**中间代码定义**

**操作数**

整形常量NUM 例如 15

$tempNO 编号为No的临时变量 例如$temp1

**赋值语句**

mov src dest其中dest为目的操作数 src为源操作数

**运算表达式:**

op opnd1 opnd2 result 或 op opnd1 result 以op为操作码、opnd1（和opnd2）为操作数的二元运算表达式或一元运算表达式，result为操作所得到的结果，操作码含义如下表：

|  |  |  |
| --- | --- | --- |
| 操作码 | 操作数数目 | 含义 |
| add | 2 | opnd1+opnd2 |
| sub | 2 | opnd1-opnd2 |
| mul | 2 | opnd1\*opnd2 |
| neg | 1 | -opnd1 |
| and | 2 | opnd1&&opnd2 |
| or | 2 | opnd1||opnd2 |
| not | 1 | !opnd1 |

**无条件转移语句：**

jmp label 其中lable是一个代表转移目标的符号标号

**条件转移语句**

jmpcode opnd1 opnd2 label

如jeq $temp1 $temp2 \_L0 (set flag $temp3)，其中jmpcode表示转移码，opnd1和opnd2表示待比较的两个操作数，各转移码及其含义如下表：

|  |  |
| --- | --- |
| 转移码 | 含义 |
| jeq | 如果opnd1==opnd2，则转移 |
| jne | 如果opnd1!=opnd2，则转移 |
| jlt | 如果opnd1<opnd2，则转移 |
| jle | 如果opnd1<=opnd2，则转移 |
| jgt | 如果opnd1>opnd2，则转移 |
| jge | 如果opnd1>=opnd2，则转移 |

另：ret 表示程序结束

**符号标号语句**

Label: stmts 可以指示转移语句的目标，例\_L1: mov $temp1 $temp2

**输入输出语句**

read $temp1 输入到$temp1

print $temp1 输出$temp1

print NUM 输出整形常量NUM 例如print 15

实例代码

|  |  |
| --- | --- |
| c1语言 | 中间语言 |
| {  int t;  input(t);  if (t > 10) output(-1);  else {  int x;  x = 1;  while (t > 0) {  x = 5 + x \* t;  t = t - 5;  }  output(x);  }  } | Read $temp1  jgt $temp1 10 \_L1  jmp \_L2  \_L1:  print -1  jmp \_L3  \_L2:  mov 1 $temp2  \_L5:  jle $temp1 0 \_L4  mul $temp2 $temp1 $temp3  add 5 $temp3 $temp2  sub $temp1 5 $temp1  jmp \_L5  \_L4:  print $temp2  \_L3:  ret |