2,label1
```

```
sll $2,$2,2
```

```
lui $4,0x1111
```

```
label1:srl $2,$2,3
```

2、D 级为 jr(rs), M 级为 CAL_R(rd).

```
ori $2,$0,0x1111
```

```
ori $4,$0,0x1010
```

```
sll $0,$2,4
```

```
and $3,$2,$4
```

```
addi $1,$0,0x3020
```

```
jr $1
```

```
or $1,$1,$4
```

```
lui $1,0x1111
```

```
beq $1,$1,label
```

```
nop
```

```
lui $2,0x1111
```

```
label:
```

```
srl $5,$1,4
```

3、D 级为 B(rs,rt), W 级为 CAL_R(rd),CAL_I(rt),jal(31),lw(rt)

(CAL_I)

```
lui $1,0x1010
lui $2,0x0101
lui $3,0x1111
beq $1,$1,label
lui $4,0x1111
lui $5,0x1111
label:addi $1,$1,4
sub $1,$1,$2
```

(CAL_R)

```
lui $1,0x1010
addi $1,$1,8
xor $2,$2,$1
lui $3,0x1111
beq $1,$2,label
lui $4,0x1111
lui $5,0x1111
label:addi $1,$1,4
sub $1,$1,$2
```

(jal)

```
lui $1,0x1010
addi $1,$1,8
xor $2,$2,$1
lui $3,0x1111
jal label
lui $4,0x1111
lui $5,0x1111
label:addi $1,$1,4
sub $1,$31,$0
beq $31,$1 label1
add $31,$2,$1
```

```
and $31,$31,$3
label1:sll $4,$3,2
srl $5,$4,8
```

(Lw)

```
lui $1,0x1010
addi $1,$1,8
xor $2,$2,$1
lui $3,0x1111
sw $2,4($0)
lw $1,4($0)
sll $3,$2,8
srl $4,$3,8
bne $4,$1,label
sub $4,$4,$1
xor $3,$2,$3
label:
addi $4,$4,6
addiu $4,$4,99
```

4、D 级为 jr(rs), W 级为 CAL_R(rd),CAL_I(rt),jal(31),lw(rt)

(CAL_I)

```
ori $2,0x3014
addi $1,$2,8
subu $3,$2,$0
sll $4,$3,5
jr $1
srl $5,$4,8
or $6,$5,$4
xor $7,$5,$6
sub $8,$7,$6
```

(CAL_R)

ori \$2,0x3014

ori \$4,\$0,8

add \$1,\$2,\$4

subu \$3,\$2,\$0

sll \$4,\$3,5

jr \$1

srl \$5,\$4,8

or \$6,\$5,\$4

xor \$7,\$5,\$6

sub \$8,\$7,\$6

(jal)

ori \$2,0x3014

ori \$4,\$0,8

add \$1,\$2,\$4

subu \$3,\$2,\$0

sll \$4,\$3,5

jal label

srl \$5,\$4,8

j end

lui \$8,0x0101

label:and \$6,\$5,\$4

jr \$31

or \$6,\$5,\$4

xor \$7,\$5,\$6

sub \$8,\$7,\$6

end:

or \$7,\$6,\$31

(lw)

ori \$2,0x3020

ori \$4,\$0,8

```

add $1,$2,$4
sw $1,4($0)
lw $2,4($0)
srl $3,$1,5
sll $4,$3,5
jr $2
srl $5,$4,8
or $6,$5,$4
xor $7,$5,$6
sub $8,$7,$6

```

5、E 级为 CAL_R,CAL_I,lw,sw(rs), M 级为 CAL_R(rd),CAL_I(rt),jal(31)

```

lui $1,0x1010
ori $1,$1,0x1100
addi $1,$1,3
addu $1,$1,$1
or $0,$1,$1
subu $1,$1,$0
srl $2,$1,6
sub $1,$1,$2
add $1,$1,$1
sll $1,$1,4
srl $1,$1,6
and $1,$1,$1
or $1,$1,$1
xor $1,$1,$1
ori $1,$1,0x1100
and $1,$2,$1

```

6、E 级为 CAL_R,CAL_I,lw,sw(rs),W 级为 CAL_R(rd),CAL_I(rt),jal(31),lw(rt)

```

lui $0,0x1111
ori $1,$0,0x0101

```

```

addi $2,$1,4
sub $3,$2,$1
or $4,$3,$2
xor $5,$4,$3
ori $6,$5,0x0010
and $7,$5,$6
subu $8,$6,$7
addiu $9,$7,0x2030
add $10,$8,$9
ori $11,$10,0x3241
sll $12,$10,9
srl $13,$12,6
xor $14,$12,$13
sw $14,0($0)
lw $15,0($0)
addi $16,$15,0x4564

```

7、E 级为 CAL_R,sw(rt),M 级为 CAL_R(rd),CAL_I(rt),jal(31)

```

lui $0,0x1111
ori $1,$0,0x0101
sw $1,0($0)
lw $1,0($0)
addi $2,$1,4
sw $2,4($0)
sub $3,$2,$1
sw $3,8($0)
lw $2,4($0)
or $4,$3,$2
sw $4,12($0)
lw $3,4($0)
lw $4,12($0)

```

```

xor $5,$4,$3
sw $5,16($0)
lw $6,16($0)
ori $6,$5,0x0010
lw $5,4($0)
and $7,$5,$6
subu $8,$6,$7
addiu $9,$8,0x2030

```

8、E 级为 CAL_R,sw(rt),W 级为 CAL_R(rd),CAL_I(rt),jal(31),lw(rt)

```

lui $0,0x1111
ori $1,$0,0x0101
sw $1,0($0)
lw $1,0($0)
addi $2,$1,4
sw $2,4($0)
sub $3,$2,$1
sw $3,8($0)
lw $2,4($0)
or $4,$3,$2
sw $4,12($0)
lw $3,4($0)
lw $4,12($0)
xor $5,$4,$3
sw $5,16($0)
lw $6,16($0)
ori $6,$5,0x0010
lw $5,4($0)
and $7,$5,$6
subu $8,$6,$7
addiu $9,$8,0x2030

```

9、M 级为 sw(rt),W 级为 CAL_R(rd),CAL_I(rt),jal(31),lw(rt)

```
beq $1,$2,label1
```

```
lui $1,0x1010
```

```
label:
```

```
ori $7,0x0004
```

```
addu $31,$31,$7
```

```
lui $9,0x1111
```

```
jr $31
```

```
lui $8,0x1111
```

```
label1:
```

```
jal label
```

```
sw $31,4($0)
```

```
lui $4,0x1111
```

```
lui $5,0x1111
```

```
lui $6,0x1111
```

```
lui $1,0x1010
```

```
addu $2,$1,$2
```

```
sw $2,4($0)
```

```
lui $1,0x1010
```

```
ori $2,$1,0x00001
```

```
sw $2,4($0)
```

```
lui $1,0x1010
```

```
ori $2,$1,0x00001
```

```
sw $2,4($0)
```

```
lw $2,4($0)
```

```
sw $2,4($0)
```

暂停

1、E 级为 CAL_I,D 级为 beq,E 级 rt=D 级 rs 或 rt

```
lui $1,0x1010
```

```

lui $2,0x1010
beq $1,$2,label
lui $3,0x1111
lui $4,0x1111
lui $5,0x1111
label: lui $6,0x1111

```

2、E 级为 CAL_R,D 级为 beq,E 级 rd=D 级 rs 或 rt

```

lui $2,0x1010
addu $1,$2,$1
beq $1,$2,label
lui $3,0x1111
lui $4,0x1111
lui $5,0x1111
label: lui $6,0x1111

```

3、E 级为 lw,D 级为 beq,E 级 rt=D 级 rs 或 rt

```

lui $1,0x1010
sw $1,4($0)
lw $2,4($0)
lw $1,4($0)
beq $1,$2,label
lui $3,0x1111
lui $4,0x1111
lui $5,0x1111
label: lui $6,0x1111

```

4、E 级为 CAL_R,D 级为 jr,E 级 rd=D 级 rs

```

lui $1,0x1010
lui $2,0x1010
beq $1,$2,label
lui $3,0x1111
label1: addu $9,$9,$31

```

```

jr $9
lui $4,0x1111
lui $5,0x1111
label:lui $6,0x1111
jal label1
lui $7,0x1111
nop

```

5、E 级为 CAL_I,D 级为 jr,E 级 rt=D 级 rs

```

lui $1,0x1010
lui $2,0x1010
beq $1,$2,label
lui $3,0x1111
label1:
ori $9,$31,0x0000
jr $9
lui $4,0x1111
lui $5,0x1111
label:
lui $6,0x1111
jal label1
lui $7,0x1111
nop

```

6、E 级为 CAL_R,D 级为 lw,E 级 rt=D 级 rs

```

lui $1,0x1010
lui $2,0x1010
beq $1,$2,label
lui $3,0x1111
label1:
lw $9,4($0)
jr $9

```

```
lui $4,0x1111
```

```
lui $5,0x1111
```

```
label:
```

```
lui $6,0x1111
```

```
jal label1
```

```
sw $31,4($0)
```

```
nop
```

```
lui $10,0x1111
```

7、E 级为 lw,D 级为 CAL_R,E 级 rt=D 级 rs 或 rt

```
lui $1,0x1010
```

```
sw $1,4($0)
```

```
lw $1,4($0)
```

```
addu $2,$1,$1
```

8、E 级为 lw,D 级为 CAL_I,E 级 rt=D 级 rs

```
lui $1,0x1010
```

```
sw $1,4($0)
```

```
lw $1,4($0)
```

```
ori $2,$1,0x1010
```

9、E 级为 lw,D 级为 sw,E 级 rt=D 级 rs

```
ori $1,$1,0x0004
```

```
sw $1,4($0)
```

```
lw $1,4($0)
```

```
sw $2,8($2)
```

10、E 级为 lw,D 级为 lw,E 级 rt=D 级 rs

```
ori $1,$1,0x0004
```

```
sw $1,4($0)
```

```
lw $2,4($0)
```

```
lw $3,0($2)
```

11、M 级为 lw,D 级为 jr,M 级 rt=D 级 rs

```
beq $1,$2,label
```



```

lui $3,0x1111
label1:
lw $9,4($0)
lui $11,0x1111
jr $9
lui $4,0x1111
lui $5,0x1111
label:
lui $6,0x1111
jal label1
sw $31,4($0)
nop
lui $10,0x1111

```

12、M 级为 lw,D 级为 beq,M 级 rt=D 级 rs 或 rt

```

beq $1,$2,label
lui $3,0x1111
label1:
lw $9,4($0)
lw $11,4($0)
beq $9,$11,label2
lui $4,0x1111
lui $5,0x1111
label:
lui $6,0x1111
jal label1
sw $31,4($0)
label2:
nop
lui $10,0x1111

```