一、前言

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

二、汇编指令

符号说明：

rs、rt、rd：通用寄存器。

若从最高位为 31 算起，rs 寄存器在编译结果中占地第 21-25 位，rt 占第 16-20 位，rd 占第 11-15 位。

op：第 26-31 位

imm：16 位补码立即数。

D:偏移量，为 16 位立即数或标号。

L：标号。

sft：偏移占第 6-10 位。

func：功能号，占第 0-5 位。

1、算术指令

1) 寄存器加法 ADD

格式：add 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

注意：加法结果大于 32 位补码，则产生溢出中断。

2) 寄存器加法 ADDU，不溢出

格式：addu rd，rs，rt

执行：rd = rs + rt

类型：R

机器码：0:6，rs:5，rt:5，rd:5，0:5，33:6

3) 立即数加法 ADDI

格式：addi rt，rs，imm

执行：rt = rs + imm

类型：I

机器码：8:6，rs:5，rt:5，imm:16

注意：16 位立即数 imm 按补码方式扩展。加法结果大于 32 位补码，则产生溢出中断。

当立即数 imm 为负时，即为减法。

4) 短乘法 MUL
格式：mul rd,rs,rt
执行：rd=rs*rt
类型：R
机器码：1ch:6,rs:5,rt:5,rd:5,0:5,2:6

5) 乘法 MULT
格式：mult rs,rt
执行：(hi,lo)=rs*rt
类型：R
机器码：0:6,rs:5,rt:5,rd:5,0:5,18h:6

6) 乘法 MULTU
格式：mult rs,rt
执行：(hi,lo)=rs*rt
类型：R
机器码：0:6,rs:5,rt:5,0:5,0:5,19h:6

7) 减法 SUB
格式: sub rd, rs,rt
执行：rd=rs-rt
类型：R
机器码：0:6,rs:5,rt:5,rd:5,0:5,22h:6
注意：16 位立即数 imm 按补码方式扩展。结果大于 32 位补码，则产生溢出中断。

8) 减法 SUBU,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:5,0:5,23h:6

2、逻辑指令

1) 立即数与 ANDI
格式：andi rt，rs，imm
执行：rt = rs&imm
类型：I
机器码：0ch:6，rs:5，rt:5，imm:16

2) 立即数或 ORI
格式：ori rt，rs，imm
执行：rt= rs|imm

类型：I
机器码：0dh:6，rs:5，rt:5，imm:16

3) 立即数异或 XORI
格式： xori rt，rs，imm
执行：rd = rs^imm
类型：I
机器码：0eh:6，rs:5，rt:5，imm:16

4) 小于立即数 SLTI
格式：slti rt，rs，imm
执行：rd =0; if (rs<imm) rt=1;
类型：I
机器码：0ah:6，rs:5，rt:5，imm:16

5) 小于无符号数 SLTIU
格式：sltiu rt，rs，imm
执行：rd =0; if (rs<imm) rt=1;
类型：I
机器码：0bh:6，rs:5，rt:5，imm:16

6) 与 AND
格式：and rd,rs,rt
执行：rd =rs&rt;
类型：R
机器码：0:6，rs:5，rt:5，rd:5,0:5,24h:6

7) 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:5,0:5,27h:6

8) 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, 25h:6

9) 逻辑左移 SLL
格式：sll rd,rt,sft
执行：rd =rt<<sft;
类型：R
机器码：0:6，0:5，rt:5，rd:5, sft:5,0:6

10) 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

11) 逻辑左移 SLLV
格式：sllv　rd,rt,rs
执行：rd =rt<<rs;
类型：R
机器码：0:6，rs:5，rt:5，rd:5, 0:5,4:6

12) 逻辑右移 SRLV
格式：srlv　rd,rt,rs
执行：rd =rt>>rs;
类型：R
机器码：0:6，rs:5，rt:5，rd:5, 0:5,6:6

13) 比较小于 SLT
格式：slt　rd, rs, rt
执行：rd =0;if (rs<rt) rd=1;
类型：R
机器码：0:6，rs:5，rt:5，rd:5, 0:5,2ah:6

14) 小于无符号 SLTU
格式：sltu　rd, rs, rt
执行：rd =0;if (rs<rt) rd=1;
类型：R
机器码：0:6，rs:5，rt:5，rd:5, 0:5,2bh:6

15) 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, 26h:6

3、数据传送

1)　读 16 位 LH
格式:　lh rt, D(rs);
执行：rt=Memory[rs+D];
类型：I

机器码：21h:6，rs:5，rt:5 , 0:5, D:16

2) 读 16 位 LHU
格式: lhu rt, D(rs);
执行：rt=Memory[rs+D];
类型：I
机器码：25h:6，rs:5，rt:5 , 0:5, D:16

3) 读 32 位 LW
格式: lw rt, D(rs);
执行：rt=Memory[rs+D];
类型：I
机器码：23h:6，rs:5，rt:5 , 0:5, D:16

4) 高位立即数 LUI
格式: lui rt, imm;
执行：rt 高=imm;
类型：R
机器码：0fh:6，0:5，rt:5 , 0:5, D:16

5) 写 16 位 SH
格式: sh rt, D(rs);
执行：Memory[rs+D]=rt;
类型：R
机器码：29h:6，rs:5，rt:5 , 0:5, D:16

6) 写 32 位 SW
格式: sw rt, D(rs);
执行：Memory[rs+D]=rt;
类型：R
机器码：2bh:6，rs:5，rt:5 , 0:5, D:16

4、控制指令

1) 相等转移 BEQ
格式: beq rs,rt,L
执行：pc+=2;if (rs==rt) pc+=L;
类型：I
机器码：4:6，rs:5，rt:5 , 0:5, L:16

2) 大于等于零转移 BGEZ
格式: bgez rs,L
执行：pc+=2;if (rs>=0) pc+=L;

类型：I
机器码：1:6，rs:5，1:5 , 0:5, L:16

3) 大于等于零调用 BGEZAL
格式: bgezal rs,L
执行：pc+=2;if (rs>=0) { pc+=L; $ra=pc}
类型：I
机器码：1:6，rs:5，17:5 , 0:5, L:16

4) 大于零转移 BGTZ
格式: bgtz rs,L
执行：pc+=2;if (rs>0) pc+=L;
类型：I
机器码：7:6，rs:5，0:5 , 0:5, L:16

5) 小于等于零转移 BLEZ
格式: blez rs,L
执行：pc+=2;if (rs<=0) pc+=L;
类型：I
机器码：6:6，rs:5，0:5 , 0:5, L:16

6) 小于零转移 BLTZ
格式: bltz rs,L
执行：pc+=2;if (rs<0) pc+=L;
类型：I
机器码：1:6，rs:5，0:5 , 0:5, L:16

7) 小于零调用 BLTZAL
格式: bltzal rs,L
执行：pc+=2;if (rs<0) {pc+=L; $ra=pc}
类型：I
机器码：1:6，rs:5，16:5 , 0:5, L:16
8) 不等于转移 BNE
格式: bne rs,rt,L
执行：pc+=2;if (rs!=rt) pc+=L;
类型：I
机器码：5:6，rs:5，rt:5 , 0:5, L:16

9) 转移 J
格式: j L
执行：pc=pc 高 4 位|(L<<2)
类型：j
机器码：2:6,L：26

10) 调用子程序 JAL
格式: jal L
执行：$ra=pc+2; j L
类型：j
机器码：3:6,L：26

11) 中断返回 ERET
格式: eret
执行：
类型：R
机器码：10h:6, 0:5, 0:5, 0:5, 0:5

12) 寄存器转移 JALR
格式: jalr rs,rd
执行：$rd=pc+2,pc=$rs
类型：R
机器码：0:6, rs:5, 0:5, rd:5, 0:5, 9:6

13) 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, rd:5, 0:5, 9:6

14) 读高字 MFHI
格式: mfhi rd
执行：rd=hi
类型：R
机器码：0:6, 0:5, 0:5, rd:5, 0:5, 10h:6

15) 读低字 MFLO
格式: mflo rd
执行：rd=lo
类型：R
机器码：0:6, 0:5, 0:5, rd:5, 0:5, 12h:6

16) 写高字 MTHI
格式: mthi rs
执行：hi = rs
类型：R
机器码：0:6, 0:5, 0:5, rd:5, 0:5, 11h:6

17) 写低字 MTLO
格式: mtlo rs

执行：lo= rs
类型：R
机器码：0:6, 0:5, 0:5, rd:5, 0:5, 13h:6

18) 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, 0ch:6

三、伪指令

1) 绝对值 ABS
格式：abs rd,rs
功能：rd=|rs|
真实指令：
```
addu rd,$zero,rs
bgez rs,1
sub rd,$zero,rs
```

2) 转移 B
格式：b L
功能：pc=pc 高 4|(L<<2)
真实指令：bgez $zero, L格式：

3) 等于零转移 BEQZ
格式：beqz rs,L
功能：pc+=2;if (rs==0) pc+=L;
真实指令：
```
beq rs,$zero,L
```

4) 大于等于零转移 BGEU
格式：bgeu rs,rt,L
功能：pc+=2;if (rs>=rt) pc+=L;
真实指令：
```
sltu $at,rs,rt
beq $zero,$at,L
```

5) 大于等于零转移 BGE
格式：bge rs,rt,L
功能：pc+=2;if (rs>=rt) pc+=L;
真实指令：
```
slt $at,rs,rt
beq $zero,$at,L
```

6) 大于转移 BGT

格式：bgt rs,rt,L
功能：pc+=2;if (rs>rt) pc+=L;
真实指令：slt $at,rt,rs
bne $zero,$at,L

7) 大于转移 BGTU
格式：bgtu rs,rt,L
功能：pc+=2;if (rs>rt) pc+=L;
真实指令：sltu $at,rt,rs
bne $zero,$at,L

8) 小于等于转移 BLE
格式：ble rs,rt,L
功能：pc+=2;if (rs<=rt) pc+=L;
真实指令：slt $at,rt,rs
beq $zero,$at,L

9) 小于等于转移 BLEU
格式：bleu rs,rt,L
功能：pc+=2;if (rs<=rt) pc+=L;
真实指令：sltu $at,rt,rs
beq $zero,$at,L

10) 小于等于转移 BLT
格式：blt rs,rt,L
功能：pc+=2;if (rs<rt) pc+=L;
真实指令：slt $at,rs,rt
bne $zero,$at,L

11) 小于等于转移 BLTU
格式：bltu rs,rt,L
功能：pc+=2;if (rs<rt) pc+=L;
真实指令：sltu $at,rs,rt
bne $zero,$at,L

12) 不等于零转移 BNEZ
格式：bnez rs,L
功能：pc+=2;if (rs!=0) pc+=L;
真实指令：bne rs,$zero,L

13) 取地址 LA
格式：la rd,L(rs)
功能：rd=rs+L
真实指令：addi rd,rs,imm

14) 立即数赋值 LI

格式：li rd,imm

功能：rd=imm

真实指令：
```
lui $at,imm
ori rd,$at,imm
```

15) 数据拷贝 MOVE

格式：move rd,rs

功能：rd=rs

真实指令：
```
add rd,rs,$zero
```

16) 取相反数 NEG，有溢出

格式：neg rd,rs

功能：rd=-rs

真实指令：
```
sub rd,$zero,rs
```

17) 取相反数 NEGU，无溢出

格式：negu rd,rs

功能：rd=-rs

真实指令：
```
subu  rd,$zero,rs
```

18) 空 NOP

格式：nop

功能：小延时

真实指令：
```
sll $zero,$zero,0
```

19) 非 NOT

格式：not rd,rs

功能：rd=-rs

真实指令：
```
nor rd,rs,$zero
```

20) 循环左移 ROL

格式：rol rd,rs,rt

功能：rd=rs<<rt

真实指令：
```
subu $at,$zero,rt
srlv $at,rs,$at
sllv rd,rs,rt
or rd,rd,$at
```

21) 循环右移 ROR

格式：ror rd,rs,rt

功能：rd=rs>>rt

真实指令：
```
subu $at,$zero,rt
```

```
sllv $at,rs,$at
srlv rd,rs,rt
or rd,rd,$at
```

22) 大于等于置 1 SGE

格式：sge rd,rs,rt

功能：rd=(rs>=rt)?1:0

真实指令：
```
bne rt,rs,2
ori rd,$zero,1
beq $zero,$zero,1
slt rd,rt,rs
```

23) 大于等于置 1 SGEU

格式：sgeu rd,rs,rt

功能：rd=(rs>=rt)?1:0

真实指令：
```
bne rt,rs,2
ori rd,$zero,1
beq $zero,$zero,1
sltu rd,rt,rs
```

24) 小于等于置 1 SLE

格式：sle rd,rs,rt

功能：rd=(rs<=rt)?1:0

真实指令：
```
bne rt,rs,2
ori rd,$zero,1
beq $zero,$zero,1
slt rd,rs,rt
```

25) 小于等于置 1 SLEU

格式：sleu rd,rs,rt

功能：rd=(rs<=rt)?1:0

真实指令：
```
bne rt,rs,2
ori rd,$zero,1
beq $zero,$zero,1
sltu rd,rs,rt
```

26) 不等于置 1 SNE

格式：sne rd,rs,rt

功能：rd=(rs!=rt)?1:0

真实指令：
```
beq rt,rs,2
ori rd,$zero,1
beq $zero,$zero,1
ori rd,$zero,0
```

27) 等于置 1 SEQ

格式：seq rd,rs,rt

功能：rd=(rs==rt)?1:0

真实指令：
```
beq rt,rs,2
ori rd,$zero,0
beq $zero,$zero,1
ori rd,$zero,1
```

28) 大于置 1 SGT

格式：sgt rd,rs,rt

功能：rd=(rs>rt)?1:0

真实指令：
```
slt rd,rt,rs
```

29) 大于置 1 SGTU

格式：sgtu rd,rs,rt

功能：rd=(rs>rt)?1:0

真实指令：
```
sltu rd,rt,rs
```

四、格式指令

源文件分为.data 和.text 两部分。.data 如果不进行数据段的定义则可以省略不写，如果写入.data，则程序代码起始增加一句 j 指令，以保证程序运行时跳转到代码段的起始位置。.text 不可以省略，否则会报错。源代码必须有起始标号以及.end+起始标号表示程序的起始位置以及结束位置，否则会报错。

格式指令开始前应有该部分数据的标号。

标号规定不可以使用英文字母及下划线以外的内容起始，以冒号加空格结束。可以与代码或者格式指令的类型处在一行。

以下为现定义的几种数据输入类型：

.asciiz （注意后面有空格）为输入字符串，一行一个字符串，逗号内容之间连接，引号内为字符串表达，无引号为字符编码，每个字符串生成后编译器自动补结束符。

例：.asciiz ‘字符串 1’,10,’字符串 2’

编译结果为“字符串 1a 字符串 2\0”所表示的编码表的字符，其中\0 为字符串结束符。

.db 为 int 数据，用逗号分隔输入，结尾不应有逗号。

dup（n）表示做 n 个同样的数据，数据类型与前面声明一样，数据内容与后面定义相同。如 .db dup(3) 1,2 则实际结果为.db 1,2,1,2,1,2

五、表达式

汇编程序中的立即数均可用表达式。表达式支持算术、逻辑运算。不支持浮点数。支持十六进制立即数(0ffh 格式表示)。支持多重括号。

例：

```
slti    $a1, $a1, 5+5*6
sll    $s2,$s2,66h<<3
la       UIOY,(19h/(3+5))($s2)
```

以上都是正确可识别的表达。

六、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。

错误号：wrong expression

表达式处理时出现错误

错误号：error in translate: ***

翻译***句真指令时表达出现错误