计算机组成原理课程设计 复杂模型机设计实验报告

一、实验平台

实验设备

TD-CMA 实验系统一套。

硬件资源描述

运算器的数据总线宽度：8位

地址总线宽带：8位

程序存储器的容量：100Hx8位

控制存储器的容量：40Hx8位

输入输出设备及地址：

表 5-3-1　I/O 地址空间分配

A7　A6	选定	地址空间
00	IOY0	00-3F
01	IOY1	40-7F
10	IOY2	80-BF
11	IOY3	C0-FF

二、系统指令集

2.1 机器指令格式

模型机支持3大类共15条指令

- 算术运算指令6条：ADD, AND, INC, SUB, OR,RR
- 控制转移类指令3条：HLT, JMP, BZC
- 数据传送类指令6条：IN, OUT, MOV, LDI, LAD, STA
- ADD, AND, INC, SUB, OR, RR, HLT, MOV为单字长指令

7 6 5 4	3 2	1 0
OP-CODE	RS	RD

RS 或 RD	选定的寄存器
00	R0
01	R1
10	R2
11	R3

- IN, OUT为双字长指令

7 6 5 4 (1)	3 2 (1)	1 0 (1)	7—0 (2)
OP-CODE	RS	RD	P

- LAD, STA, JMP, BZC指令格式

7 6 5 4 (1)	3 2 (1)	1 0 (1)	7—0 (2)
OP-CODE	M	RD	D

寻址模式 M	有效地址 E	说　明
00	E = D	直接寻址
01	E = (D)	间接寻址
10	E = (RI) + D	RI 变址寻址
11	E = (PC) + D	相对寻址

助记符号	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
INC 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	当FC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

将OR改写为CNT指令，用于计数

按直接寻址的方式将操作数读入并打到A，执行A+1->A，将A的数据存回原位

CNT 为双字节指令，采用直接寻址

OP	RD	RS	D
1001	**	**	直接地址

将HALT改为LR，用于左移

将RD的操作数不带进位地左移一位，存回RD

LR 为单字节指令

OP	RD	RS
0101	RD	RS

2.2 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

具体微命令含义

> M23：全是0
>
> CN：ALU选择移位运算时是否带进位（高有效，代表带进位）
>
> WR：写入信号（高有效）
>
> RD：读取信号（高有效）
>
> IOM：选择是对I/O 还是对MEM 进行读写操作
>
> - IOM=0，从I/O读取
> - IOM=1，从MEM读取
>
> S3...S0：选择ALU逻辑功能
>
> A、B、C字段：编码表示的一系列微命令
>
> - A：寄存器写入允许信号
> - B：寄存器输出允许信号
> - C：P<1>、P<2>、P<3>表示微地址转移判别标志，有效时，下条微指令地址根据操作码进行改变，LDPC实现PC+1操作
>
> UA_5...UA_0：下条微指令的地址

2.3 微指令列表

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

地址	十六进制表示	高五位	S3-S0	A 字段	B 字段	C 字段	UA5-UA0
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	011	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	011	000	000	000001
12	06 B2 01	00000	1101	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	06 92 1A	00000	1101	001	001	000	011010
1A	00 02 01	00100	0000	000	001	000	000001
1B	00 53 41	00000	0000	101	001	101	000001
1C	10 10 1D	00010	0000	001	000	000	011001
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

地址	十六进制表示	高五位	S3-S0	A 字段	B 字段	C 字段	UA5-UA0
20	10 60 8C	00010	0000	110	000	010	001100
21	00 24 22	00000	0000	010	010	000	100010
22	03 B2 01	00000	0111	011	001	000	000001
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	001100
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	001	011	000	100001
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	110	110	101	011100
3A	00 16 17	00000	0000	001	011	000	010111
3B	00 00 01	00000	0000	000	000	000	000001
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

三、指令执行过程分析（以LAD(相对寻址)为例）

3.1 指令格式

LAD为双字长指令,其指令码为**1100**

有两位寻址模式码，可以确定四种寻址方式

RD为目标寄存器，D为8位偏移量

7 6 5 4 (1)	3 2 (1)	1 0 (1)	7—0 (2)
OP-CODE	M	RD	D

3.2 指令样例

我们以**1100 11 01 0000 0011**为例，跟踪指令执行过程（假设该指令开始于主存0000 0000位置）

地址	内容
0000 0000	1100 1101
0000 0001	0000 0011

3.3 执行过程

3.3.1 取指译码

微指令从00H地址开始执行

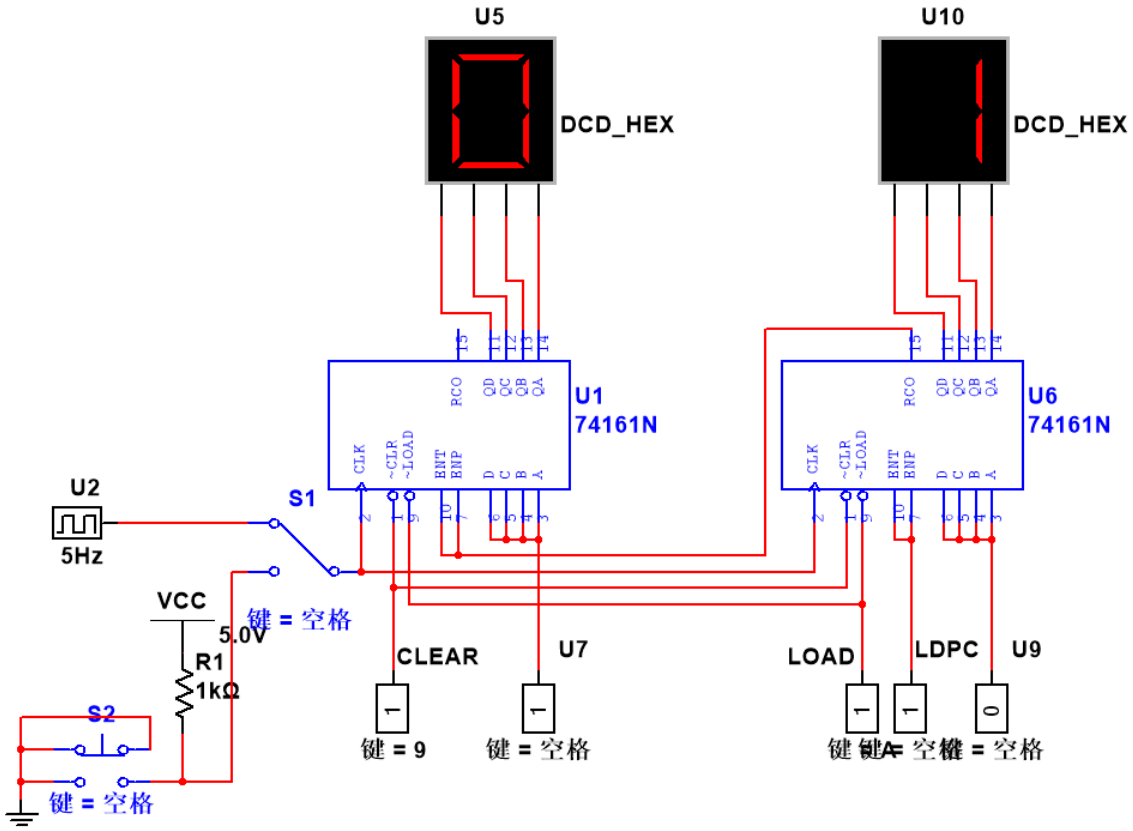
地址	十六进制表示	高五位	S_3-S_0	A字段	B字段	C字段	uA_5-uA_0
00H	00 00 01	00000	0000	000	000	000	000001
			直通	NOP	NOP	NOP	

执行过程：直接转到地址为01H的微指令

地址	十六进制表示	高五位	S_3-S_0	A字段	B字段	C字段	uA_5-uA_0
01H	00 6D 43	00000	0000	110	110	101	000011
			直通	LDAR	PC_B	LDPC	

执行过程：打开PC输出控制三态门，将PC中的地址打到AR，然后PC+1，然后转到地址为03H的微指令

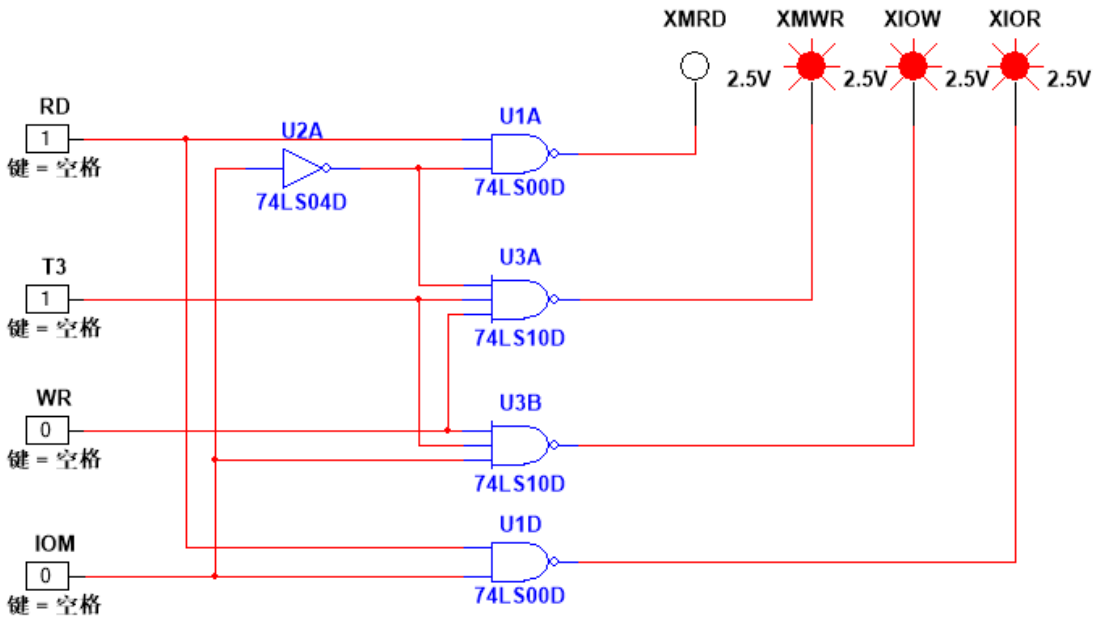
仿真模拟：



地址	十六进制表示	高五位	S_3-S_0	A字段	B字段	C字段	uA_5-uA_0
03H	10 70 70	00010	0000	111	000	001	110000
		RD	直通	LDIR	NOP	P<1>	

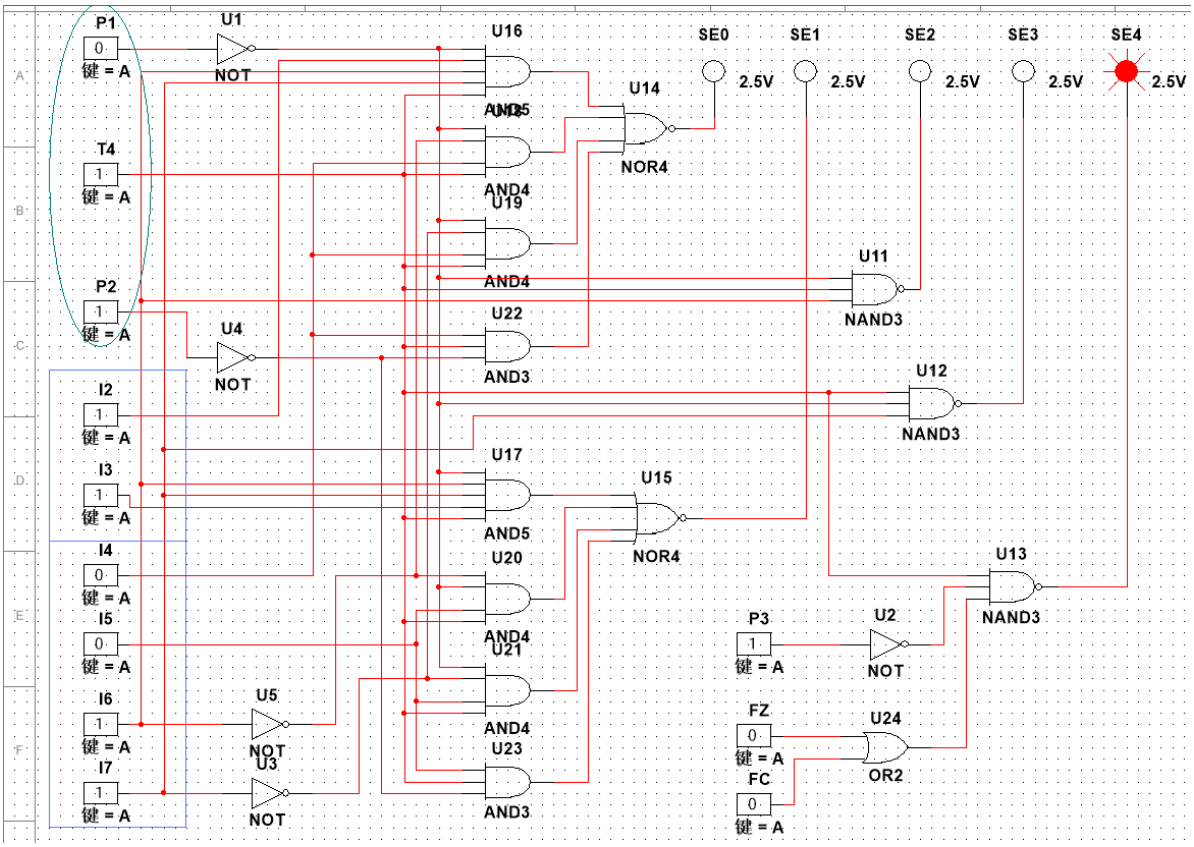
执行过程：MEM读取控制信号RD有效，将AR地址对应的指令打入IR，P<1>有效，下条微指令将根据指令操作码产生

读取仿真：



从MEM中读取指令

指令译码仿真：



由仿真结果可知，SE_4-SE_0=10000，微地址中的uA_3-uA_0的触发器的置位端PR有效，uA_3-uA_0被置为1，所以最后得到的微地址为111111（3FH），指令译码完成

3.3.2 执行指令

（1）操作数准备阶段

地址	十六进制表示	高五位	S_3-S_0	A字段	B字段	C字段	uA_5-uA_0
3FH	00 6D 6C	00000	0000	110	110	101	101100
			直通	LDAR	PC_B	LDPC	

执行过程：打开PC输出控制三态门，将PC中的地址打到AR，然后PC+1，然后转到地址为2CH的微指令

仿真模拟：略

地址	十六进制表示	高五位	S_3-S_0	A字段	B字段	C字段	uA_5-uA_0
2CH	10 10 2D	00010	0000	001	000	000	101101
		RD	直通	LDA	NOP	NOP	

执行过程：MEM读取控制信号RD有效，将AR地址对应的指令打入A，然后转到地址为2DH的微指令

地址	十六进制表示	高五位	S_3-S_0	A字段	B字段	C字段	uA_5-uA_0
2DH	00 2C 2E	00000	0000	010	110	000	101110
			直通	LDB	PC_B	NOP	

执行过程：打开PC输出控制三态门，将PC中的地址打到B，然后转到地址为2EH的微指令

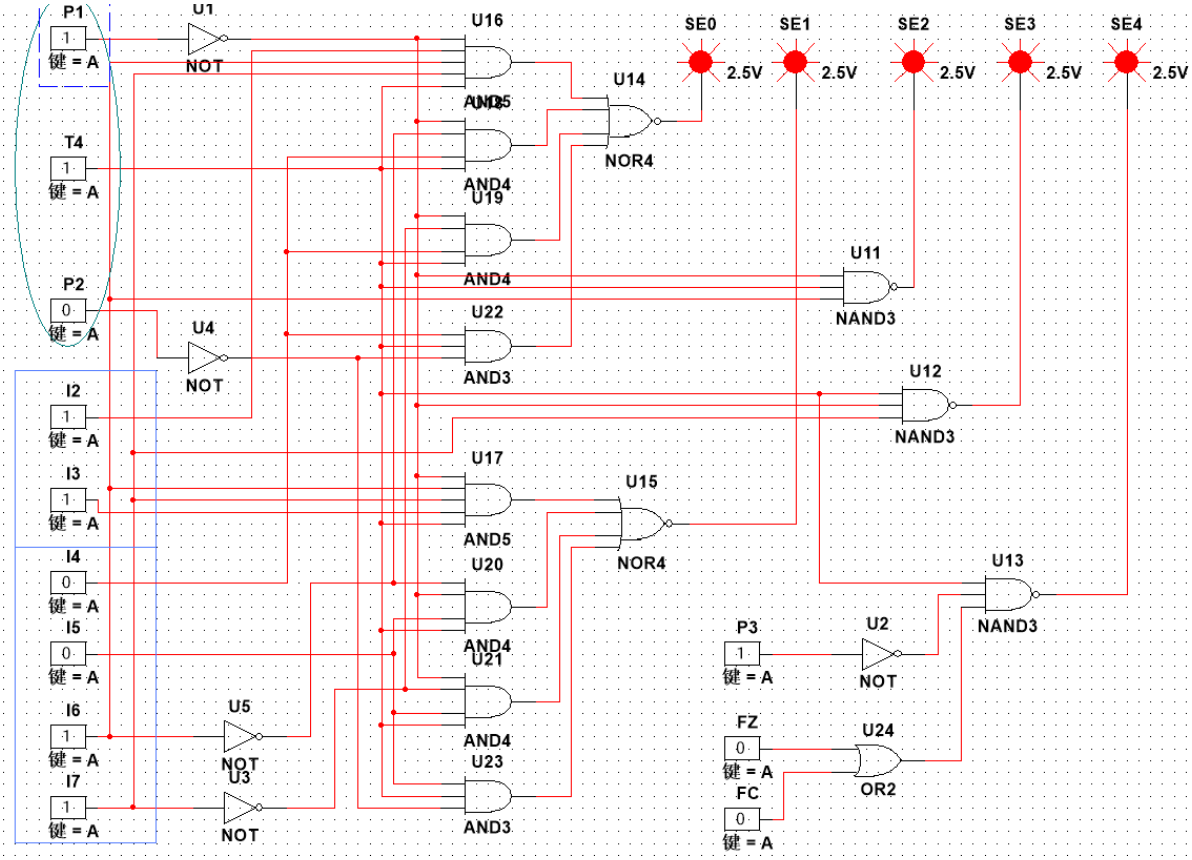
地址	十六进制表示	高五位	S_3-S_0	A字段	B字段	C字段	uA_5-uA_0
2EH	04 E2 2F	00000	1001	110	001	000	101111
			F=A+B	LDAR	ALU_B	NOP	

执行过程：打开ALU输出控制三态门，执行算术运算F=A+B，将运算结果打到AR，然后转到地址为2FH的微指令

地址	十六进制表示	高五位	S_3-S_0	A字段	B字段	C字段	uA_5-uA_0
2FH	04 92 8C	00000	1001	001	001	010	001100
			F=A+B	LDA	ALU_B	P<2>	

执行过程：打开ALU输出控制三态门，执行算术运算F=A+B，将运算结果打到A；P<2>有效，下条微指令将根据指令操作码产生

指令译码仿真：



由仿真结果可知，SE_4-SE_0=11111，微地址中的uA_5-uA_0的触发器的置位端PR均无效，所以无需修改微地址，最后得到的微地址为001100（0CH），指令译码完成

操作数准备完成。

（2）执行动作

地址	十六进制表示	高五位	S_3-S_0	A字段	B字段	C字段	uA_5-uA_0
0CH	10 30 01	00010	0000	011	000	000	000001
				LDRi	NOP	NOP	

执行过程：MEM读取控制信号RD有效，将AR地址对应的数据打入RD，然后转到地址为01H的微指令（公操作）

相对寻址方式的LAD指令完成

四、新增指令

4.1 将OR改写为CNT指令，用于计数

按直接寻址的方式将操作数读入并打到A，执行A+1->A，将A的数据存回原位

CNT 为双字节指令，采用直接寻址

OP	RD	RS	D
1001	**	**	直接地址

微程序

地址	高五位	S_3-S_0	A字段	B字段	C字段	uA_5-uA_0	说明
39H	00000	0000	110	110	101	011100	PC->AR;PC+1
			LDAR	PC_B	LDPC	1CH	暂时保持不变
1CH	00010	0000	001	000	000	011001	MEM->A
	RD		LDA	NOP	NOP	19H	读入数据
19H	00000	1101	001	001	000	011010	A+1->A
			LDA	ALU_B	NOP	1AH	数据+1
1AH	00100	0000	000	001	000	000001	A->MEM
	WR	直通A	NOP	ALU_B	NOP	01H	写回数据到原址

4.2 将HLT改为LR，用于左移

将RD的操作数不带进位地左移一位，存回RD

LR 为单字节指令

OP	RD	RS
0101	RD	RS

微程序

地址	高五位	S_3-S_0	A字段	B字段	C字段	uA_5-uA_0	说明
35H	00000	0000	001	011	000	100001	RD->A
			LDA	RD_B	NOP	21H	
21H	00000	0000	010	010	000	100010	RS->B
			LDB	RS_B	NOP	22H	
22H	00000	0111	011	001	000	000001	RD左移一位->RD
	CN=0	左移	LDRi	ALU_B	NOP	01H	

五、程序设计

5.1 求平均数

求多个数的加权平均数，每个数有8位，取低四位为数值，高四位为权值，存放地址为60H-68H

基本流程:

装入和初值

装入计数初值

LOOP:读入第一个数

求低四位

判断是否为0,为零则跳出

否则和加上该数

计数值+1（CNT）

JMP LOOP

R0 和 R1 辅助数（0FH） R2 数据地址 R3 读入数据

50H 存放计数值

除法运算

R0 和 R1 辅助数（0FH） R2 计数值 R3 被除次数

循环相减，计数

被除数为0则跳出

输出

LDI R0 00H 00H 0110 0000

01H 0000 0000

LDI R1 03H 02H 0110 0001

03H 0000 0011

ADD R0 R1 04H 0000 0100

OUT 40H R0 05H 0011 0000

06H 0100 0000

汇编程序

地址	内容	助记符	说明
00H	0110 0000	**START**:LDI R0,00H	装入和初值00H到R0
01H	0000 0000		
02H	0110 0001	LDI R1,00H	装入计数初值00H到R1

地址	内容	助记符	说明
03H	0000 0000		
04H	1101 00 01	STA 00,[50H],01	计数初值写入MEM
05H	0101 0000		
06H	0110 0001	LDI R1,0FH	装入0FH,用于取低四位
07H	0000 1111		
08H	0110 0010	LDI R2,60H	读入数据始地址
09H	0110 0000		
0AH	1100 1011	**LOOP**: LAD R3,[RI],00H	从MEM 读入数据送R3
0BH	0000 0000		变址寻址，偏移量为00H
0CH	0001 0111	AND R3 R1	得到R3低四位
0DH	1111 0000	BZC RESULT	后四位为0，跳转14H
0EH	0001 0100		
0FH	0000 1100	ADD R0 R3	累加求和
10H	1001 0000	CNT 00,00,50H	计数值+1
11H	0101 0000		
12H	0111 0010	INC RI	变址寄存加1，指向下一数据
13H	1110 0000	JMP **LOOP**	继续循环
14H	0000 1010		
15H	1100 0010	LAD R2 00 [50H]	从50H读入计数值到R2
16H	0101 0000		
17H	0110 0011	LDI R3, 00H	被除次数置为00H，存入R3
18H	0000 0000		
19H	1000 1100	**LOOP**:SUB R0 R2	循环相减
1AH	1111 0000	BZC RESULT	FZ或FC有效则跳出(1FH)
1BH	0001 1111		
1CH	0111 0011	INC R3	被除次数+1
1DH	1110 0000	JMP **LOOP**	继续循环
1EH	0001 1000		
1FH	0011 1100	OUT 40H R3	

地址	内容	助记符	说明
20H	0100 0000		
21H	1110 0000	JMP **START**	跳转至START
22H	0000 0000		
60H	0000 0010	数据	
61H	0000 0011		33/7=5
62H	0000 0110		
63H	0000 1010		
64H	0000 0110		
65H	0000 0101		
66H	0000 0010		
67H	0000 0000		

5.2 求加权平均数

求多个数的加权平均数，每个数有8位，取低四位为数值，高四位为权值，存放地址为60H-68H

基本流程:

装入和初值

装入计数初值

LOOP:读入第一个数

求低四位

判断是否为0,为零则跳出

求高四位

判断是否为0，为0则跳出

否则权值累加

开始乘法

每次看乘数末位是否为1

是则积+被乘数

然后乘数右移，被乘数左移

JMP LOOP

R0 乘数1(H)　R1 辅助数（0FH）　R2 数据地址　R3 乘数2（L）

50H 存放和值

51H 存放权值和

52H 存放数据地址

除法运算

R0 和值　R1 辅助数（0FH）　R2 权值和　R3 被除次数

循环相减，计数

被除数为0则跳出

输出

汇编程序

地址	内容	助记符	说明
00H	0110 0000	LDI R0,00H	装入和初值00H到R0
01H	0000 0000		
02H	0110 0001	LDI R1,00H	装入计数初值00H到R1
03H	0000 0000		
04H	1101 0000	STA 00,[50H],R0	和值初值写入MEM
05H	0101 0000		
06H	1101 0001	STA 00,[51H],R1	权值和初值写入MEM
07H	0101 0001		
08H	0110 0010	LDI R2,60H	读入数据始地址
09H	0110 0000		
0AH	1101 0010	STA 00,[52H],R2	将数据地址写入52H
0BH	0101 0010		
0CH	0110 0001	**LOOP**:LDI R1,0FH	装入0FH,用于取低四位
0DH	0000 1111		
0EH	1100 0111	LAD 01 R3,52H	从MEM 读入数据送R3
0FH	0101 0010		间接寻址，数据地址在52H
10H	1001 0000	CNT 00,00,52H	数据地址+1

地址	内容	助记符	说明
11H	0101 0010		
12H	0100 1100	MOV R0 R3	复制一份R3到R0
13H	0001 0111	AND R3 R1	得到R3低四位
14H	1111 0000	BZC RESULT	低四位为0则转移(3BH)
15H	0011 1011		
16H	0110 0001	LDI R1,F0H	装入F0H,用于取高四位
17H	1111 0000		
18H	0001 0100	AND R0 R1	得到R3高四位
19H	1111 0000	BZC RESULT	高四位为0则转移(38H)
1AH	0011 1000		
1BH	0110 0001	LDI R1,01H	右移值为两位
1CH	0000 0010		
1DH	1010 0100	RR R0 R1	右移4次,得到高四位值
1EH	1010 0100	RR R0 R1	
1FH	1100 0001	LAD 00 [51H] R1	读取权值和存于R1
20H	0101 0001		
21H	0000 0001	ADD R1 R0	权值累加
22H	1101 0001	STA 00 [51H] R1	写回权值和
23H	0101 0001		
24H	0110 0001	LDI R1 01H	R1判断乘数（[R3]）末尾是否为1
25H	0000 0001		
26H	0110 0010	LDI R2 00H	R2用来累加乘法结果
27H	0000 0000		
28H	0001 1101	**LOOP2**:AND R1 R3	如果[R1]末尾为1，则R2加上R0
29H	1111 0000	BZC RESULT	为0则不相加(2CH)
2AH	0010 1100		
2BH	0000 0010	ADD R2 R0	累加
2CH	1010 0111	RR R3(R1)	R3右移
2DH	1111 0000	BZC BZC RESULT	移位后为0则跳出(34H)

地址	内容	助记符	说明
2EH	0011 0100		
2FH	0101 0100	LR R0(R1)	左移R0
30H	0110 0001	LDI R1 01H	重新把R1赋值为01H
31H	0000 0001		
32H	1110 0000	JMP **LOOP2**	继续循环(28H)
33H	0010 1000		
34H	1100 0001	LAD R1 00 [50H]	取之前乘法累加和
35H	0101 0000		
36H	0000 1001	ADD R1 R2	乘法结果累加
37H	1101 0001	STA 00 [50H] R1	写回
38H	0101 0000		
39H	1110 0000	JMP **LOOP**	继续循环（0CH）
3AH	0000 1100		
3BH	1100 0001	LAD R1 00 [50H]	从50H读入和值到R1
3CH	0101 0000		
3DH	1100 0010	LAD R2 00 [51H]	从51H读入权值和到R2
3EH	0101 0001		
3FH	0110 0011	LDI R3, 00H	被除次数置为00H，存入R3
40H	0000 0000		
41H	1000 1001	**LOOP**:SUB R1 R2	循环相减
42H	1111 0000	BZC RESULT	FZ或FC有效则跳出(47H)
43H	0100 0111		
44H	0111 0011	INC R3	被除次数+1
45H	1110 0000	JMP **LOOP**	继续循环(41H)
46H	0001 1000		
47H	0011 1100	OUT 40H R3	输出
48H	0100 0000		
60H	0010 0010	数据	

地址	内容	助记符	说明
61H	0001 0011		
62H	0100 0110		
63H	0000 1010		
64H	0011 0110		
65H	0100 0101		
66H	0001 0010		
67H	1000 0100		
68H	0000 0000		