

目 录

第 1 章 总体设计方案 .....	0
1.1 设计原理 .....	0
1.2 设计思路 .....	0
1.3 设计环境 .....	0
第 2 章 详细设计方案.....	4
2.1 算法与程序的设计与实现 .....	4
2.2 流程图的设计与实现.....	5
第 3 章 程序调试与结果测试 .....	9
3.1 程序调试 .....	9
列举出调试过程中存在的问题 .....	9
3.2 程序测试及结果分析 .....	9
参考文献 .....	10
附录（源代码）.....	11

# 第 1 章 总体设计方案

## 1.1 设计原理

在计算机中，二进制的基本算术运算多是采用补码进行的。本次课设的补码两位乘是根据补码一位乘推导出来，补码两位乘可以使用和补码一位乘类似的硬件，而计算速度接近补码一位乘的两倍。本次课设需要构造补码两位乘规则表，计算中可以根据查表依次运算下去。

表1 补码两位乘规则表

$Y_{n+1}Y_nY_{n-1}$	$Y_{n+1}+Y_n-2Y_{n-1}$	操作
000	0	部分积+0, 右移2位
001	-2	部分积加2[-X]补, 右移两位
010	1	部分积+[X]补, 右移两位
011	-1	部分积+[-X]补, 右移两位
100	1	部分积+[X]补, 右移两位
101	-1	部分积+[-X]补, 右移两位
110	2	部分积+2[X]补, 右移两位
111	0	部分积+0, 右移两位

## 1.2 设计思路

- (1) 被乘数 X, 乘数 Y 均以补码形式参加运算, 乘法的每一步对乘数的两位进行, 即右移两位。
- (2) 乘数[Y]在计算时采用两位符号位, 初始附加位添一个0。
- (3) 被乘数[X]在计算时采用三位符号位。
- (4) 部分积初始为0, 计算过程中采用三位符号位。
- (5) 每一步计算, 按照乘数的最低三位的值查找操作规则表1, 根据上一步的部分积和规则表的操作, 计算本步的部分积。
- (6) 每步操作后, 将部分积和乘数右移两位, 乘数右移后, 原来的最低两位被移除舍弃, 其余各位位权依次下降。重复(4), 但最后一步操作不右移。

## 1.3 设计环境

利用伟福 COP2000 型计算机组成原理实验仪软件和计算机，在 COP2000 实验仪软件上编程实现补码两位乘。伟福 COP2000 实验仪软件的指令集分为如下大类：算术运算指令、逻辑运算指令、移位指令、数据传输指令、跳转指令、中断

返回指令、输入输出指令。

本程序所用到的相应的指令集如下：

(1) 算术逻辑运算指令

ADD	A,R?	将寄存器 R? 的值加入到累加器A 中
ADD	A,#II	立即数# II 加入到累加器A 中
SUB	A,#II	从累加器中减去立即数后加入到累加器A 中
AND	A,R?	累加器 A 与寄存器 R?的值
OR	A,R?	累加器A 或寄存器 R?的值

(2) 数据传送指令

MOV	A,R?	将寄存器 R?的值送到累加器 A 中
MOV	A,MM	将存储器 MM 地址中的值送到累加器A 中
MOV	MM,A	将累加器 A 中的值送到存储器 MM 地址中
MOV	R?,A	将累加器A 中的值送到寄存器 A 中
MOV	R?#I	将立即数 II 送到寄存器 R?中
RR	A	累加器右移指令
RL	A	累加器左移指令

(3) 跳转指令

JC	MM	若进位标志置1, 跳转到MM 地址
••	•M	若零标志位置1, 跳转到MM 地址
JMP	MM	跳转到 MM

如下为 COP2000 计算机组成原理实验软件截图：



文件(F) 编辑(E) 汇编(A) 运行(R) 帮助(H)

×号 烟明 IA

反汇编 A.ASM EM 程序] 结构图 图示帮助|逻辑分析

```

MOU A,R2
RR A
MOU R2,A
MARK2:
MOU A,9BH
SUB A,#01H
JZ LOOP2
MOU 9BH,A
JMP MARK1
LOOP3:
MOU A,R1
RRC A
JC LOOP5
    
```

IA:E0 ST:00 PC:60 MAR:9B A D:05 R:02 ROEB

IN:00 OUT:0 EM:AD ALU R1:A

uPCAO R2:C6

IR:AD A:05 W:80 R3:96

uM:C6 FF FF

69:1 Insert RT' RTO DBUS:FF ABUS:5F IBUS:A0

指令系统	助记符	状态	微地址	微程序	数据输出	数据打入	地址输出	运算器	移位控制	uPC	PC													
		TO	D9	CBFFFF	浮空	指令寄存器IR	PC输出	A输出		写入														
	5F JC 71	T1	->AD	CBFFFF	存储器值	寄存器助	输出	输出																
	XRI	EMWR	EMRD	PCOE	EMER	TREN	ETRT	FTF	HARER	NARDE	OUTEN	STEN	RRD	BWR	CN	FEN	X2	X1	管	WEN	AEN	S2	≧	≦
	✓				✓				✓		✓		✓	✓	✓	✓	✓	✓	□	✓	✓	✓	✓	□
	PC:60	uPC:A0	A:05	W:8B	C:0	Z:0	RD:EB	R1:0A	R2:C6	R3:96	IR:A0	ST:00	IA:E0	MAR:9B	IN:00	OUT:08	ABUS:5F	DBUS:FF	IBUS:AD			L		

文件(F) 编辑(E) 汇编(A) 运行(R) 帮助(H)

反汇编 A.ASH EM 程序 结构 图示帮助|逻辑分析

```

51 DG RR
52 82 MoU R2
53 789B MaU A,
553C01 SUB A
57842E Z 2E
59889B MOU QB.
5B AC44 JMP
5[ 71 MOU A
5E D8 RRC A
5F AB71 B
61 81 MOU R1.
6272
63 D0 RR A
6482 HOU
65 AC53 JHP R2
    
```

指令系统 | 微程序跟踪

助记符	状态	微地址	微程序	数据输出	数据打入	地址输出	运算器	移位控制	uPC	PC
	TO	D9	CBFFFF	浮空	指令寄存器IR	PC输出	A输出		写入	+
5F JC 71	T1	->AD		CERFTE存储器值	寄存器PC	PC输出	A输出		1	写

图2 COP2000 运行环境



## 第 2 章 详细设计方案

### 2.1 算法与程序的设计与实现

本课设采用伟福 COP2000 实验仪软件和计算机实现补码两位乘功能，利用伟福 COP2000 的指令集编程实现。

算法具体描述：

- (1) 计算复杂性包括： $+ [X]_{补}$ ， $+ 2[X]_{补}$ ， $+ [-X]_{补}$ ， $+ 2[-X]_{补}$ ；
- (2) 进行初始化，并且设置计数单元与建立特殊表。因为每次计算需要进行三次移位，所以用 `MOVA,#03H` 与 `MOV9AH,A` 来将循环次数存入内存单元。然后再进行初始化操作，将存储部分积的 `R1` 置零，再将用来存储特殊表的 `90H,97H` 内存单元置零。接下来再通过输入的被乘数判断其正负，如果被乘数是负，在前面两位加上1，如果是整数则无需操作，直接构造特殊表。
- (3) 输入乘数后，将乘数移至最前端，执行 `LOOP2`，首先设置移位次数，取出成熟后，判断乘数的后两位及附加位的值，根据所查的值访问特殊表，进而确定加数，之后将部分积与加数相加得到新的部分积，得到结果后，判断总循环是否结束，之后进行判断移位的操作，部分积的最高位如果是1，移位后仍然置1。部分积最低位如果是1，乘数右移之后最高位置1。
- (4) 根据乘数后两位及附加位来确定 `R3`，并通过 `R3` 间接访问特殊表。将访问的数据作为加数与部分积相加，然后判断两次循环是否结束，如果两次运算循环结束，则结束运算，否则将部分积与乘数同时右移两位再根据有效位来通过 `R3` 访问特殊表。相加移位运算结束后再将部分积与乘数同时右移一位，将数据位置有效，输出数据。





## 2.2 流程图的设计与实现

本部分包含实现中主要部分的流程图

### 1. 构造规则表:

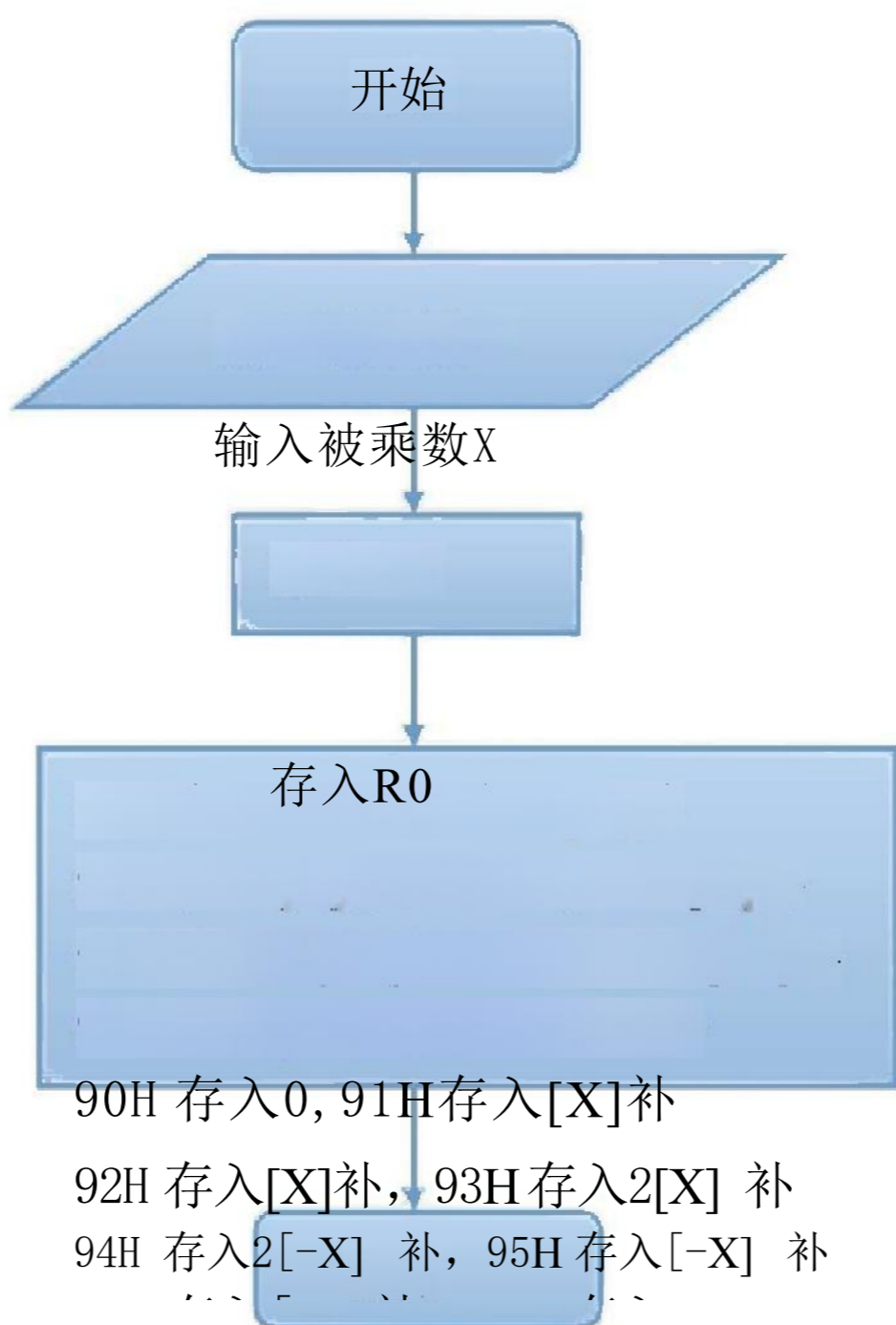


图2. 2. 1 构造规则表流程图



2. 乘数处理操作:

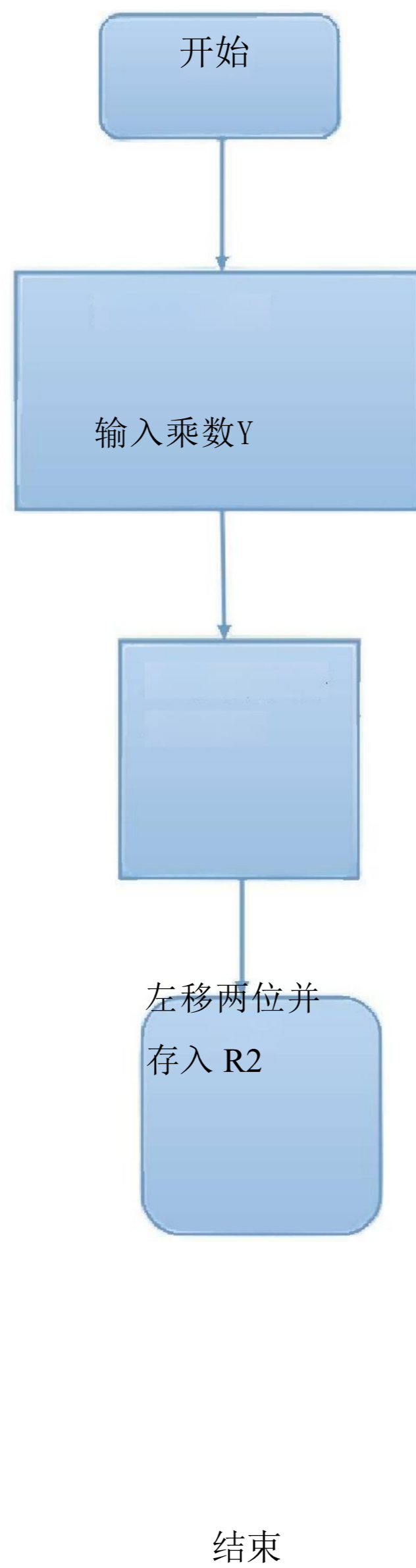


图2.2.2 乘数处理操作流程图



3. 运算

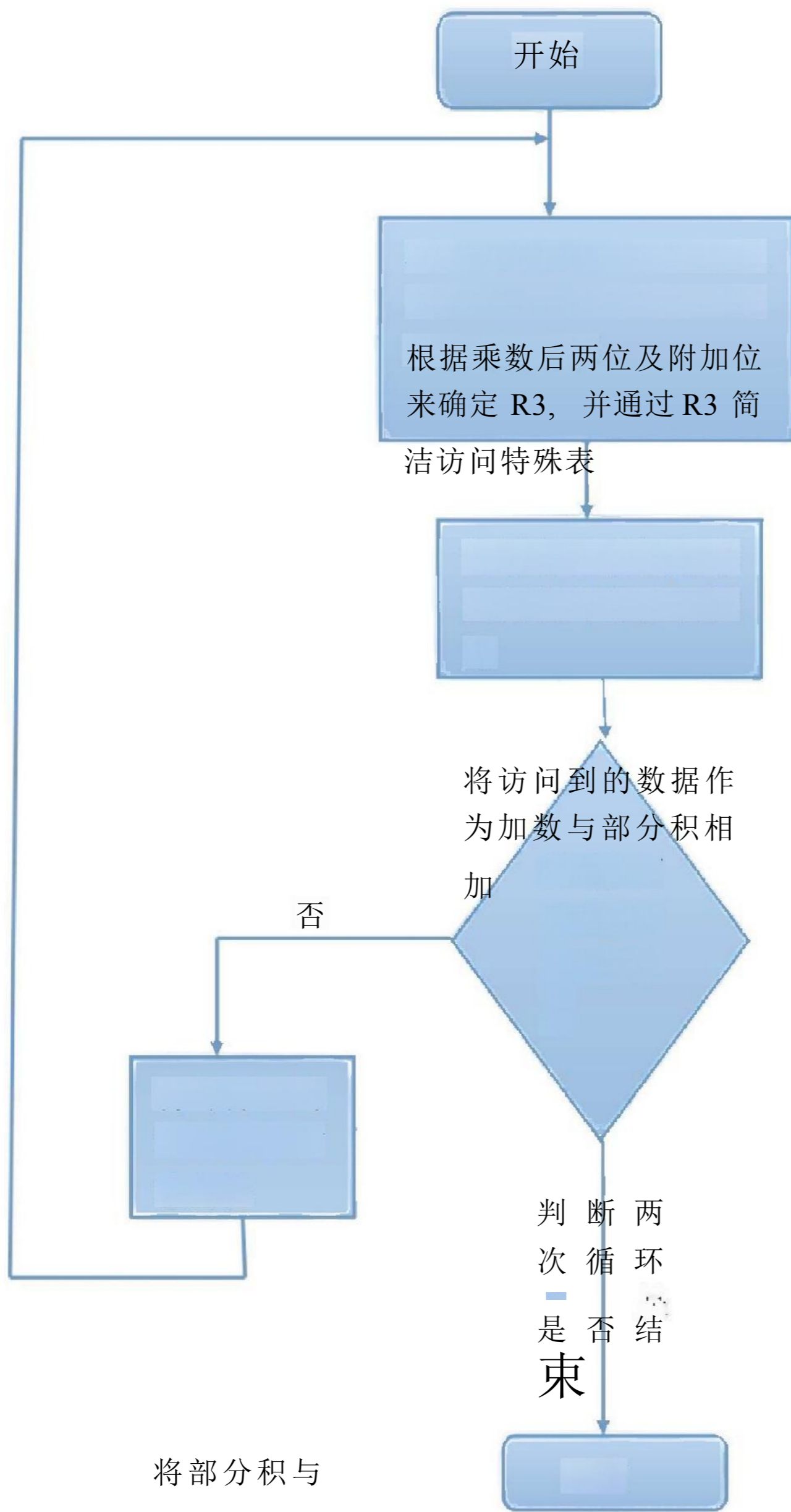


图2.2.3 运算流程图

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：  
<https://d.book118.com/047054035061006064>