浙江大学

MIPS 汇编语言规范

【PentiumX】

魏世嘉
刘俊灏
陈　俊
孙　同

二零一四年十二月

目录

一、前言

汇编指令采用精简指令方式 RISC。32 位等长指令。不分大小写。MIPS 架构，R、I、J 三种类型。

二、汇编指令

指令集的实现采用多时钟 CPU 方式。扩充 ALU 功能。增加了协处理器。由 MMU 负责存储器读写。
符号说明：
rs、rt、rd：通用寄存器。
imm：16 位补码立即数。
dat：数据
adr：地址
func：功能号
其它。协处理器。

1、算术指令

1．寄存器加法 ADD
格式：ADD　rs，rt，rd
执行：rd = rs + rt
类型：R
机器码：0:6，rs:5，rt:5，rd:5，0:5，32:6
注意：加法结果大于 32 位补码，则产生溢出中断。

2．立即数加法 ADDU
格式：ADDi　rs，rt，imm
执行：rd = rs + imm
类型：R
机器码：0:6，rs:5，rt:5，rd:5，0:5，32:6
注意：16 位立即数 imm 按补码方式扩展。不溢出

3.减法有溢出 S UB
格式：sub rd,rs,rt
执行：rd=rs-rt
类型：R
机器码：0:6，rs:5，rt:5，rd:5，0:5，34:6

4.SUBU 减法不溢出
格式：subu rd,rs,rt
执行：rd=rs-rt
类型：R
机器码：0:6，rs:5，rt:5，rd:5，0:5，35:6

5.立即数加，有溢出 ADDI
格式：addi rt,rs,imm
执行：rt=rs+imm
类型：I
机器码：8:6 ，rs:5， rd:5， imm:16
备注：imm 符号扩展

6 立即数加，不溢出 ADDIU
格式：addiu rt,rs,imm
执行：rt=rs+imm
类型：I
机器码：9:6 ，rs:5， rd:5， imm:16

2、逻辑指令

1.立即数与 ANDI
格式：addi rt,rs,imm
执行：rt=rs&imm
类型：I
机器码：12:6 ，rs:5， rd:5， imm:16

2.与 ADN
格式：and rd,rs,rt
执行：rd=rs&rt
类型：R
机器码：0:6，rs:5，rt:5，rd:5，0:5，32:6

3.或非 NOR
格式：nor rd,rs,rt
执行：rd=~(rs|rt)
类型：R
机器码：0:6，rs:5，rt:5，rd:5，0:5，39:6

4.或 OR
格式：or rd,rs,rt
执行：rd=rs|rt
类型：R
机器码：0:6，rs:5，rt:5，rd:5，0:5，37:6

5.异或 XOR
格式：xor rd,rs,rt
执行：rd=rs^rt
类型：R
机器码：0:6，rs:5，rt:5，rd:5，0:5，38:6

6.立即数或 ORI
格式：ori rt,rs,imm
执行：rt=rs|imm
类型：I
机器码：14:6 ，rs:5， rd:5， imm:16
备注：imm 无符号扩展 ，填 0

7.立即数异或 XORI
格式： xori rt,rs,imm
执行： rd=rs^imm
类型：I
机器码：13:6 ，rs:5， rd:5， imm:16

3、转移指令

1.寄存器转移 JALR
格式：jalr rs,rd
执行：$rd=PC+4, PC=rs
类型：R
机器码：0:6，rs:5 , 0:5 ， rd:5 , 0:5 , 9:6

2.寄存器转移 JR
格式：jr rs
执行：PC=rs
类型：R
机器码：0:6，rs:5 , 0:5 ， 0:5 , 0:5 , 8:6

3.转移 J
格式：j L
执行：PC=PC 高 4|(L<<2)
类型：J
机器码：2:6，L:30

4.调子程序 JAL
格式：jal L
执行：$ra=PC+4, j L
类型：J
机器码：3:6，L:30

4、移位指令

1.逻辑左移 SLL
格式：sll rd,rt,sft
执行：rd=rt<<sft< td="">
类型：R
机器码：0:6，0:5，rt:5，rd:5，sft:5，0:6

2.算术右移 SRA
格式：sra rd,rt,sft
执行：rd=rt>>>sft
类型：R
机器码：0:6，0:5，rt:5，rd:5，sft:5，3:6

3.逻辑右移 SRL
格式：srl rd,rt,sft
执行：rd=rt>>sft
类型：R
机器码：0:6，0:5，rt:5，rd:5，sft:5，2:6

5、比较指令

1.比较 SLT
格式：slt rd,rs,rt
执行：rd=0;if(rs< td="">
类型：R
机器码：0:6，rs:5，rt:5，rd:5，0:5，42:6
注意：有符号比较

2.小于无符号数 SLTU
格式：sltu rd,rs,rt
执行：rd=0;if(rs< td="">
类型： R
机器码：0:6，rs:5，rt:5，rd:5，0:5，43:6
注意：无符号数比较

3.小于立即数 SLTI
格式：slti rt,rs,imm
执行：rd=0;if(rs< td="">
类型：I
机器码：10:6 ，rs:5， rd:5， imm:16

4.小于无符号数 SLTIU
格式：sltiu rt,rs,imm
执行：rd=0;if(rs< td=""><>
类型：I
机器码：11:6 ， rs:5， rd:5， imm:16

6.系统指令

1.系统调用 SYSCALL
格式：syscall
执行：$v0 功能号，$a 参数
类型：R
机器码：0:6，0:5，0:5，0:5，0:5，12:6

2. 中断返回 eret
格式：eret
执行：EXL=0;PC=EPC
类型：R
机器码：10:6，16:5，0:5，0:5，0:5，24:6

3.读协 0 寄存器 MFC0
格式：mfc0 rt,rd
执行：rt=协 0.rd
类型：R
机器码：10:6，0:5，rt:5，rd:5，0:5，0:6

4.写协 0 寄存器 MTC0
格式：mtc0 rt,rd
执行：协 0.rd=rt
类型：R
机器码：10:6，4:5，rt:5，rd:5，0:5，0:6

7.条件指令

1.相等转移 BEQ
格式：beq rs,rt,L
执行：PC+=4;if(rs==rt)PC+=L
类型：I
机器码：4:6 ， rs:5， rd:5， imm:16

2.BNE 不等于转移
格式：bne rs,rt,L
执行：PC+=4;if(rs!=rt)PC+=L
类型：I
机器码：5:6 ， rs:5， rd:5， imm:16

8.存储指令：

1.读 2 字节 LH
格式：lh rt,D(rs)
执行：rt=Memory[rs+D]
类型：I
机器码：33:6 ， rs:5， rd:5， imm:16

2.读 2 字节，高位无符号扩展 LHU
格式：lhu rt,D(rs)
执行：rt=Memory[rs+D]
类型：I
机器码：37:6 ， rs:5， rd:5， imm:16

3.高位立即数 LUI
格式：lui rt,imm
执行：rt 高=imm
类型：I
机器码：15:6 ， 0:5， rd:5， imm:16

4.读四字节 LW
格式：lw rt,D(rs)
执行：rt=Memory[rs+D]
类型：I
机器码：35:6 ， rs:5， rd:5， imm:16

5.写 2 字节 SH
格式：sh rt,D(rs)
执行：Memory[D+rs]=rt
类型：I
机器码：41:6 ， rs:5， rd:5， imm:16

6.写 4 字节 SW
格式：sw rt,D(rs)
执行：Memory[D+rs]=rt
类型：I
机器码：43:6 ， rs:5， rd:5， imm:16

三、伪指令

伪指令相当于 X86 里的宏，不同在于不能伪指令自我定义。为了方便编程而设。相当于一至三条真实指令。

1.MOVE

格式：move rd,rs

功能：rd=rs

真实指令：add rd,rs,$zero

说明：数据拷贝

2.LA

格式：la rd,imm(rs)

功能：rd=rs+imm

真实指令：addi rd,rs,imm

说明：取地址

3.ABS

格式：abs rd,rs

功能：rd=|rs|

说明： 取绝对值

4LI

格式：li rd,imm

功能：rd=imm

说明：立即数赋值

5.NOP

格式：nop

功能：小延时

真实指令：sll $zero,$zero,0

说明：空操作

四、格式指令

格式指令主要指示汇编器产生机器码的方式。说明程序变量。代码装载地址。等。相当于 X86 汇编里的伪指令。

序号	名称	类	格式	说明	备注
1	.2byte	数据	变量：.2byte.<表>	2 字节变量，变量为起始地址	变量用 10 进制字符标示 例如： A: .ascii d B: .float 12.3
3	.8byte	数据	变量：.8byte.<表>	8 字节变量，变量为起始地址	
4	.double	数据	变量：.double<表>	双精度变量	
5	.word	数据	变量：.word<表>	4 字节变量	
7	.float	数据	变量：.float <表>	单精度浮点数变量	
9	.ascii	数据	变量：.ascii <表>	字符变量	
10	.asciiz	数据	变量：.asciiz <表>	字符串变量	
11	.data	格式		数据段	
12	.text	格式		代码段	

五、错误信息

汇编器能够处理程序中的错误，显示错误原因、行号。程序员可根据显示错误信息修改程序。

Error 1：can't find file XXXX (找不到文件)
Error 2: Symbol not defined （符号未定义）
Error 3：Extra characters on line （多余的字符）
Error 4： Wrong type of register （寄存器出错）
Error5： Operand type must match （操作数类型不匹配）
Error6 : Value is out of range (数值太大,超过允许值.例如:"MOV AL ,100H)

六、总结

所选指令为最小指令集。由于硬件资源的限制，没有乘除法指令、浮点指令，一些操作相对比较麻烦。

下一步可选的扩展指令。进一步改进方向。

附录：

表格 1 Appendix 1 错误信息一览表

序号	错误提示	描述
1	Error 1：can't find file XXXX	找不到文件（代码文件）
2	Error 2: Symbol not defined	符号未定义
3	Error 3：Extra characters on line	多余的字符
4	Error 4： Wrong type of register	寄存器出错
5	Error5： Operand type must match	操作数类型不匹配
6	Error6 : Value is out of range	数值太大,超过允许值.例如:"MOV AL ,100H

表格 2 Appendix 2 Instruction Set

序	指令	功能	格式	执行	型	类	OP	RS	RT	RD	SFT	备注
0	--	--	--	--	-	-	6	5	5	15		<-位数)
1	addi	立即数加，有溢出	addi rt,rs,imm	rt=rs+imm	I	算术	8	rs	rt	.		imm 符号扩展
2	addiu	立即数加，不溢出	addiu rt,rs,imm	rt=rs+imm	I	算术	9	rs	rt	.		imm 符号扩展
3	andi	立即数与	addi rt,rs,imm	rt=rs&imm	I	逻辑	0x0C	rs	rt	.		imm 无符号扩展(填 0)
4	beq	相等转移	beq rs,rt,L	PC+=4;if(rs==rt)PC+=L	I	条件	4	rs	rt	.		.
5	bne	不等于转移	bne rs,rt,L	PC+=4;if(rs!=rt)PC+=L	I	条件	5	rs	rt	.		.
6	lh	读 2 字节	lh rt,D(rs)	rt=Memory[rs+D]	I	存储	0x21	rs	rt	.		.
7	lhu	读 2 字节	lhu rt,D(rs)	rt=Memory[rs+D]	I	存储	0x25	rs	rt	.		高位无符号扩展
8	lui	高位立即数	lui rt,imm	rt 高=imm	I	数据	0x0F	0	rt	.		.
9	lw	读字(4 字节)	lw rt,D(rs)	rt=Memory[rs+D]	I	存储	0x23	rs	rt	.		.
10	ori	立即数或	ori rt,rs,imm	rt=rs|imm	I	逻辑	0x0D	rs	rt	.		imm 无符号扩展(填 0)
11	sh	写 2 字节	sh rt,D(rs)	Memory[D+rs]=rt	I	存储	0x29	rs	rt	.		.
12	slti	小于立即数	slti rt,rs,imm	rd=0;if(rs< td=""><>	I	比较	0x0A	rs	rt	.		.
13	sltiu	小于无符号数	sltiu rt,rs,imm	rd=0;if(rs< td=""><>	I	比较	0x0B	rs	rt	.		.
14	sw	写字(4 字节)	sw rt,D(rs)	Memory[D+rs]=rt	I	存储	0x2B	rs	rt	.		.
15	xori	立即数异或	xori rt,rs,imm	rd=rs^imm	I	逻辑	0x0E	rs	rt	.		imm 无符号扩展(填 0)
16	j	转移	j L	PC=PC 高 4|(L<<2)	J	转移	2	.				.
17	jal	调子程序	jal L	$ra=PC+4, j L	J	转移	3	.				.
18	add	加法，有溢出	add rd,rs,rt	rd=rs+rt	R	算术	0	rs	rt	rd	0	.
19	addu	加法，不溢出	addu rd,rs,rt	rd=rs+rt	R	算术	0	rs	rt	rd	0	.
20	and	与	and rd,rs,rt	rd=rs&rt	R	逻辑	0	rs	rt	rd	0	.
21	jalr	寄存器转移	jalr rs,rd	$rd=PC+4, PC=rs	R	转移	0	rs	0	rd	0	.
22	jr	寄存器转移	jr rs	PC=rs	R	转移	0	rs	0	0	0	.
23	nor	或非	nor rd,rs,rt	rd=~(rs|rt)	R	逻辑	0	rs	rt	rd	0	.

24	or	或	or rd,rs,rt	rd=rs|rt	R	逻辑	0	rs	rt	rd	0	.
25	sll	逻辑左移	sll rd,rt,sft	rd=rt<<sft< td=""><></sft<>	R	移位	0	0	rt	rd	sft	.
26	slt	比较	slt rd,rs,rt	rd=0;if(rs< td=""><>	R	比较	0	rs	rt	rd	0	有符号比较
27	sltu	小于无符号数	sltu rd,rs,rt	rd=0;if(rs< td=""><>	R	比较	0	rs	rt	rd	0	无符号比较
28	sra	算术右移	sra rd,rt,sft	rd=rt>>>sft	R	移位	0	0	rt	rd	sft	.
29	srl	逻辑右移	srl rd,rt,sft	rd=rt>>sft	R	移位	0	0	rt	rd	sft	.
30	sub	减法，有溢出	sub rd,rs,rt	rd=rs-rt	R	算术	0	rs	rt	rd	0	.
31	subu	减法，不溢出	subu rd,rs,rt	rd=rs-rt	R	算术	0	rs	rt	rd	0	.
32	xor	异或	xor rd,rs,rt	rd=rs^rt	R	逻辑	0	rs	rt	rd	0	.
33	syscall	系统调用	syscall	$v0 功能号，$a 参数	R	系统	0	0	0	0	0	.
34	eret	中断返回	eret	EXL=0;PC=EPC	R	系统	0x10	16	0	0	0	.
35	mfc0	读协 0 寄存器	mfc0 rt,rd	rt=协 0.rd	R	系统	0x10	0	rt	rd	0	.
36	mtc0	写协 0 寄存器	mtc0 rt,rd	协 0.rd=rt	R	系统	0x10	4	rt	rd	0	.
37	abs	绝对值	abs rd,rs	rd=|rs|	伪	算术	.	.	.	.	.	.
38	la	取地址	la rd,imm(rs)	rd=rs+imm	伪	存储	.	.	.	.	.	addi rd,rs,imm
39	li	立即数赋值	li rd,imm	rd=imm	伪	数据	.	.	.	.	.	.
40	move	数据拷贝	move rd,rs	rd=rs	伪	存储	.	.	.	.	.	add rd,rs,$zero
41	nop	空操作	nop	小延时	伪	系统	.	.	.	.	.	sll $zero,$zero,0