

目录

一、实验目的	1
二、实验内容.....	1
三、实验原理（16 分）	1
3.1、指令系统及分析（4 分）	1
3.2、指令框图及分析（4 分）	3
3.3、指令系统对应微程序二进制代码及分析（4 分）	4
3.4、机器程序及分析（4 分）	6
四、实验步骤（4 分）	10
4.1、微程序写入及校验（2 分）	10
4.2、机器程序写入及校验（2 分）	11
五、实验结果及分析（16 分）	11
5.1、演示程序一（8 分）	11
5.2、演示程序二（8 分）	12
六、实验问题及思考（4 分）	13
七、实验验收答辩环节问题和解答（20 分）	14

一、实验目的

综合运用所学计算机组成原理知识，设计并实现较为完整的计算机。

二、实验内容

1. 设计并实现一套完整的指令系统；
2. 设计并实现完整的计算机（采用上述指令系统）；
3. 利用该计算机实现用海伦公式求解三角形面积。

海伦公式即：

$$S = \sqrt{p(p-a)(p-b)(p-c)}$$

$$p = \frac{1}{2}(a+b+c)$$

从 IN 单元输入三角形的三个边长a,b,c，输出三角形边长S。

三、实验原理（16 分）

3.1、指令系统分析（4 分）

模型机指令分为 3 大类，包括运算类指令、控制转移类指令和数据传送类指令，共 15 条，同时为了方便实现海伦公式，又新增一条 SAR 指令。

运算类指令包括逻辑运算、移位运算和算术运算，设计有 ADD、AND、DEC、SUB、OR、RR，以及新增的 SHR，共 7 条。所有运算类指令均为单字节指令，寻址方式采用寄存器直接寻址。

控制转移类指令包括 HLT、JMP、BZC，共 3 条。控制转移类指令用以控制程序的分支和转移，其中 HLT 为单字节指令，JMP 和 BZC 为双字节指令。

数据传送类指令有 IN、OUT、MOV、LDI、LAD、STA，共六条。数据传送类指令用以完成寄存器和寄存器、寄存器和 IO 设备、寄存器和存储器之间的数据交换，除了 MOV 指令为单字节指令外，其余均为双字节指令。

模型机各条指令的汇编符号、指令格式和指令功能如下表：

表 1 指令描述

汇编符号	指令格式				指令功能
MOV RD, RS	0100	RS	RD		RS→RD
ADD RD, RS	0000	RS	RD		RD+RS→RD
SUB RD, RS	1000	RS	RD		RD-RS→RD
AND RD, RS	0001	RS	RD		RD∧RS→RD
OR RD, RS	1001	RS	RD		RD∨RS→RD
RR RD, RS	1010	RS	RD		RS 右环移→RD
SHR RD, RS	1011	RS	RD		RS 逻辑右移→RD
DEC RD	0111	**	RD		RD-1→RD
LAD M D, RD	1100	M	RD	D	E→RD
STA M D, RS	1101	M	RD	D	RD→E
JMP M D	1110	M	**	D	E→PC
BZC M D	1111	M	**	D	FC=1 或 FZ=1 时, E→PC
IN RD, P	0010	**	RD	P	[P]→RD
OUT P, RS	0011	RS	**	P	RS→[P]
LDI RD, D	0110	**	RD	D	D→RD
HALT	0101	**	**		停机

3.2、指令框图及分析（4分）

根据机器指令系统要求，设计微程序流程图，如下图所示：

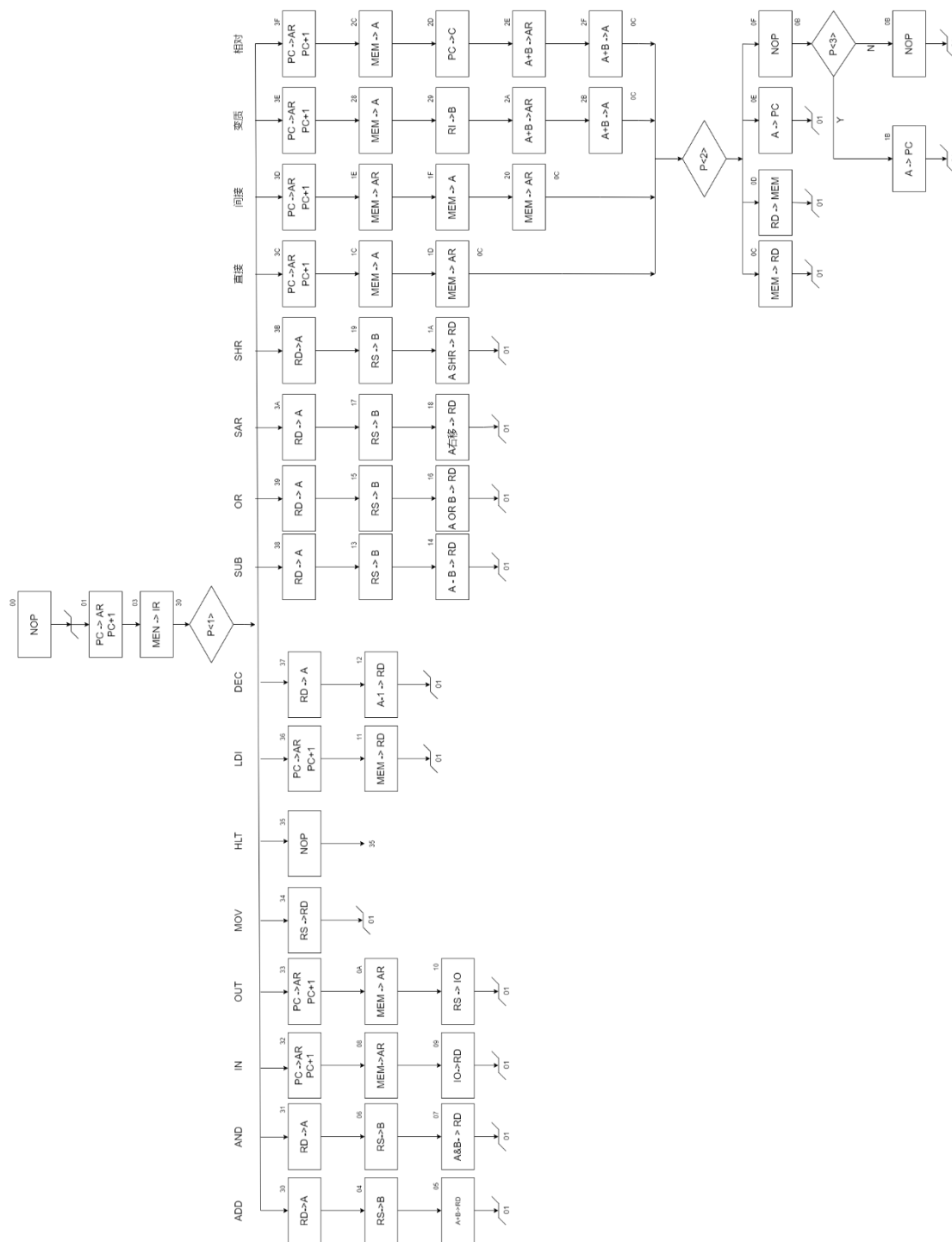


图 1 微程序流程图

3.3、指令系统对应微程序二进制代码及分析（4 分）

系统建议的微指令格式如下：

表 2 微指令格式

23	22	21	20	19	18-15	14-12	11-9	8-6	5-0
M23	CN	WR	RD	IOM	S3-S0	A 字段	B 字段	C 字段	UA5-UA0

A 字段

14	13	12	选择
0	0	0	NOP
0	0	1	LDA
0	1	0	LDB
0	1	1	LDRi
1	0	0	保留
1	0	1	LOAD
1	1	0	LDAR
1	1	1	LDIR

B 字段

11	10	9	选择
0	0	0	NOP
0	0	1	ALU_B
0	1	0	RS_B
0	1	1	RD_B
1	0	0	RI_B
1	0	1	保留
1	1	0	PC_B
1	1	1	保留

C 字段

8	7	6	选择
0	0	0	NOP
0	0	1	P<1>
0	1	0	P<2>
0	1	1	P<3>
1	0	0	保留
1	0	1	LDPC
1	1	0	保留
1	1	1	保留

参照微程序流程图，将每条微指令代码化，译成二进制代码表如下：

表 3 二进制代码表

地址	十六进制表示	高五位	S3-S0	A 字段	B 字段	C 字段	UA5-UA0
00	00 00 01	00000	0000	000	000	000	000001
01	00 6D 43	00000	0000	110	110	101	000011
03	10 70 70	00010	0000	111	000	001	110000
04	00 24 05	00000	0000	010	010	000	000101
05	04 B2 01	00000	1001	011	001	000	000001
06	00 24 07	00000	0000	010	010	000	000111
07	01 32 01	00000	0010	011	001	000	000001
08	10 60 09	00010	0000	110	000	000	001001
09	18 30 01	00011	0000	011	000	000	000001

0A	10 60 10	00010	0000	110	000	000	010000
0B	00 00 01	00000	0000	000	000	000	000001
0C	10 30 01	00010	0000	011	000	000	000001
0D	20 06 01	00100	0000	000	0011	000	000001
0E	00 53 41	00000	0000	101	001	101	000001
0F	00 00 CB	00000	0000	000	000	011	001011
10	28 04 01	00101	0000	000	010	000	000001
11	10 30 01	00010	0000	001	000	000	000001
12	06 32 01	00000	1100	011	001	000	000001
13	00 24 14	00000	0000	010	010	000	010100
14	05 B2 01	00000	1011	011	001	000	000001
15	00 24 16	00000	0000	010	010	000	010110
16	01 B2 01	00000	0011	011	001	000	000001
17	00 24 18	00000	0000	010	010	000	011000
18	02 B2 01	00000	0101	011	001	000	000001
19	00 24 1A	00000	0000	010	010	000	011010
1A	03 32 01	00000	0110	011	001	000	000001
1B	00 53 41	00000	0000	101	001	101	000001
1C	10 10 1D	00010	0000	001	000	000	011101
1D	10 60 8C	00010	0000	110	000	010	001100
1E	10 60 1F	00010	0000	110	000	000	011111
1F	10 10 20	00010	0000	001	000	000	100000
20	10 60 8C	00010	0000	110	000	010	001100
28	10 10 29	00010	0000	001	000	000	101001
29	00 28 2A	00000	0000	010	100	000	101010
2A	04 E2 2B	00000	1001	110	001	000	101011
2B	04 92 8C	00000	1001	001	001	010	001101
2C	10 10 2D	00010	0000	001	000	000	101101
2D	00 2C 2E	00000	0000	010	110	000	101110

2E	04 E2 2F	00000	1001	110	001	000	101111
2F	04 92 8C	00000	1001	001	001	010	001100
30	00 16 04	00000	0000	001	011	000	000100
31	00 16 06	00000	0000	001	011	000	000110
32	00 6D 48	00000	0000	110	110	101	001000
33	00 6D 4A	00000	0000	110	110	101	001010
34	00 34 01	00000	0000	011	010	000	000001
35	00 00 35	00000	0000	000	000	000	110101
36	00 6D 51	00000	0000	110	110	101	010001
37	00 16 12	00000	0000	001	011	000	010010
38	00 16 13	00000	0000	001	011	000	010011
39	00 16 15	00000	0000	001	011	000	010101
3A	00 16 17	00000	0000	001	011	000	010111
3B	00 16 19	00000	0000	001	011	000	011001
3C	00 6D 5C	00000	0000	110	110	101	011100
3D	00 6D 5E	00000	0000	110	110	101	011110
3E	00 6D 68	00000	0000	110	110	101	101000
3F	00 6D 6C	00000	0000	110	110	101	101100

3.4、机器程序及分析（4 分）

机器程序大致分为两部分，即先求三角形面积的平方和，再将该数开方，流程图如下：

7

开方过程算法如下：

由等差序列公式可知，从 1 开始的 n 个奇数之和等于 n^2 ，即

$$1 + 3 + \dots + (2n - 1) = \frac{n(2n - 1 + 1)}{2} = n^2$$

因此计算 a 的开方时，只需从 a 项奇数开始依次计算等差序列和，将项数减一并比较结果和被开方数，当序列和等于 a 或第一次小于 a 时，输出结果为项数+1（因为在比较前项数减了 1）。易知，输出的结果是 \sqrt{a} 的整数部分。

根据上述要求可以得到以下程序，地址和内容均为二进制数：

地址	内容	助记符
00000000	00100000	IN R0,00H
00000001	00000000	
00000010	00100001	IN R1,00H
00000011	00000000	
00000100	00100010	IN R2,00H
00000101	00000000	
00000110	11010000	STA 80H,R0
00000111	10000000	
00001000	00000100	ADD R0,R1
00001001	00001000	ADD R0,R2
00001010	01100011	LDI R3,01H
00001011	00000001	
00001100	10111100	SHR R0,R3
00001101	01000011	MOV R3,R0
00001110	10000100	SUB R0,R1
00001111	01001101	MOV R1,R3
00010000	10001001	SUB R1,R2
00010001	01000010	MOV R2,R0
00010010	11010011	STA 70H,R3
00010011	01110000	
00010100	01100011	LDI R3,01H
00010101	00000001	
00010110	01110001	DEC R1
00010111	11110000	BZC RESULT1
00011000	00011111	
00011001	00001000	LOOP1: ADD R0,R2
00011010	01110001	DEC R1
00011011	11110000	BZC RESULT1
00011100	00011111	
00011101	11100000	JMP LOOP1
00011110	00011001	

00011111	11000001	RESULT1: LAD 00,70H
00100000	01110000	
00100001	01110001	DEC R1
00100010	11110000	BZC RESULT2
00100011	00101011	
00100100	01000010	MOV R2,R0
00100101	00001000	LOOP2: ADD R0,R2
00100110	01110001	DEC R1
00100111	11110000	BZC RESULT2
00101000	00101011	
00101001	11100000	JMP LOOP2
00101010	00100101	
00101011	11000010	RESULT2: LAD 00 R2,80H
00101100	10000000	
00101101	11000001	LAD 00 R1,70H
00101110	01110000	
00101111	10001001	SUB R1,R2
00110000	01000010	MOV R2,R0
00110001	01110001	DEC R1
00110010	11110000	BZC RESULT3
00110011	00111010	
00110100	00001000	LOOP3: ADD R0,R2
00110101	01110001	DEC R1
00110110	11110000	BZC RESULT3
00110111	00111010	
00111000	11100000	JMP LOOP3
00111001	00110100	
00111010	01100010	RESULT3: LDI R2,0
00111011	00000000	
00111100	11010000	STA 81H,R0
00111101	10000001	
00111110	11010000	STA 82H,R0
00111111	10000010	
01000000	01000010	MOV R2,R0
01000001	01100011	LOOP6: LDI R3,1
01000010	00000001	
01000011	01100001	LDI R1,1
01000100	00000001	
01000101	01100000	LDI R0,0
01000110	00000000	
01000111	00000100	LOOP5: ADD R0,R1
01001000	00001101	ADD R1,R3
01001001	00001101	ADD R1,R3

01001010	10001110	SUB R2,R3
01001011	11110000	BZC RESULT4
01001100	01001111	
01001101	11100000	JMP LOOP5
01001110	01000111	
01001111	11000001	RESULT4: LAD 00 R1,81H
01010000	10000001	
01010001	11000010	LAD 00 R2,82H
01010010	10000010	
01010011	10001110	SUB R2,R3
01010100	11010010	STA 82H,R2
01010101	10000010	
01010110	10000001	SUB R1,R0
01010111	01100011	LDI R3,7FH
01011000	01111111	
01011001	00011101	AND R1,R3
01011010	01100011	LDI R3,0
01011011	00000000	
01011100	10001101	SUB R1,R3
01011101	11110000	BZC RESULT5
01011110	01100100	
01011111	11000001	LAD 00 R1,81H
01100000	10000001	
01100001	10000001	SUB R1,R0
01100010	11110000	BZC LOOP6
01100011	01000001	
01100100	01100011	RESULT5: LDI R3,1
01100101	00000001	
01100110	00001110	ADD R2,R3
01100111	00111000	OUT 40H,R2
01101000	01000000	
01101001	01010000	HALT

四、实验步骤（4 分）

4.1、微程序写入及校验（2 分）

将微程序写入.txt 文件中，选择“转储”→“装载数据”命令，将该文件装载入实验系统。选择“转储”→“刷新指令区”命令，读出下位机所有的机器指令和微指令，校验微指令后，发现微指令和文件中相同。

4.2、机器程序写入及校验（2分）

将机器指令程序写入.txt 文件中，选择“转储-装载数据”命令，将该文件装载入实验系统。选择“转储”→“刷新指令区”命令，读出下位机所有的机器指令和微指令，校验机器指令后，发现机器指令和文件中相同。

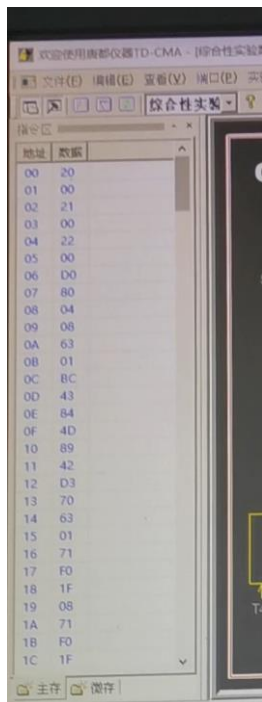


图 3 机器程序校验示意图

五、实验结果及分析（16分）

5.1、演示程序一（8分）

数据（3分）：

输入：3H, 3H, 3H

结果（2分）：

输出：2H

分析（3分）：

在海伦公式的计算过程中，由于半周长并非整数，舍入后得到面积的平方

$$4 \times (4 - 3) \times (4 - 3) \times (4 - 3) = 4$$

开方后便得到结果 2。但是边长为 3 的等边三角形面积应该为 $\frac{4}{9}\sqrt{3} \approx 3.89$ ，因此用该计算机计算三角形面积存在较大误差，该误差主要由舍入引起。

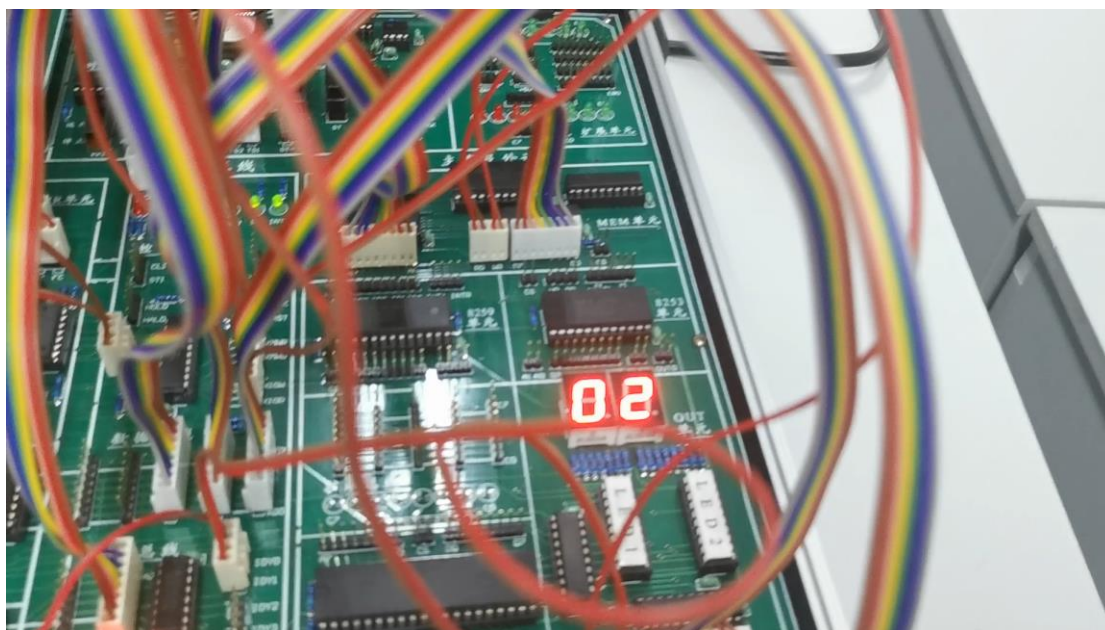


图 4 输出结果示意图

5.2、演示程序二（8 分）

数据（3 分）：

输入：2H, 3H, 3H

结果（2 分）：

输出：2H

分析（3 分）：

当三边长分别为 2, 3, 3 时，三角形半周长正好为 4，则根据海伦公式得到

$$4 \times (4 - 2) \times (4 - 3) \times (4 - 3) = 8$$

开方后得到结果 2。该三角形实际面积为 $2\sqrt{2} \approx 2.83$ ，计算结果是实际面积的整数部分，符合前面的假设。

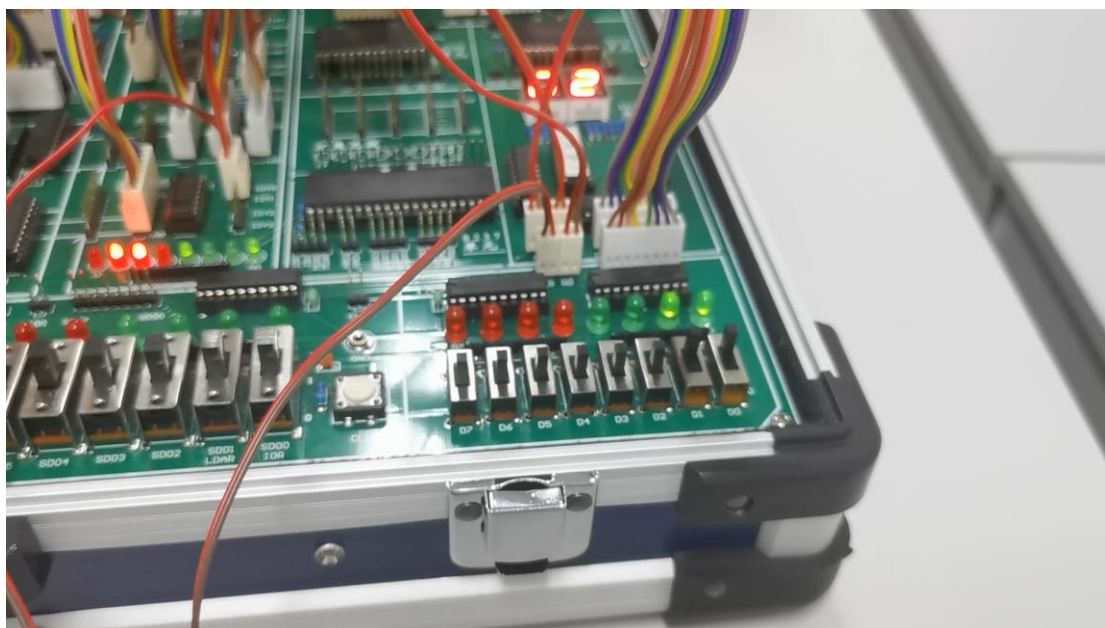


图 5 输出结果示意图

六、实验问题及思考（4 分）

1、当前所实现计算机，是否完整？如果不完整，还缺少哪些部件？

答：

当前所实现计算机，包含了运算器、存储器、控制器、输入设备、输出设备五大功能部件，是完整的。

2、当前所实现计算机，是否能实现除法运算？如果能，可通过哪些指令实现除法运算？

答：

能。由于缺少取反运算，采用不恢复余数法实现除法运算会相对麻烦，但当前计算机仍然可以使用恢复余数法实现除法运算。计算时先做减法，若余数为正，接下来再将除数右移一位相减；若余数为负，则恢复原来的余数，再将除数右移一位相减。

需要的指令有：① 运算类指令：ADD、SUB、SHR

② 控制转移类指令：JMP、BZC

③ 数据传送类指令：MOV

3、当前所实现计算机，还能实现哪些更复杂的计算？请举例说明。

答：

可以实现乘法运算、幂运算、开方运算等。

① 实现乘法运算 $a \cdot b$ 时，只需将 a 与自己相加的过程循环 b 次即可。

② 当前计算机只能实现整数的整数次幂运算，如 x^y 。和乘法运算相同，在实现指数运算 x^y 时，只需将 $x \cdot x$ 的过程循环 $y - 1$ 次即可。

③ 在实现开方运算乃至 $\sqrt[y]{x}$ 时，需将 a^y 和 x 比较，采用顺序搜索或二分查找的方式找到合适的值，得出结果 a 。但由于机器的限制，结果只能以整数的形式表示，因此可能存在较大误差。

4、当前所实现计算机，指令系统的双字长指令是如何实现的？

答：

在取指操作完成后，IR 中的指令内容不变，再进行一次“PC→AR, PC+1”取出第二个字节的地址，并将 PC 移动到下一字节。随后，从主存中取出第二个字节的内容，进行相应的操作。以 LDI 为例，其指令码和寄存器地址作为第一个字节内容保存在 IR 中，在从第二个字节取出操作数 D 后便可进行 D→RD。

七、实验验收答辩环节问题和解答（20 分）

1. 问题：对指令系统有何修改？

答：①修改 INC 为 DEC；②新增 SHR 指令。

2. 问题：如何实现上述修改？

答：

① 通过将地址为 12 的微指令的 S3-S0 段从 1101 改为 1100，修改 INC 为 DEC；

② 将地址为 3B 的微指令修改为 001619（RD→A），并新增地址为 19 的微指令 00241A（RS→B）和地址为 1A 的微指令 033201（A 逻辑右

移→RD), 从而实现新增 SHR 指令。

3. 问题：时钟周期是什么？

答：时钟周期又称节拍脉冲，一个机器周期中包含若干个时钟周期，一个时钟周期中包含若干个节拍脉冲，每个时钟周期完成一次基本的微操作。时钟周期的宽度取决于 CPU 完成一次基本微操作所需的时间。

4. 问题：机器周期和时钟周期有何区别？

答：通常把一条指令周期划分为若干个机器周期，每个机器周期完成一个基本操作；一个机器周期中包含若干个时钟周期，每个时钟周期完成一次基本的微操作。