第5章

8位模型计算机原理与设计

5.1 8位模型CPU结构

1. 运算部件
2. 寄存器组
3. 指令寄存器
4. 程序计数器
5. 地址寄存器
6. 标志寄存器
7. 微命令产生部件
8. 时序系统

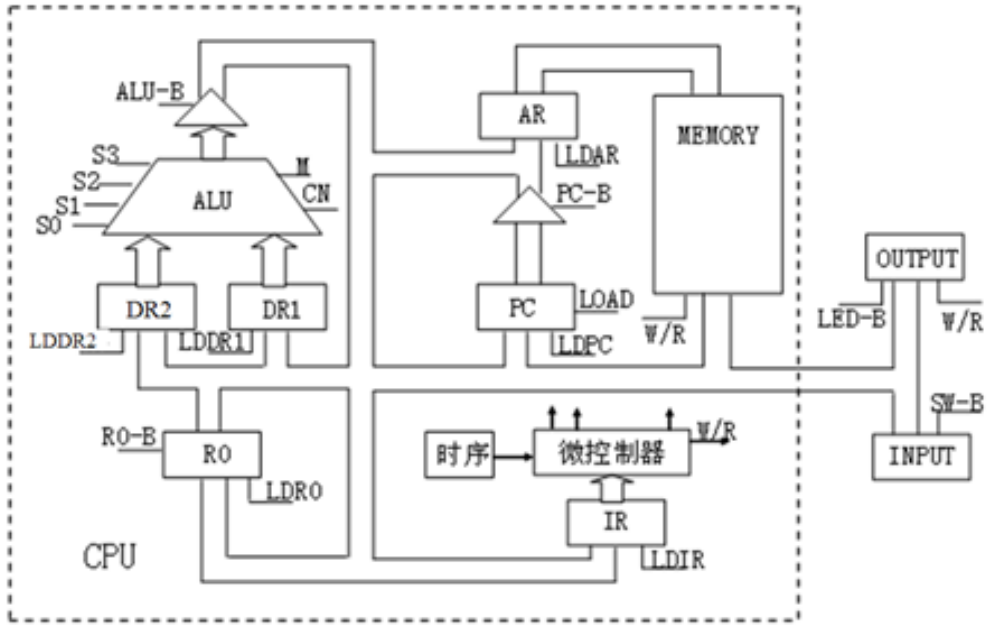


图 5-1　8 位模型 CPU 的结构

5.2 指令系统结构及其功能的确定

5.2.1. 模型机指令系统

表 5-1 指令的基本格式

位	7 6 5 4	3 2	1 0
功能	OP-CODE	rs	rd

表 5-2 寄存器操作数

rs 或 rd	选定的寄存器
00	R0
01	R1
10	R2

表 5-3 模型机指令系统及其指令编码形式

指令助记符	机器指令码	Addr 地址码	功能说明
IN	80 H		"INPUT"中的数据→R0
ADD addr	90 H	XX H	R0+[addr] ->R0
STA addr	A0 H	XX H	R0 -> [addr]
OUT addr	B0 H	XX H	[addr] -> BUS
JMP addr	C0 H	XX H	addr →PC

5.1 8位模型CPU结构

5.2.2. 拟定指令流程和微命令序列

1. 微程序控制概念

（1）微命令和微操作。

（2）微指令、微地址。

（3）微周期。

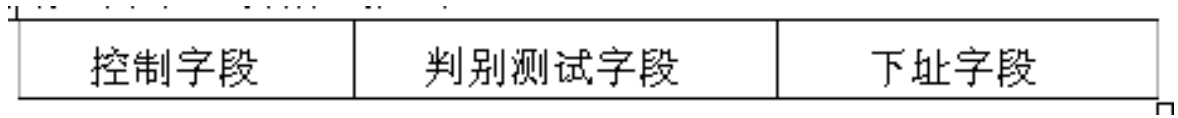
（4）微程序。

5.1 8位模型CPU结构

5.2.2. 拟定指令流程和微命令序列

2. 微指令格式

（1）水平型微指令。



（2）垂直型微指令。

（3）水平型微指令与垂直型微指令的比较

5.1 8位模型CPU结构

5.2.2. 拟定指令流程和微命令序列

3. 模型机的微指令

表 5-4 24 位微代码定义

24	23	22	21	20	19	18	17	16	15 14 13	12 11 10	9 8 7	6	5	4	3	2	1
S3	S2	S1	S0	M	Cn	WE	A9	A8	A	B	C	uA5	uA4	uA3	uA2	uA1	uA0
操作控制信号									译码器	译码器	译码器	下地址字段					

表 5-5 A、B、C 各字段功能说明

A 字段				B 字段				C 字段			
15	14	13	选择	12	11	10	选择	9	8	7	选择
0	0	0		0	0	0		0	0	0	
0	0	1	LDRi	0	0	1	RS-B	0	0	1	P (1)
0	1	0	LDDR1	0	1	0	RD_B	0	1	0	P (2)
0	1	1	LDDR2	0	1	1	RJ_B	0	1	1	P (3)
1	0	0	LDIR	1	0	0	SFT_B	1	0	0	P (4)
1	0	1	LOAD	1	0	1	ALU-B	1	0	1	LDAR
1	1	0	LDAR	1	1	0	PC-B	1	1	0	LDPC

5.1 8位模型CPU结构

5.2.2. 拟定指令流程和微命令序列

4. 微指令的执行方式

5. 时序安排

6. 拟定指令流程和微命令序列

7. 形成控制逻辑

5.1 8位模型CPU结构

5.2.3 微程序设计方法

1. **IN指令**
2. **ADD指令**
3. **STA指令**
4. **OUT指令**
5. **JMP指令**

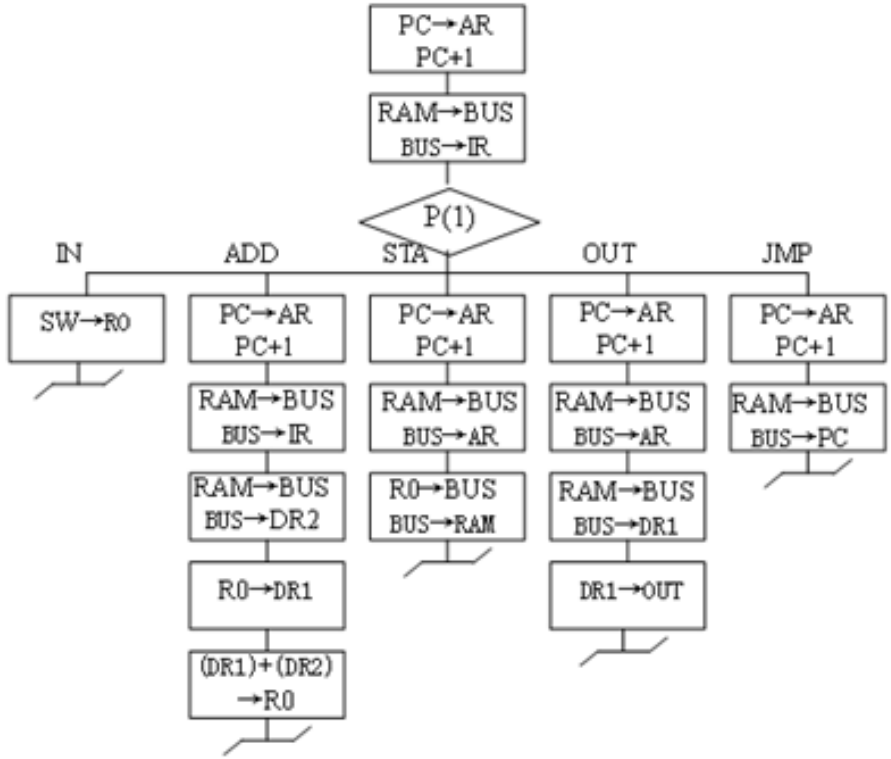


图 5-2　微程序流程图

5.3 CPU硬件系统设计

5.3.1 CPU顶层设计

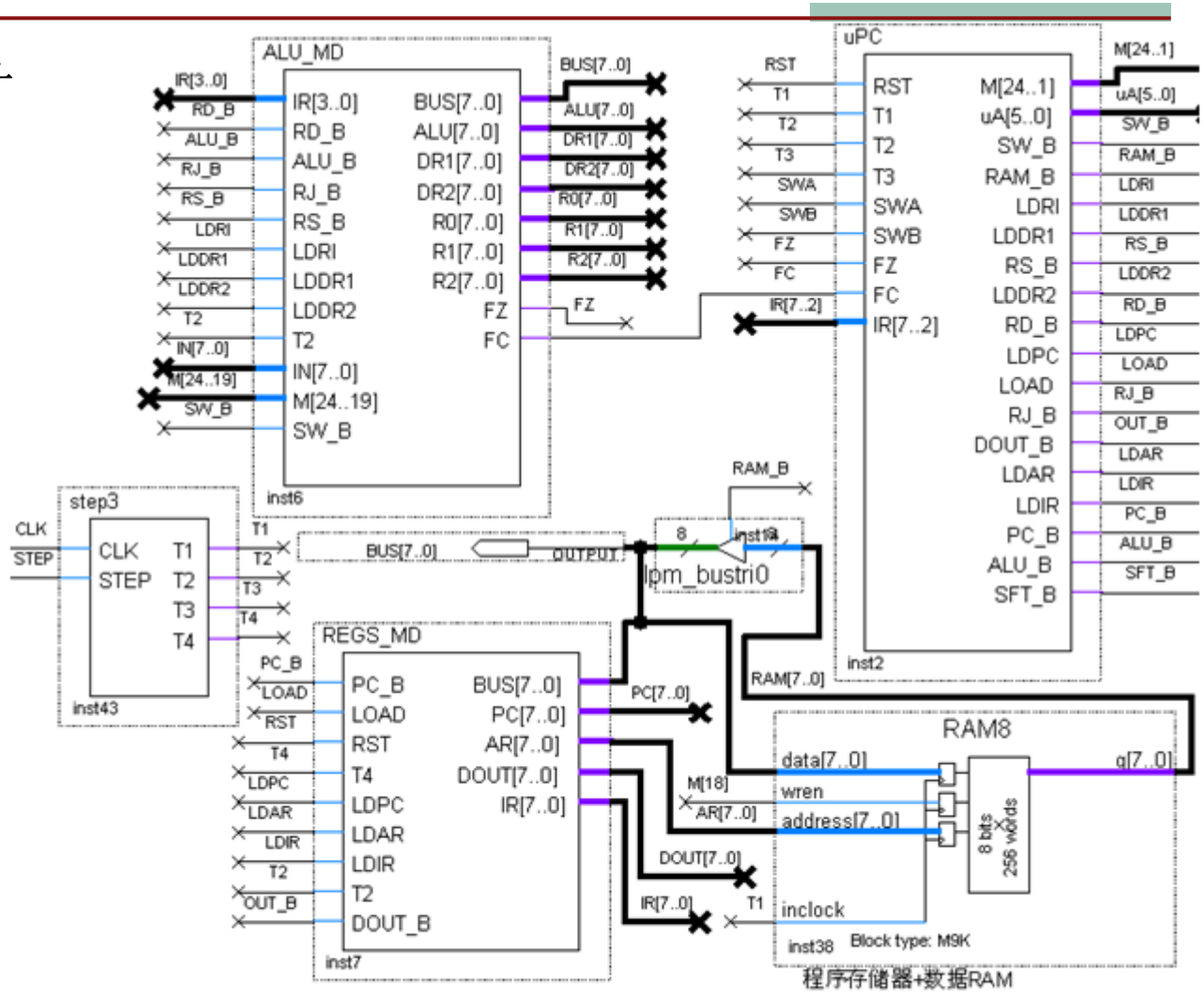


图 5-3　8 位模型机 CPU 的顶层设计电路原理图

5.3 CPU硬件系统设计

5.3.1 CPU顶层设计

1、主要功能模块电路结构介绍

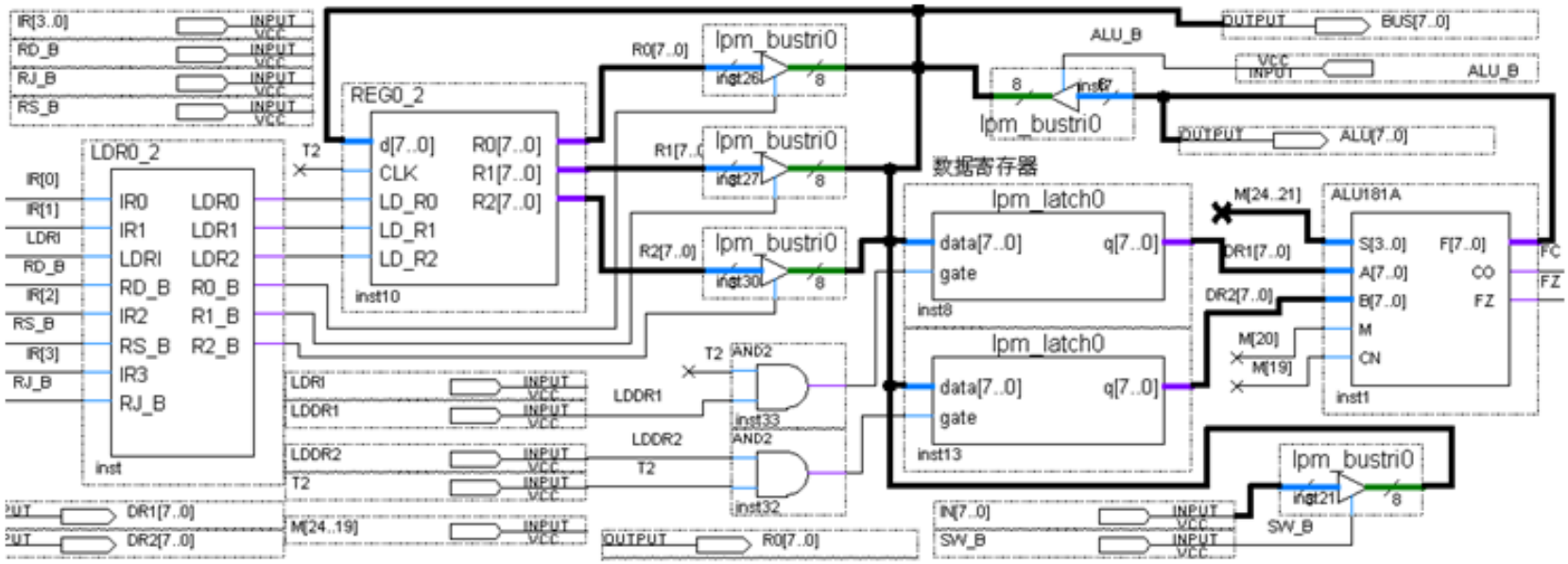


图 5-4　图 5-3 的顶层设计中模块 ALU_MD 内部的电路原理图

5.3 CPU硬件系统设计

5.3.1 CPU顶层设计

1、主要功能模块电路结构介绍

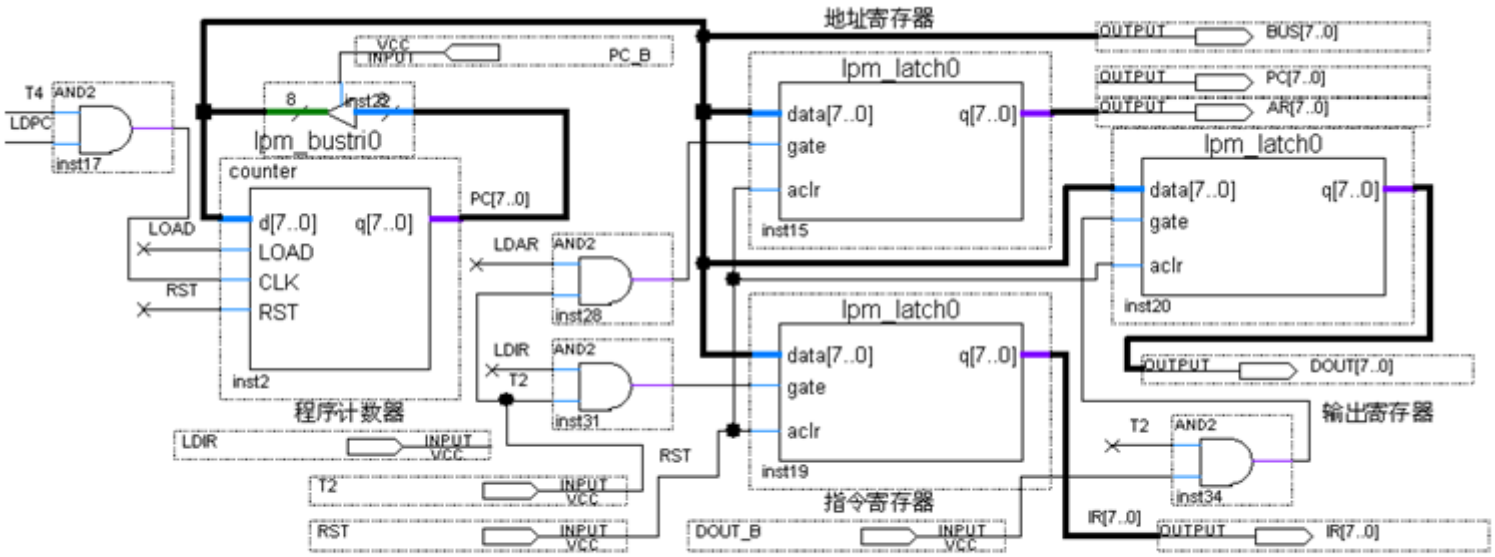


图 5-5　图 5-3 的顶层设计中模块 REGS_MD 内部的电路原理图

5.3 CPU硬件系统设计

5.3.1 CPU顶层设计

1、主要功能模块电路结构介绍

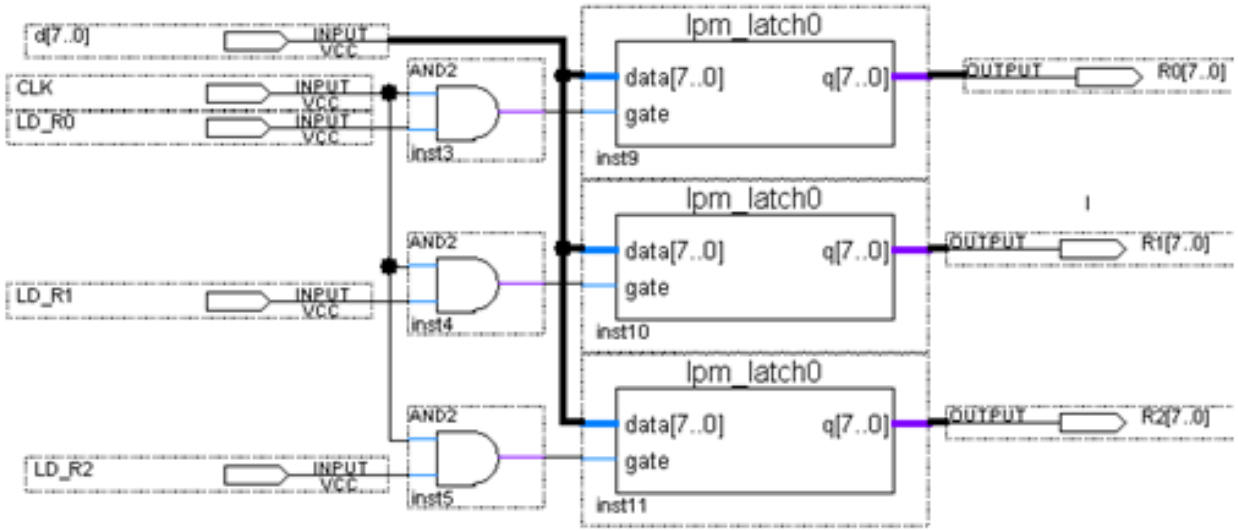


图 5-6　图 5-4 中 REG0_2 内部的电路图

5.3 CPU硬件系统设计

5.3.1 CPU顶层设计

2、REGS_MD模块时序性能仿真测试

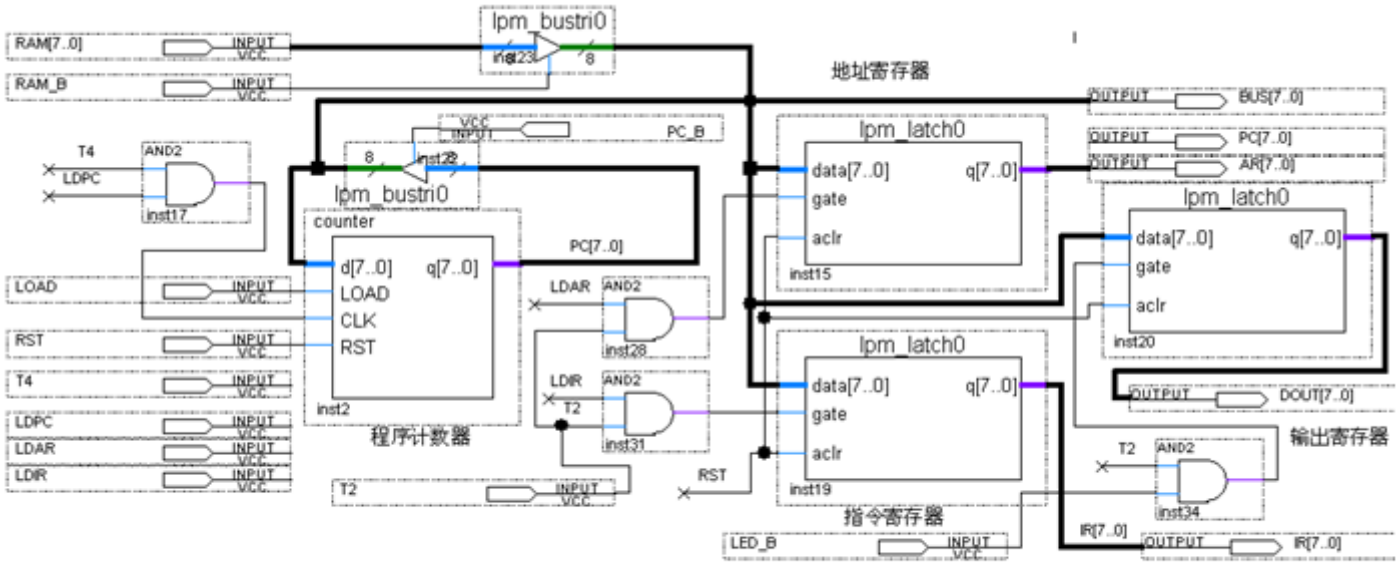


图 5-7　加了 RAM 数据线的 REGS_MD 模块的电路原理图

5.3 CPU硬件系统设计

5.3.1 CPU顶层设计

2、REGS_MD模块时序性能仿真测试

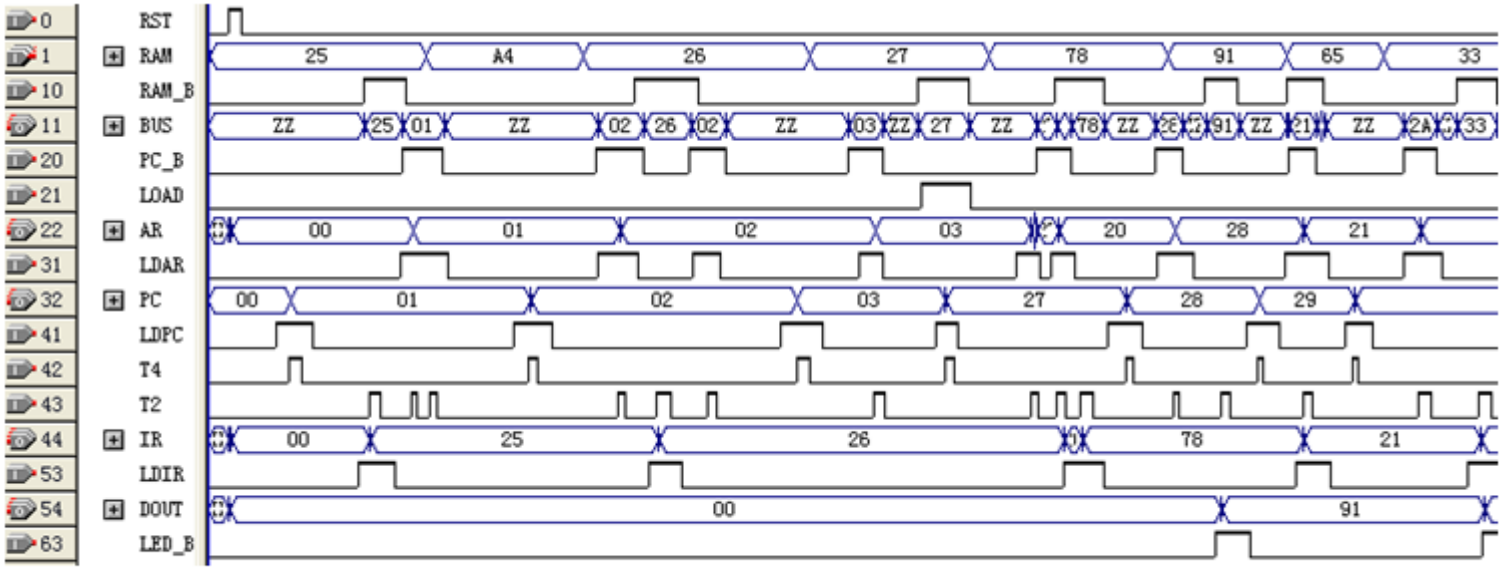


图 5-8　测试图 5-7 电路的时序仿真波形图

5.3 CPU硬件系统设计

5.3.1 CPU顶层设计

3、ALU_MD模块时序性能仿真测试

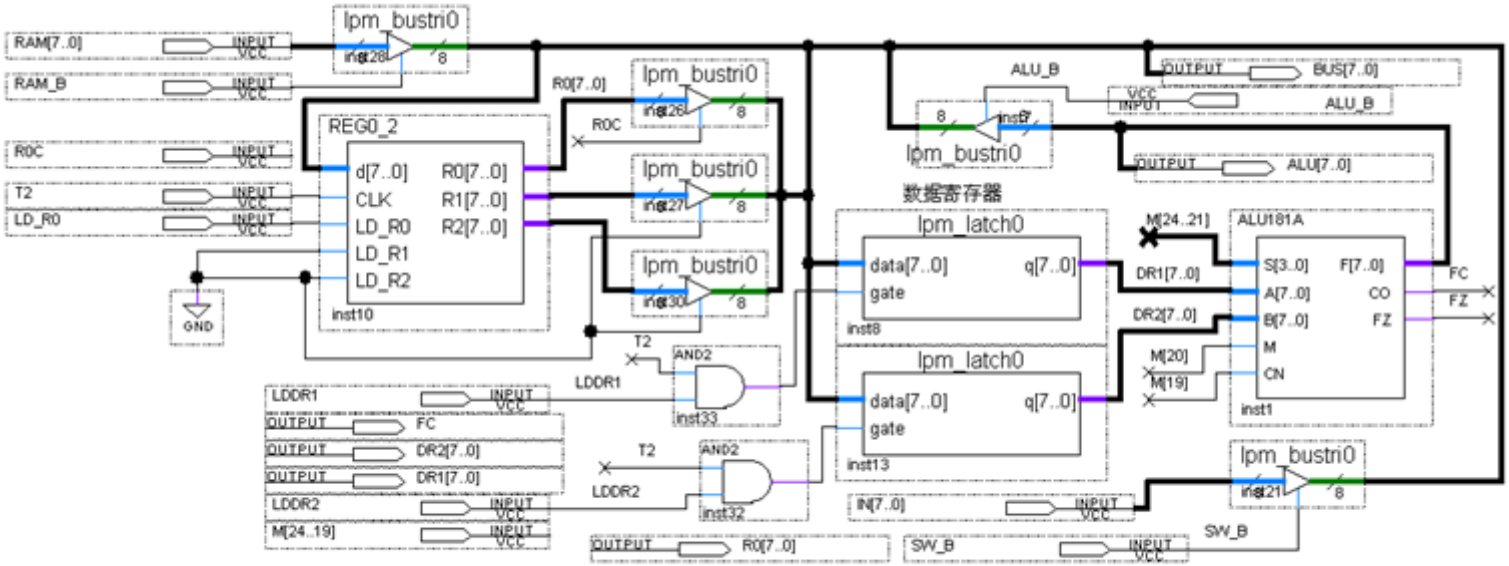


图 5-9　加了 RAM 数据线的 ALU_MD 模块的电路原理图

5.3 CPU硬件系统设计

5.3.1 CPU顶层设计

3、ALU_MD模块时序性能仿真测试

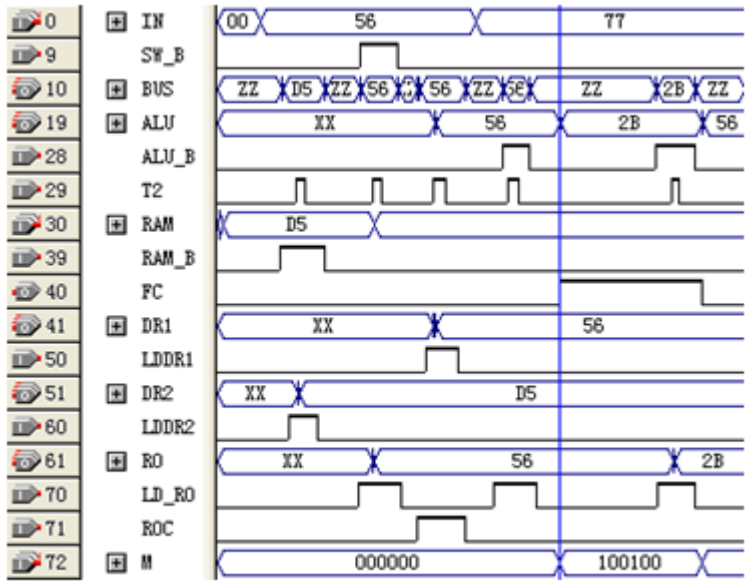


图 5-10　测试图 5-9 电路的时序仿真波形图

5.3 CPU硬件系统设计

5.3.1 CPU顶层设计

4、uPC模块时序性能仿真测试

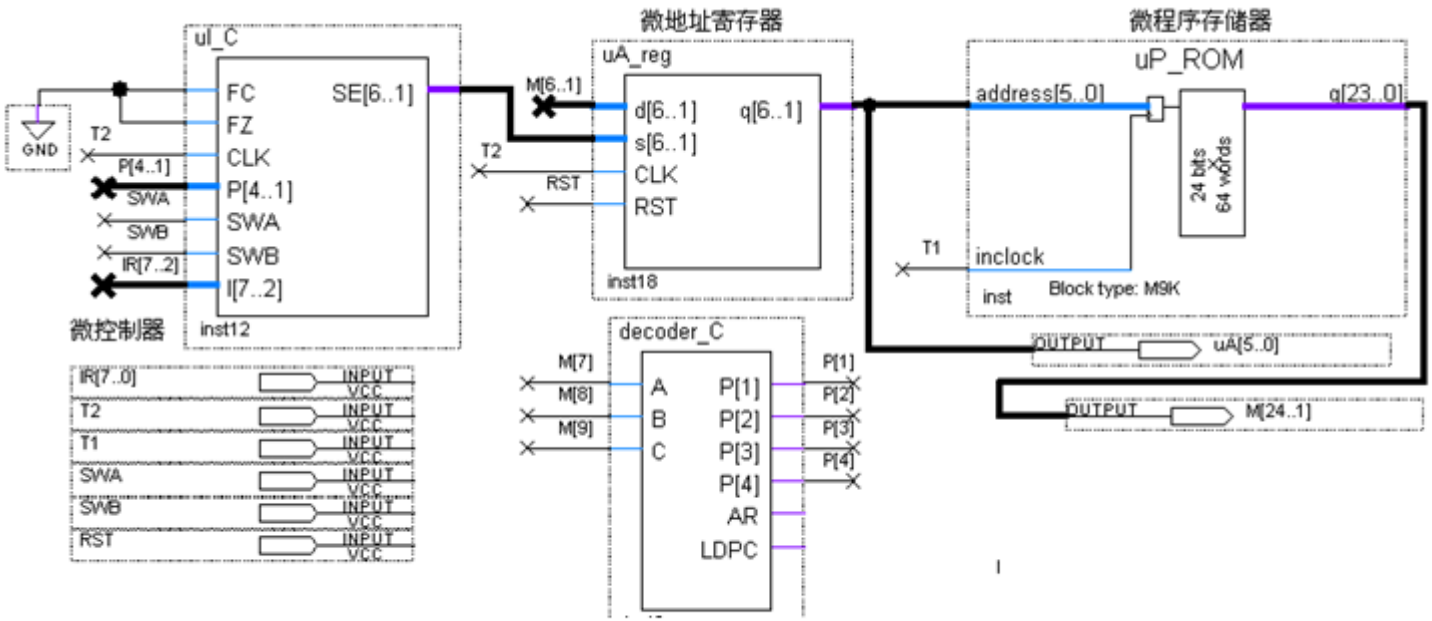


图 5-11　uPC 模块的测试电路图

5.3 CPU硬件系统设计

5.3.1 CPU顶层设计

4、uPC模块时序性能仿真测试

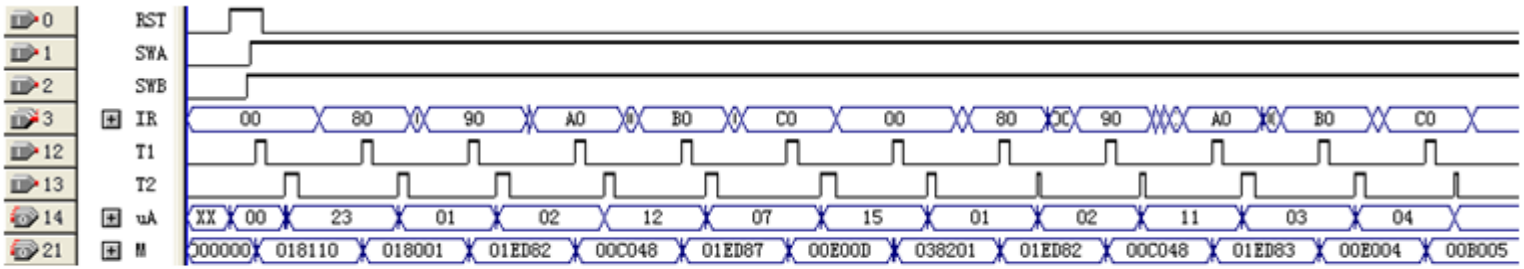


图 5-12　测试图 5-11 电路的时序仿真波形图

5.3 CPU硬件系统设计

5.3.2 取指令和指令译码

1. 取指令阶段
2. 分析取数阶段
3. 执行阶段

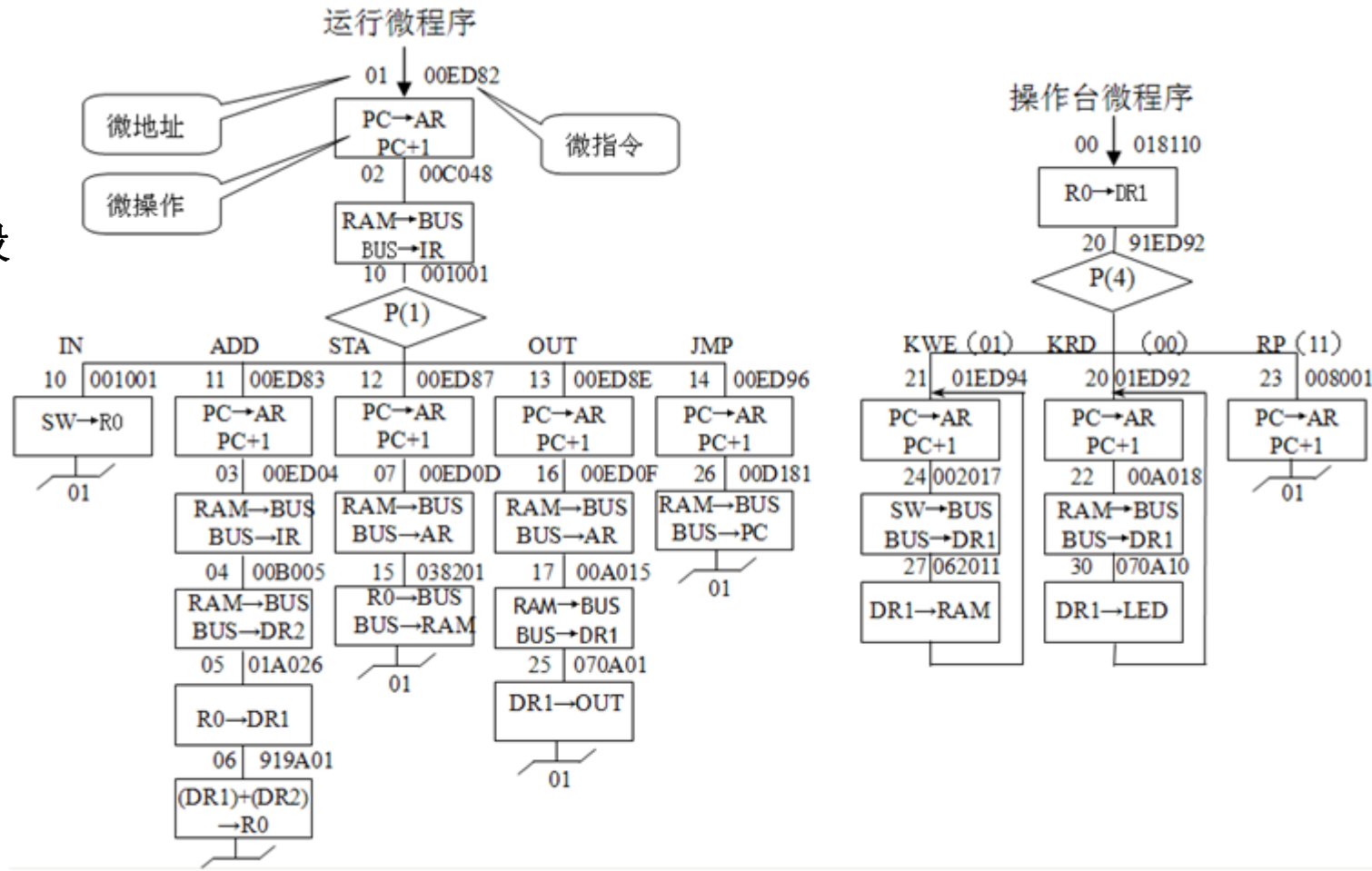


图 5-13　微程序及微指令流程图

5.3 CPU硬件系统设计

5.3.3 设计微代码表

表 5-6 微代码表

微地址	微指令	S3	S2	S1	S0	M	CN	WE	A9	A8	A	B	C	uA5-uA0
0 0	018110	0	0	0	0	0	0	0	1	1	000	000	100	010000
0 1	01ED82	0	0	0	0	0	0	0	1	1	110	110	110	000010
0 2	00C048	0	0	0	0	0	0	0	0	1	100	000	001	001000
0 3	00E004	0	0	0	0	0	0	0	0	1	110	000	000	000100
0 4	00B005	0	0	0	0	0	0	0	0	1	011	000	000	000101
0 5	01A206	0	0	0	0	0	0	0	1	1	010	001	000	000110
0 6	919A01	1	0	0	1	0	0	0	1	1	001	101	000	000001
0 7	00E00D	0	0	0	0	0	0	0	0	1	110	000	000	001101
1 0	001001	0	0	0	0	0	0	0	0	0	001	000	000	000001
1 1	01ED83	0	0	0	0	0	0	0	1	1	110	110	110	000011
1 2	01ED87	0	0	0	0	0	0	0	1	1	110	110	110	000111
1 3	01ED8E	0	0	0	0	0	0	0	1	1	110	110	110	001110
1 4	01ED96	0	0	0	0	0	0	0	1	1	110	110	110	010110
1 5	038201	0	0	0	0	0	0	0	1	1	000	001	000	000001
1 6	00E00F	0	0	0	0	0	0	0	0	1	110	000	000	001111
1 7	00A015	0	0	0	0	0	0	0	0	1	010	000	000	010101
2 0	01ED92	0	0	0	0	0	0	0	1	1	110	110	110	010010
2 1	01ED94	0	0	0	0	0	0	0	1	1	110	110	110	010100
2 2	01A010	0	0	0	0	0	0	0	1	1	010	000	000	010000
2 3	018001	0	0	0	0	0	0	0	1	1	000	000	000	000001
2 4	062011	0	0	0	0	0	1	1	0	0	010	000	000	010001
2 5	010A01	0	0	0	0	0	0	0	1	0	000	101	000	000001
2 6	00D181	0	0	0	0	0	0	0	0	1	101	000	110	000001
2 7	062011	0	0	0	0	0	1	1	0	0	010	000	000	010001
3 0	070A10	0	0	0	0	0	1	1	1	0	000	101	000	010000

5.3 CPU硬件系统设计

5.3.4 建立数据与控制通路

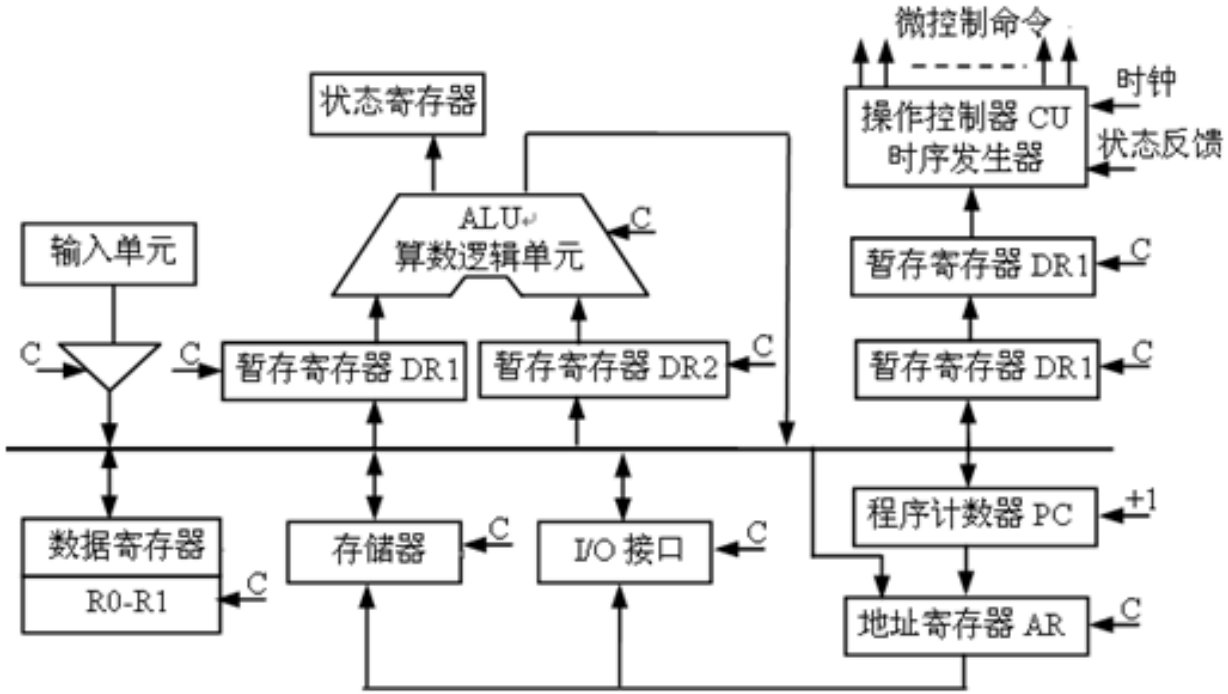


图 5-14 6 模型机 CPU 的数据通路框图

5.3 CPU硬件系统设计

5.3.5 控制执行单元

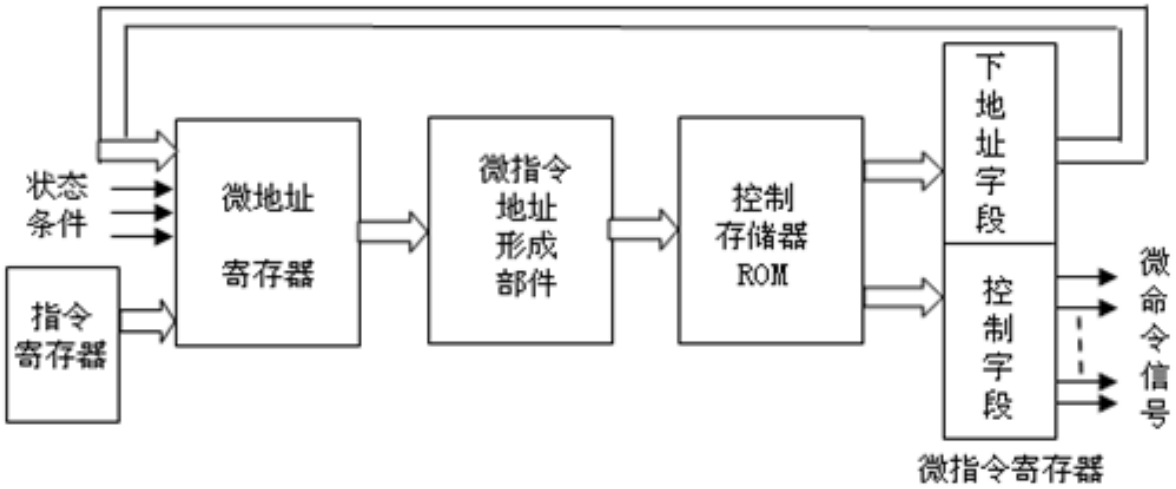


图 5-15 微程序控制的基本模块与控制流程

5.3 CPU硬件系统设计

5.3.6 在模型机中运行软件

1. 输入指令IN的执行过程

表 5-7 示例程序 模型机的指令及编码形式

地址	内容(16 进制)	助记符	说明
00H	80	IN R0	“INPUT” → R0，键盘输入数据
01H	90	ADD [0AH]	[R0]+[0AH] → R0，作加法后结果送 R0
02H	0A		
03H	A0	STA [0BH]	[R0] → [0BH]，将 R0 的内容送 RAM 的 0B 单元
04H	0B		
05H	B0	OUT [0BH]	[0BH] “OUTPUT”，显示输出数据
06H	0B		
07H	C0	JMP [08H]	[09H] → PC ，以[08H]内容为转移地址
08H	00		
09H	00		
0AH	D5	DB D5H	被加数（自定）
0BH	AC		此地址将放求和结果，但此时的数据是 AC

5.3 CPU硬件系统设计

5.3.6 在模型机中运行软件

1. 输入指令IN的执行过程

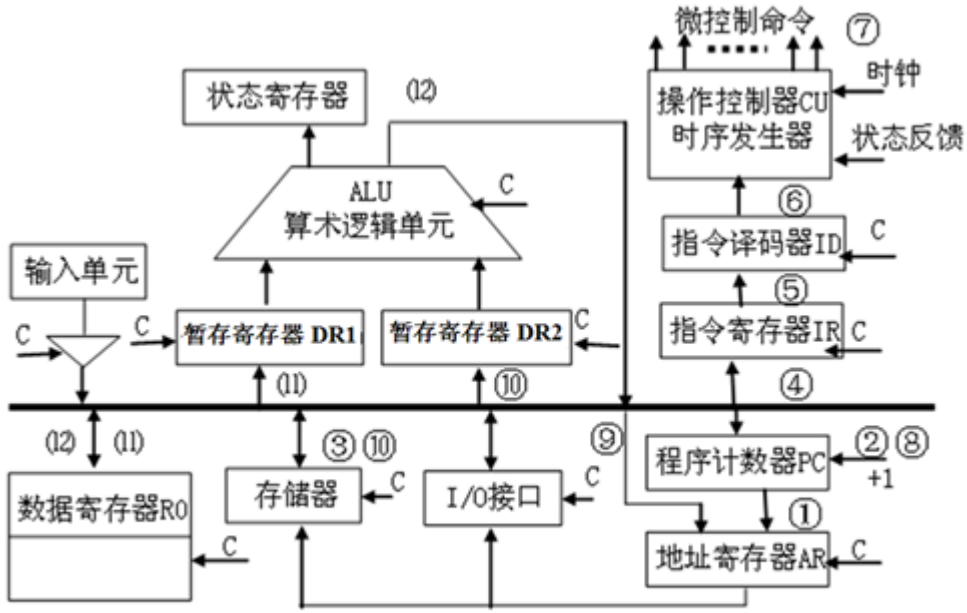


图 5-16 IN 指令的执行过程

2. 加法指令ADD的执行过程

5.3.7 模型机整机系统时序仿真

表 5-8　微指令执行情况

STEP	后续 uA 微地址	MC 微指令	PC	IR 指令	完成功能	执行结果
1	00	018110	00	00	控制台（读/写/运行）功能判断	控制台操作：转入程序运行方式
2	23	018001			SWB、WSA=（11）转 RP，分支转移	转程序执行方式
3	02	01ED82	01		**执行第 1 条指令**.（输入 IN）	PC－AR=00H,PC+1=01H,AR 指向指令地址
4	10	00C048		80	取指令,将 RAM 中的指令送指令寄存器	RAM(00H)=00→BUS→IR=00H
5	01	001001	02		接收来自 IN 输入的数据，送 R0 寄存器	R0=56H，键 1、键 2 输入数据 56H
6	02	01ED82			**执行第 2 条指令**,（加法 ADD）	PC－AR=01H,PC+1=02H,AR 指向指令地址
7	11	00C048		90	取指令	RAM(01H)=90H→BUS→IR=90H
8	03	01ED83	03		间接寻址，AR 指向取数的间接地址	PC→AR=02H,PC+1=03H，RAM=90H
9	04	00E004			取数地址送 AR	RAM(02)=0AH→BUS→AR=0AH
10	05	00B005			从 RAM 中取数送 DR2	RAM(0AH)=D5H→BUS→DR2=D5H
11	06	01A206			将 R0 的数据送 DR1	(R0)=56H→BUS→DR1=56H
12	01	919A01			加法运算：(DR1)+(DR2)→R0	56H+D5H=12BH→R0=2BH,进位 FC=1
13	02	01ED82	04		**执行第 3 条指令**（存储 STA）	PC→AR=03H，PC+1=04H
14	12	00C048		A0	取指令	RAM(03H)=20H→BUS→IR=A0H
15	07	01ED87	05		间接寻址，AR 指向存数的间接地址	PC→AR=04H,PC+1=05H
16	15	00E00D			存数的地址送 AR	RAM(04)=0BH→BUS→AR=0BH
17	01	038201			R0 的内容存入 RAM(0BH)单元	(R0)=2BH→BUS→RAM(0BH)=2BH,此时 RAM 输出 RAM 0B 单元的原数据 ACH
18	02	01ED82	06		**执行第 4 条指令**（输出 OUT）	PC→AR=05H，PC+1=06H
19	13	00C048		B0	取指令	RAM(05H)=B0H→BUS→IR=B0H
20	16	01ED8E	07		间接寻址，AR 指向取数的间接地址	PC→AR=06H，PC+1=07H
21	17	00E00F			取数地址送 AR	RAM(06)=0BH→BUS→AR=0BH
22	25	00A015			从 RAM 中取数送 DR1	RAM(0BH)=2BH→BUS→DR1=2BH
23	01	010A01			DR1 的内容送输出单元 DOUT	DR1=2BH→BUS→DOUT=2BH
24	02	01ED82	08		**执行第 5 条指令**（转移 JMP）	PC→AR=07H，PC+1=08H
25	14	00C048		C0	取指令	RAM(07H)=C0H→BUS→IR=C0H
26	26	01ED96	09	C0	间接寻址，AR 指向转移的间接地址	PC→AR=08H，PC+1=09H
27	01	00D181	00		转移地址送 PC，转到 00H。	RAM(08H)=00→BUS→PC=00H
28	02	01ED82	01		**执行第 1 条指令——程序循环**	PC→AR=00H，PC+1=01H
29	10	00C048	01	00	取指令	
…						

5.3 CPU硬件系统设计

5.3.7 模型机整机系统时序仿真

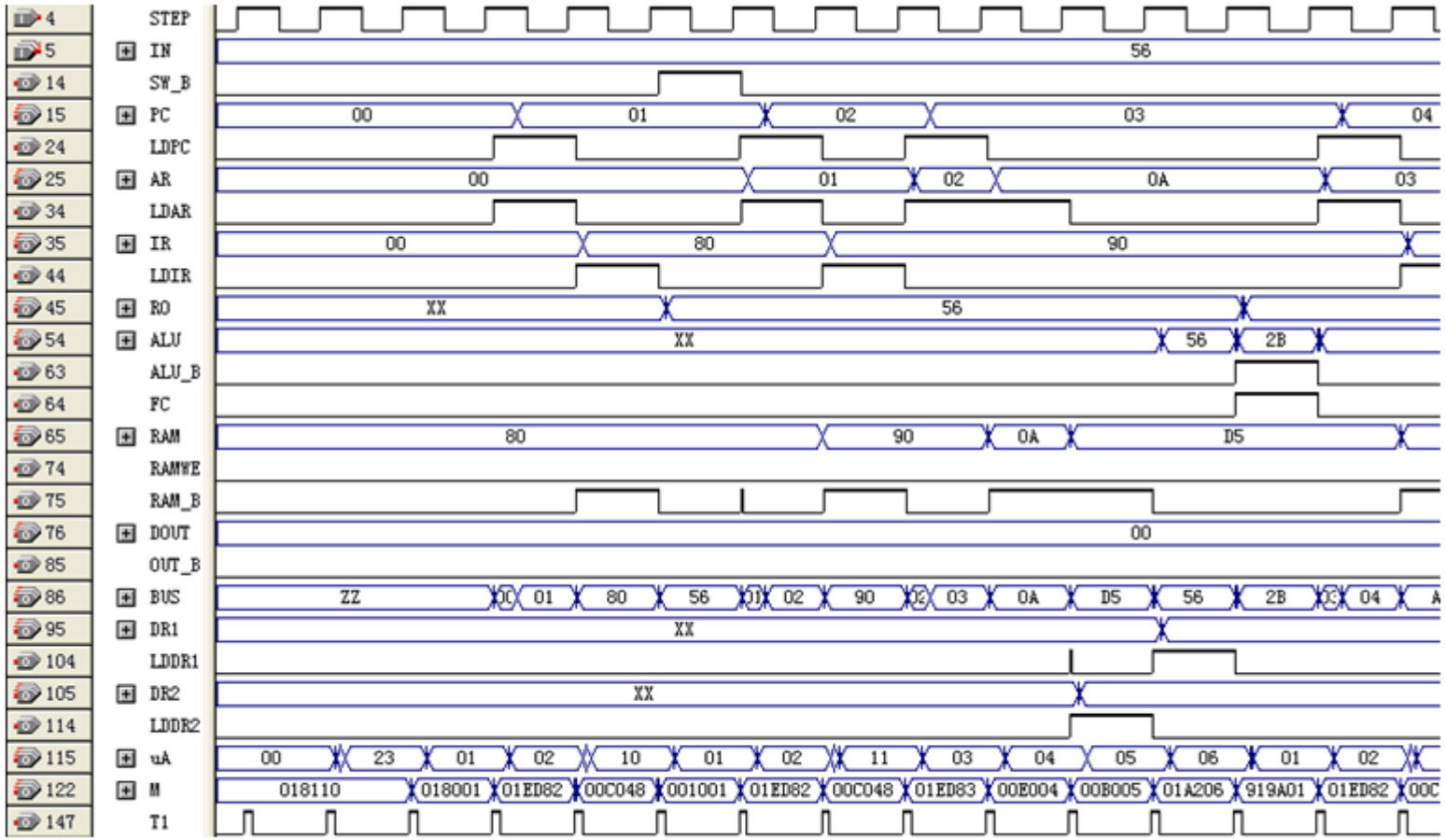


图 5-17(a) 模型机运行表 5-7 程序的整机仿真波形前半部

5.3 CPU硬件系统设计

5.3.7 模型机整机系统时序仿真

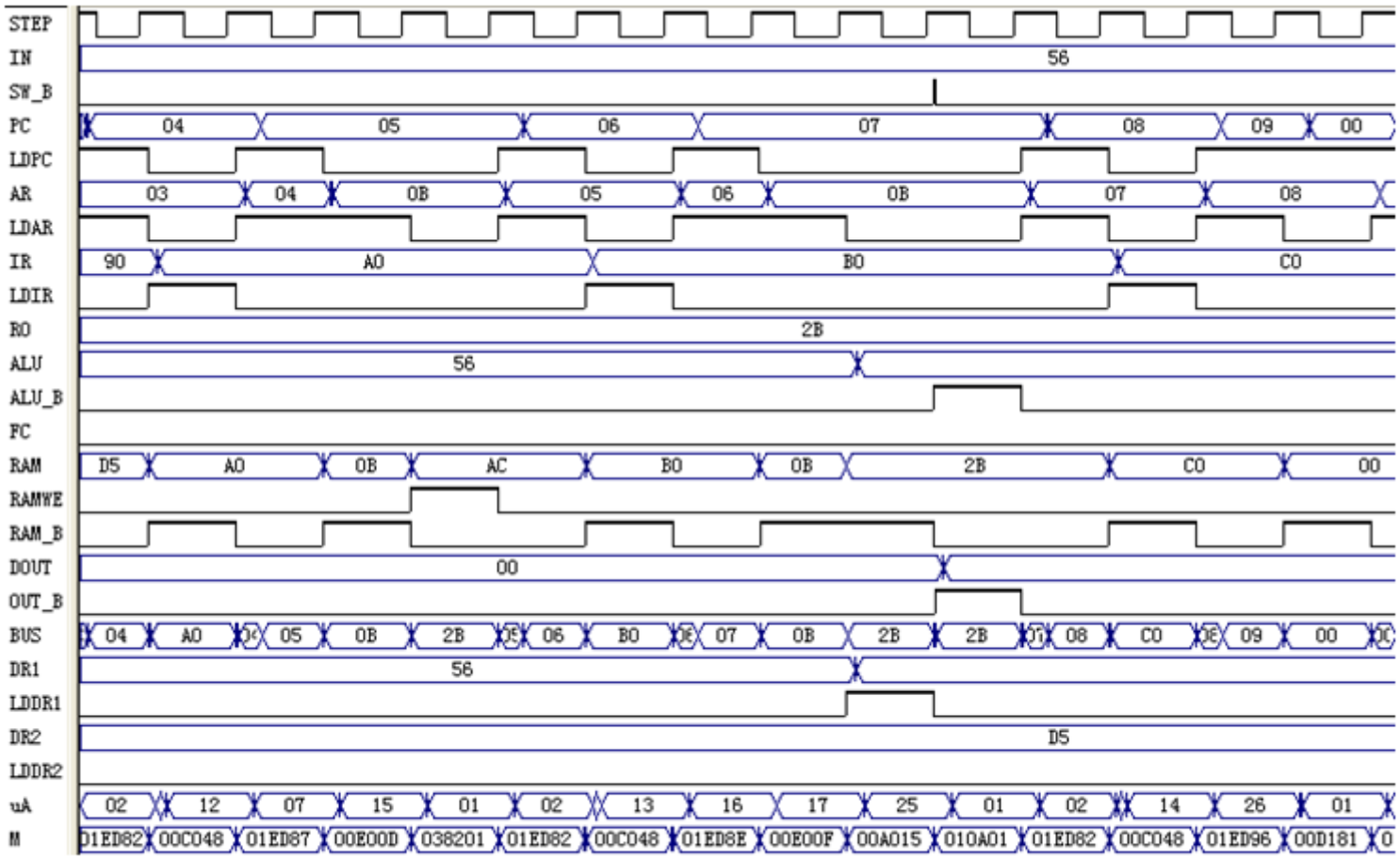


图 5-17(b) 模型机运行表 5-7 程序的整机仿真波形后半部

5.3 CPU硬件系统设计

5.3.8 模型机系统硬件功能测试

1、嵌入式逻辑分析仪SignalTap II测试与分析

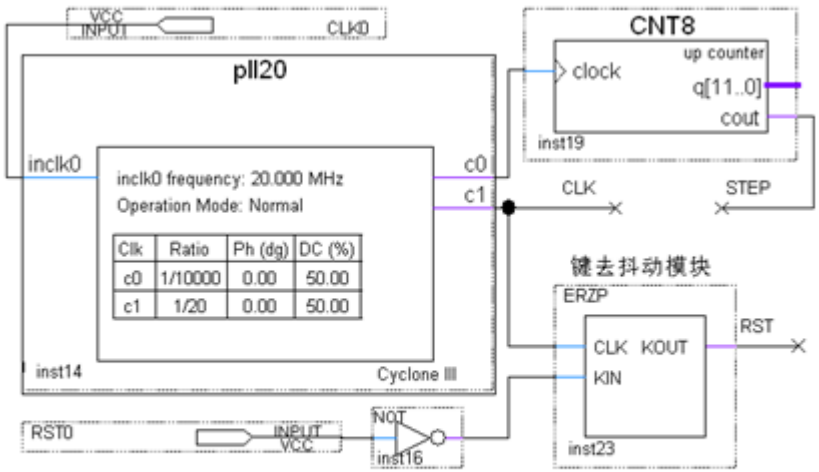


图 5-18 加入锁相环的电路方案 1

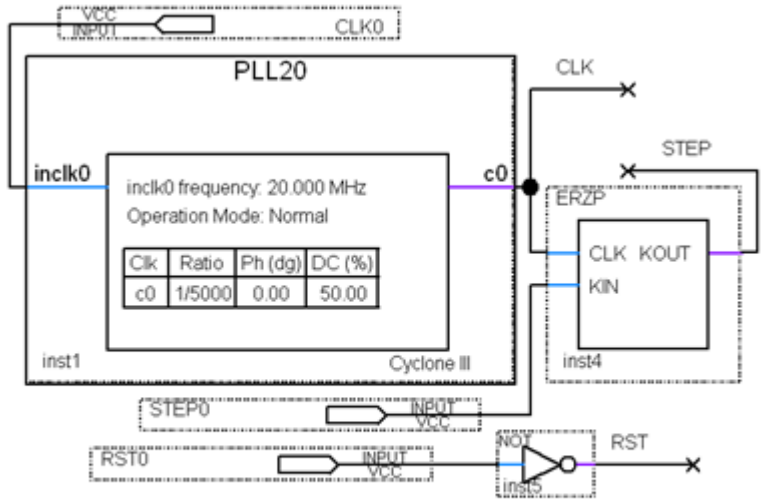


图 5-19 加入锁相环的电路方案 2

5.3 CPU硬件系统设计

5.3.8 模型机系统硬件功能测试

1、嵌入式逻辑分析仪SignalTap II测试与分析

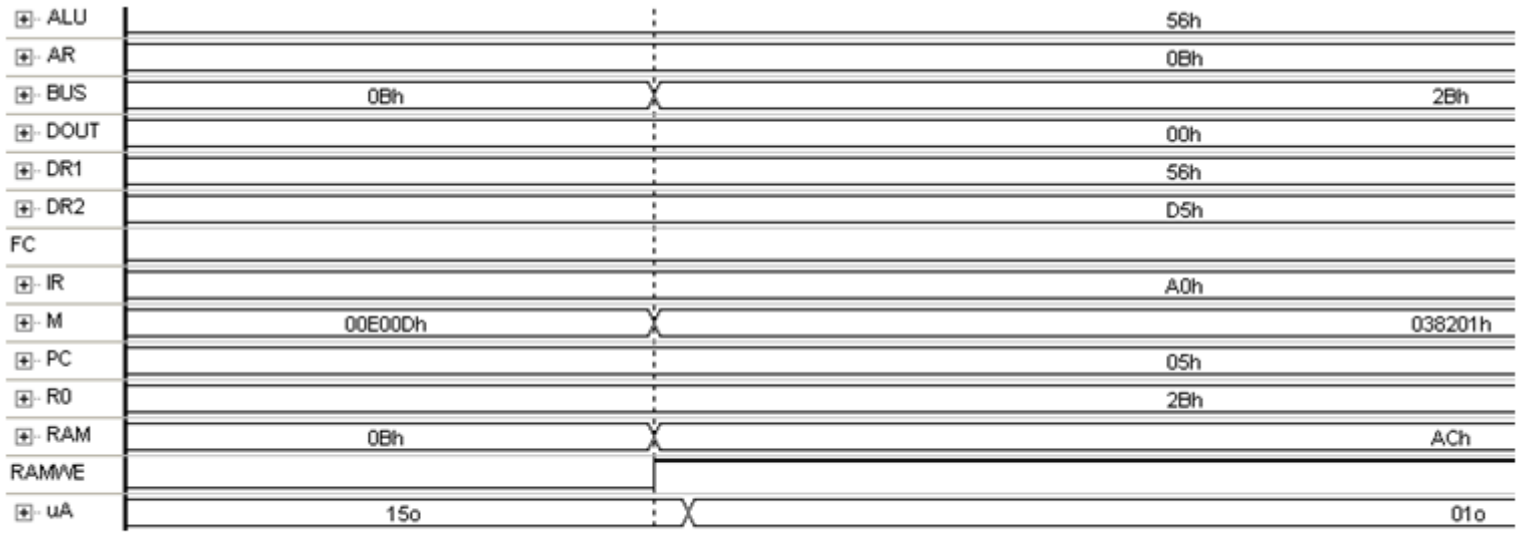


图 5-20 嵌入式锁相环对模型机执行向 RAM 写数据 2BH 时的实时测试波形

5.3 CPU硬件系统设计

5.3.8 模型机系统硬件功能测试

2、利用In-System Sources & Probes进行实时测试

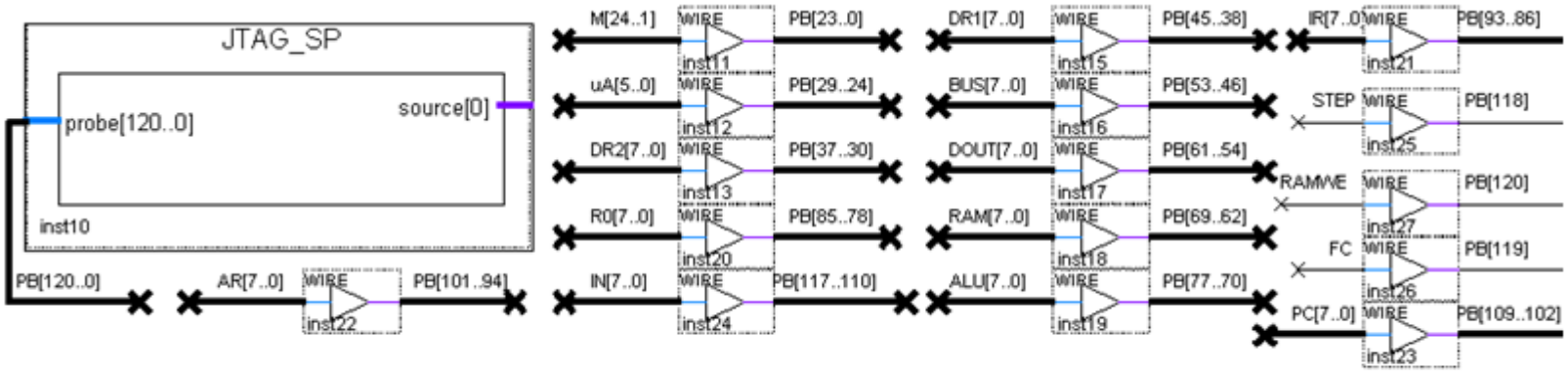


图 5-21　In-System Sources and Probes 与模型机的待测信号连接情况

5.3 CPU硬件系统设计

5.3.8 模型机系统硬件功能测试

2、利用In-System Sources & Probes进行实时测试

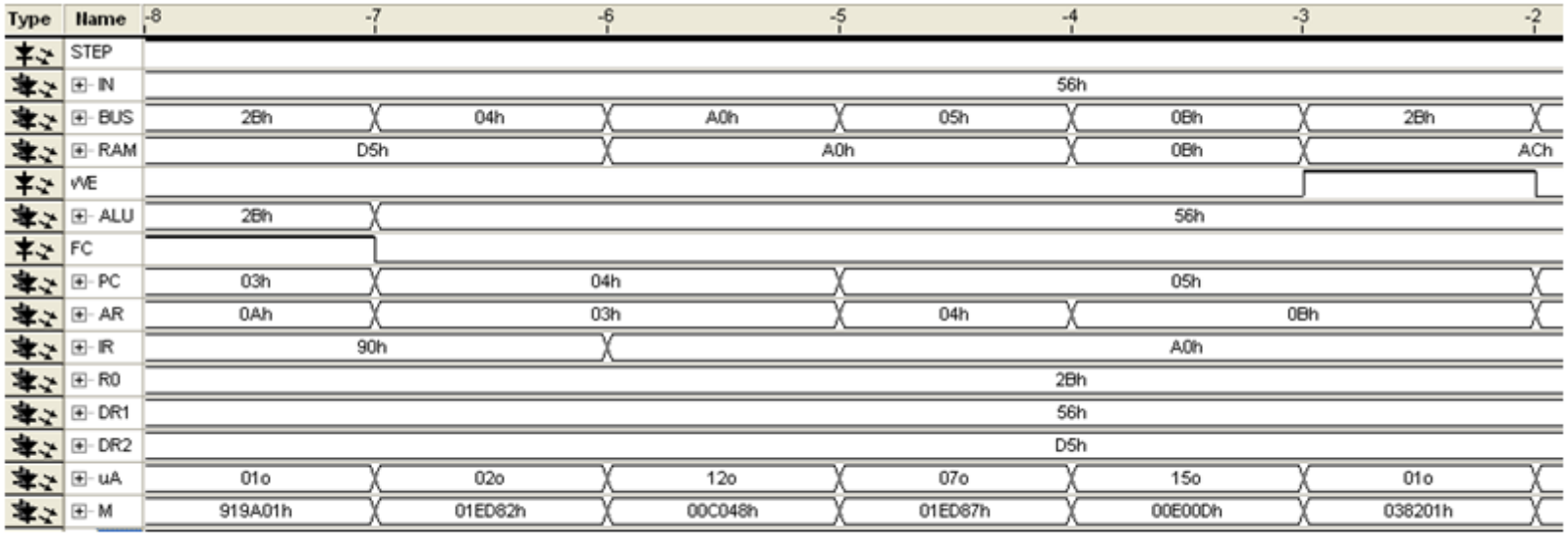


图 5-22　In-System Sources and Probes 测试模型机的实时信号界面

5.3 CPU硬件系统设计

5.3.8 模型机系统硬件功能测试

3、利用In-System Memory Content Editor实时测试内部RAM

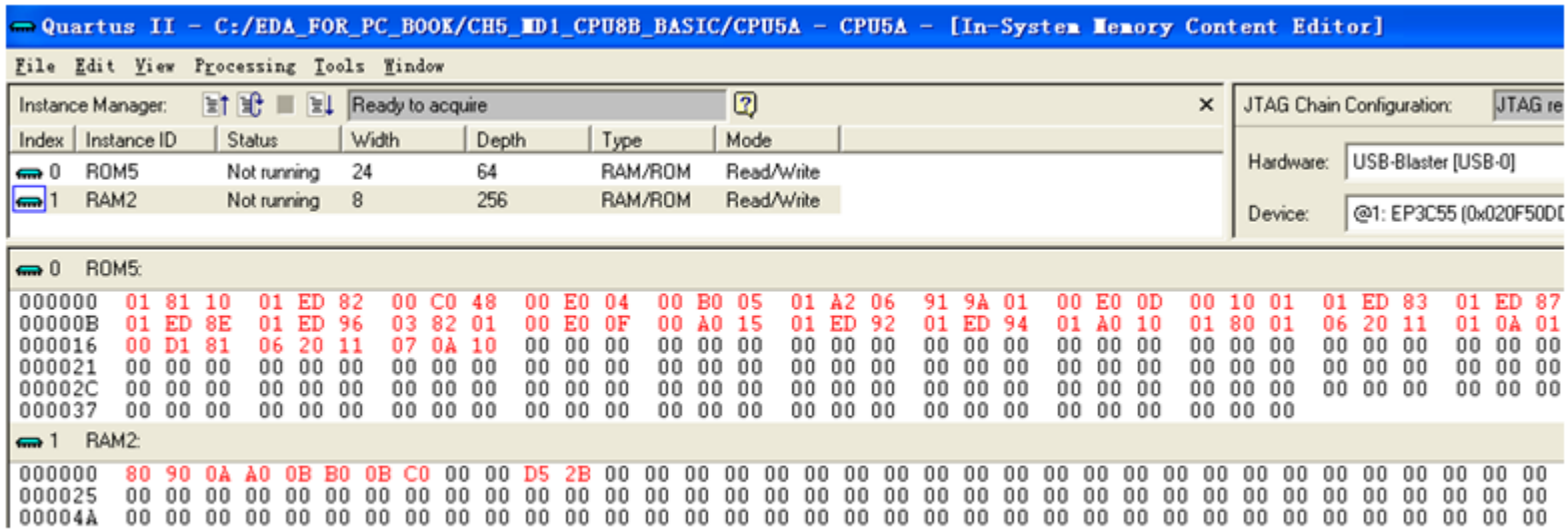


图 5-23 In-System Memory Content Editor 对模型机内 RAM/ROM 数据的实测情况

4、利用外接液晶显示器进行实时观察

5.4 具有移位功能的CPU设计

5.4.1 移位运算器与ALU的结合设计

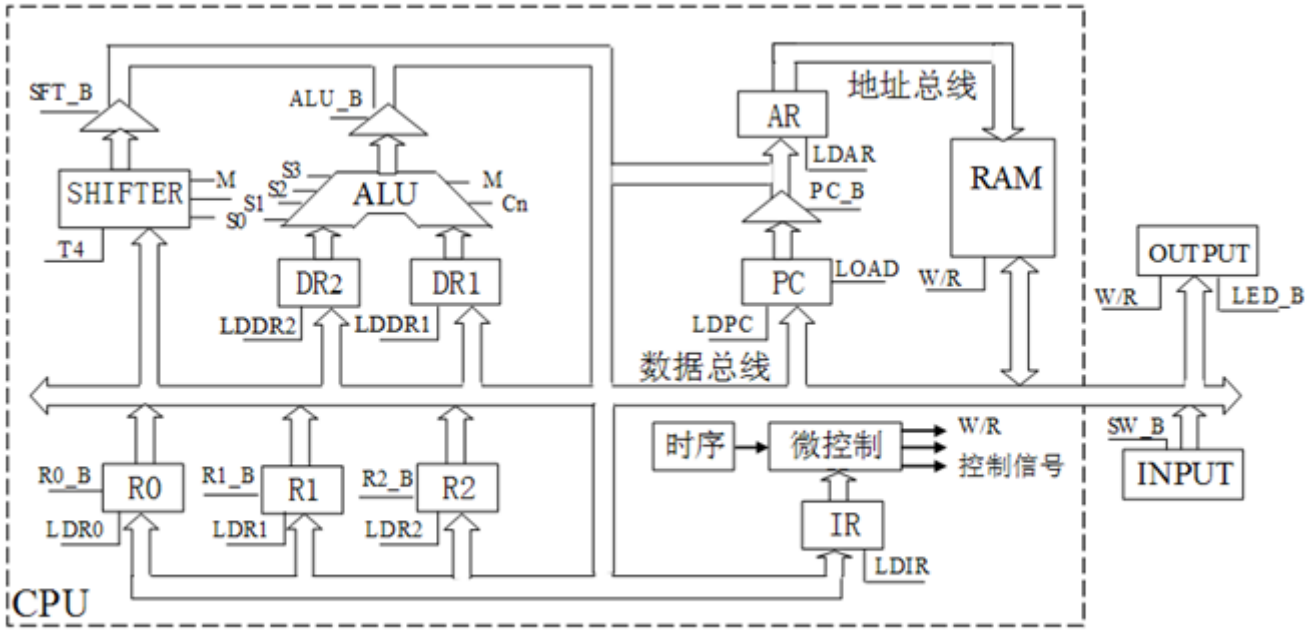


图 5-23　带移位运算的 CPU 数据通路框图

5.4 具有移位功能的CPU设计

5.4.1 移位运算器与ALU的结合设计

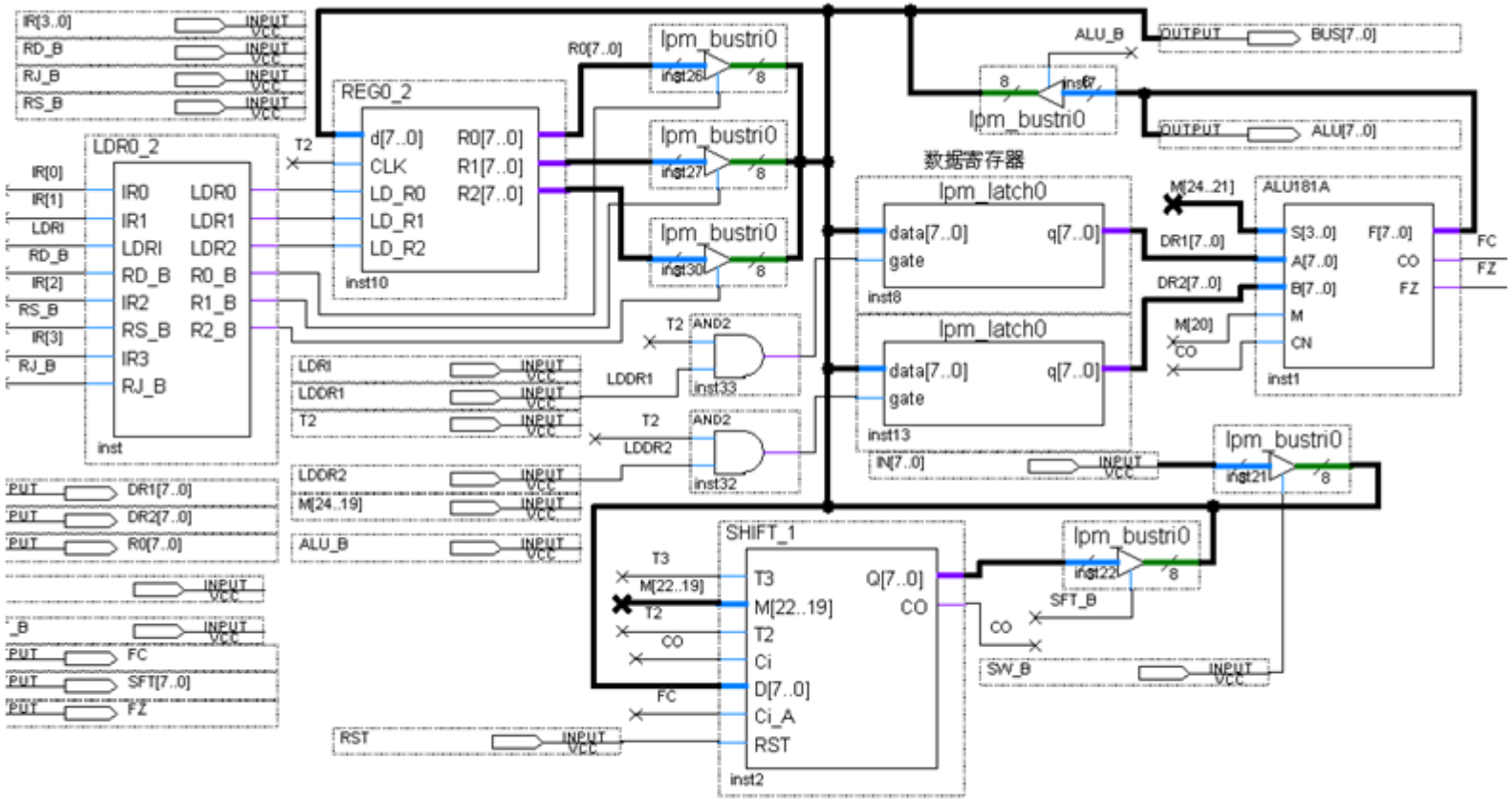


图 5-24 原来图 5-3 中的 ALU_MD 模块加了移位寄存器等辅助电路的新电路

5.4 具有移位功能的CPU设计

5.4.1 移位运算器与ALU的结合设计

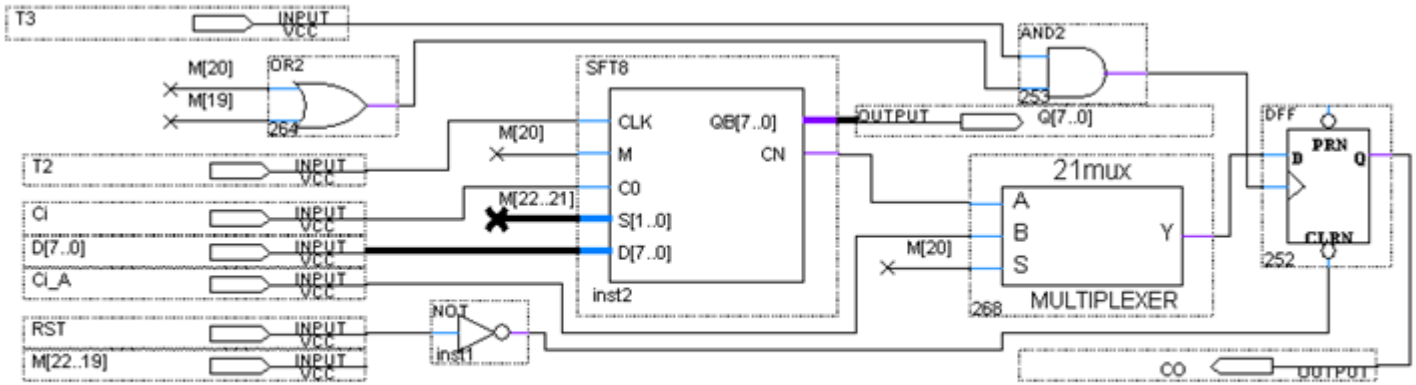


图 5-25 移位运算模块 SFIFT_1 内部电路的结构

5.4 具有移位功能的CPU设计

5.4.1 移位运算器与ALU的结合设计

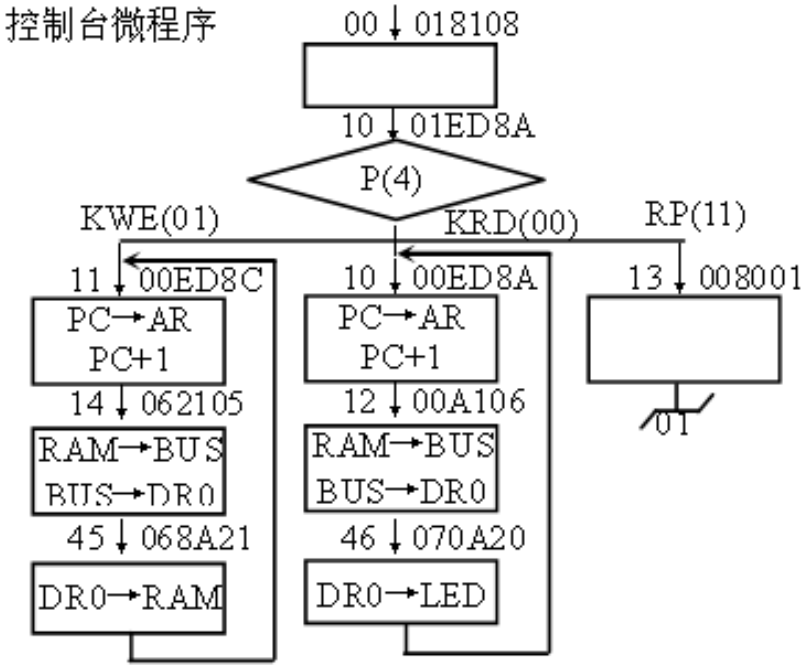


图 5-26　控制台微程序

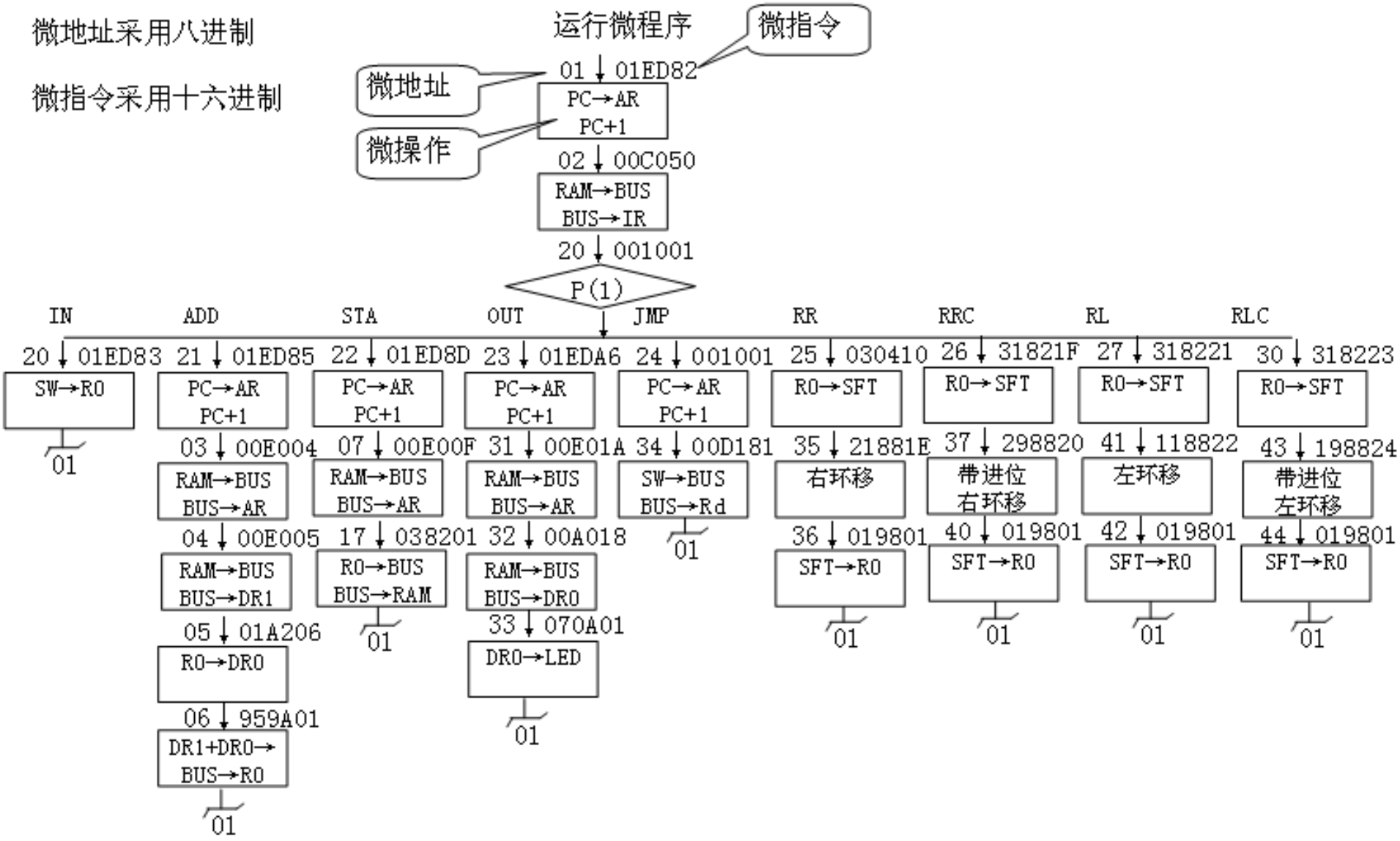


图 5-27 含移位运算指令的微程序流程图

5.4 具有移位功能的CPU设计

5.4.1 移位运算器与ALU的结合设计

表 5-9 带移位运算模型机微程序代码表

微地址	微指令	S3 S2 S1 S0 M CN WE A9 A8	A	B	C	uA5…uA0
00	018108	0 0 0 0 0 0 0 1 1	000	000	100	0 0 1 0 0 0
01	01ED82	0 0 0 0 0 0 0 1 1	110	110	110	0 0 0 0 1 0
02	00C050	0 0 0 0 0 0 0 0 1	100	000	001	0 1 0 0 0 0
…略，详见本实验工程文件 LPM_rom 中的文件 rom_6.mif						
22	019801	0 0 0 0 0 0 0 1 1	001	100	000	0 0 0 0 0 1
23	198824	0 0 0 1 1 0 0 1 1	000	100	000	1 0 0 1 0 0
24	019801	0 0 0 0 0 0 0 1 1	001	100	000	0 0 0 0 0 1

ROM_6.mif

Addr	+0	+1	+2	+3	+4	+5	+6	+7
00	018108	01ED82	00C050	00E004	00B005	01A206	959A01	00E00F
08	01ED8A	01ED8C	00A008	008001	062009	062009	070A08	038201
10	001001	01ED83	01ED87	01ED99	01ED9C	31821D	31821F	318221
18	318223	00E01A	00A01B	070A01	00D181	21881E	019801	298820
20	019801	118822	019801	198824	019801	018110	000000	000000

图 5-28 含移位运算指令的模型机微代码表对应的 mif 文件：ROM_6.mif

5.4 具有移位功能的CPU设计

5.4.2 测试程序设计和模型机时序仿真

RAM_6.MIF

Addr	+0	+1	+2	+3	+4	+5	+6	+7
00	00	10	0D	80	00	60	70	20
08	0E	30	0E	40	00	45	00	00
10	00	00	0E	40	00	00	00	00

图 5-29 测试程序的 mif 文件

表 5-10 测试程序：指令及编码形式

RAM 地址	内容	助记符	说 明
00H	00H	IN	IN 端口数据→R0
01H	10H	ADD [0DH]	R0+[0DH] →R0
02H	0DH		
03H	80H	RLC	带进位循环左移
04H	00H	IN	IN 端口数据→R0
05H	60H	RRC	带进位循环右移
06H	70H	RL	自循环左移
07H	20H	STA [0EH]	R0→[0EH]
08H	0EH		
09H	30H	OUT [0EH]	[0EH]→OUTPUT
0AH	0EH		
0BH	40H	JMP [addr]	00→PC
0CH	00H		自定存数单元
0DH	45H		常数

5.4 具有移位功能的CPU设计

5.4.2 测试程序设计和模型机时序仿真

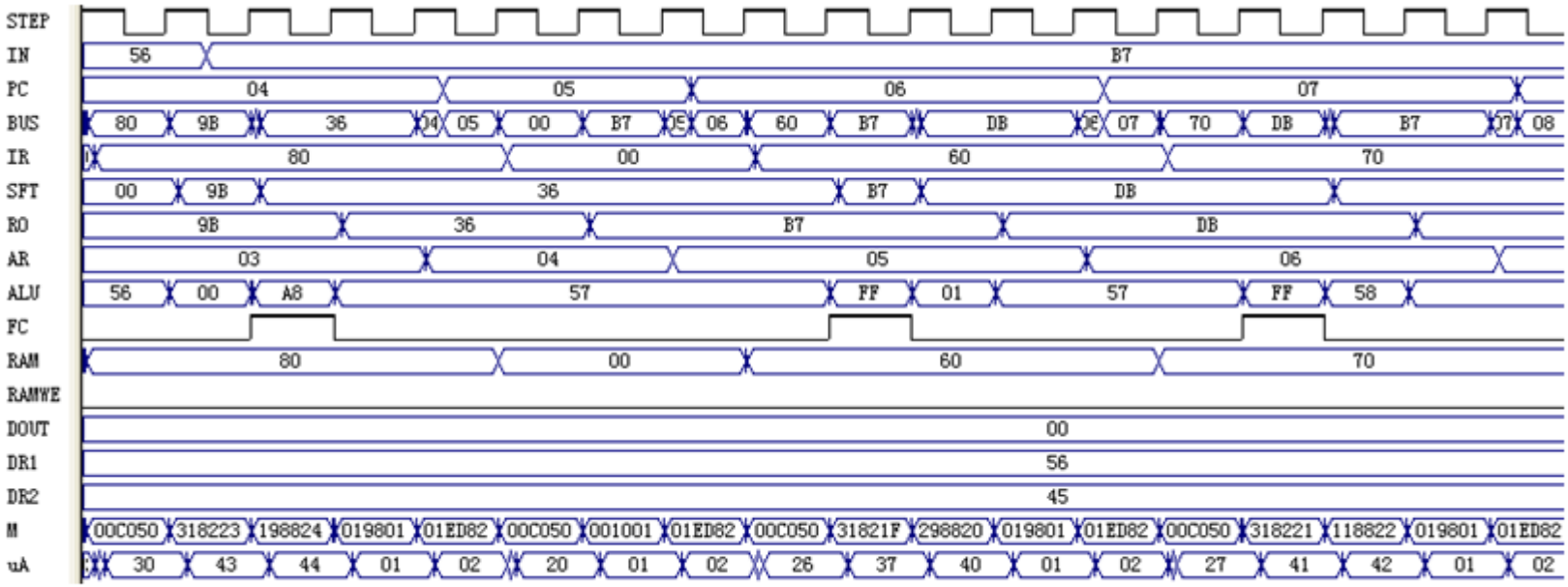


图 5-30 模型机运行表 5-10 程序的部分时序仿真波形

表 5-11　微指令执行流程表

STEP	后续uA微地址	MC微指令	PC	IR指令	完成功能	执行结果
1	00	018108	00	00	控制台（读/写/运行）功能判断	控制台操作，微指令从00H开始执行
2	13	018101			SWB、WSA=（11）转RP，分支转移	
3	01	008001			转程序执行方式	
4	02	01ED82	01		执行第1条指令.（输入IN）	PC→AR=00H，PC+1=01H
5	20	00C050		00	取指令,将RAM中的指令送指令寄存器	RAM(00H)=00→BUS→IR=00H
6	01	001001			接收IN输入端口的数据，送R0寄存器	R0=56H，IN端口输入数据56H
7	02	01ED82	02		执行第2条指令，（加法ADD）	PC→AR=01H，PC+1=02H，指向指令地址
8	21	00C050		10	取指令,将RAM中的指令送指令寄存器	AR=01H，RAM=10H→BUS→IR=10H
9	03	01ED83	03		指向操作数地址	PC→AR=02H，PC+1=03H
10	04	00E004			间接寻址，以RAM内容做操作数地址	RAM(02H)=0DH→BUS→AR=0DH
11	05	00B005			将操作数送DR2	RAM(0DH)=45H→BUS→DR2=45H
12	06	01A206			将R0的内容送DR1	(R0)=56H→BUS→DR1=56H
13	01	959A01			完成加法运算：(DR1)+(DR2) →R0	56H+45H=9BH，ALU=9BH，R0=9BH
14	02	01ED82	04		执行第3条指令，指向指令地址	PC→AR=03H，PC+1=04H
15	30	00C050		80	取指令 （带C左循环RLC）	IR=80H
16	43	318223			R0的内容送移位寄存器SFT	(R0)=9BH →BUS→SFT=9BH
17	44	198824			带进位C左循环	SFT=36H，BUS=36H
18	01	019801			结果送R0	SFT=36H→BUS→ R0=36H
19	02	01ED82	05		执行第4条指令，指向指令地址	PC→AR=04H，PC+1=05H，IN端口B7H
20	20	00C050		00	取指令 （输入IN）	IN=B7H，指令IR=00H，
21	01	001001			接收键1、2输入的数据，送R0寄存器	R0=B7H
22	02	01ED82	06		执行第5条指令，指向指令地址	PC→AR=05H，PC+1=06H
23	26	00C050		60	取指令 （带C右循环RRC）	指令IR=60H
24	37	31821F			R0的内容送移位寄存器SFT	(R0)=B7H→BUS→SFT=B7H
25	40	298820			SFT带进位C右循环	SFT=DBH
26	01	019801			结果送R0	SFT=DBH→BUS→ R0=DBH

27	02	01ED82	07		执行第6条指令，指向指令地址	PC→AR=06H，PC+1=07H
28	27	00C050		70	取指令 （左循环RL）	指令IR=70H
29	41	318221			R0的内容送移位寄存器SFT	(R0)=DBH →BUS→SFT=DBH
30	42	118822			SFT不带进位左循环	SFT=B7H
31	01	019801			结果送R0	SFT=B7H→BUS→ R0=B7H
32	02	01ED82	08		执行第7条指令，指向指令地址	PC→AR=07H，PC+1=08H
33	22	00C050		20	取指令 （存储STA）	指令IR=20H
34	07	01ED87	09		间接寻址，以RAM内容做存数地址	PC→AR=08H，PC+1=09H，RAM=20H
35	17	00E00F			RAM内容送地址寄存器AR	RAM(08H)=0EH→BUS→AR=0EH
36	01	038201			将R0的内容存入RAM(0EH)地址单元	R0=B7H→BUS→RAM(0EH)=B7H
37	02	01ED82	0A		执行第8条指令，指向指令地址	PC→AR=09H，PC+1=0AH
38	23	00C050		30	取指令（输出OUT）	RAM(09)=30H→BUS→IR=30H(指令)
39	31	01ED99	0B		间接寻址，以RAM内容作取数地址	PC→AR=0AH，PC+1=0BH
40	32	00E01A			RAM内容送地址寄存器AR	RAM(0AH)=0EH→BUS→AR=0EH
41	33	00A01B			从RAM(0EH)取数，送DR1	RAM(0EH)=B7H→BUS→DR1=B7H
42	01	070A01			DR1的内容送OUT输出端口	(DR1)= B7H→BUS→OUT=B7H
43	02	01ED82	0C		执行第9条指令，指向指令地址	PC→AR=0BH，PC+1=0CH
44	24	00C050		40	取指令 （转移JMP）	指令IR=40H
45	34	01ED9C	0D		间接寻址，以RAM内容作转移地址	PC→AR=0CH，PC+1=0DH
46	01	00D181	00		将RAM内容送PC，实现程序转移	RAM(0CH)=00. →BUS→PC=00H
47	02	01ED82	01		执行第1条指令，程序循环	PC→AR=00H，PC+1=01H
48	20	00C050		00	取指令	指令IR=00H
...	01	001001		00	输入数据……	

5.5 含更多指令的模型机设计

5.5.1 指令系统的格式与指令

1. 数据格式。

表 5-12 数据格式

7	6 5 4 3 2 1 0
符号	尾 数

2. 指令格式。

（1）算术逻辑指令

表 5-13 算术逻辑指令格式

7 6 5 4	3 2	1 0
操 作 码	Rs	Rd

表 5-14 寄存器编码

Rs 或 rd	选定的寄存器
00	R0
01	R1
10	R2

5.5 含更多指令的模型机设计

5.5.1 指令系统的格式与指令

2．指令格式。

（2）访问指令及转移指令。

表 5-15 寄存器编码

7 6	5 4	3 2	1 0
00	M	操作码	Rd
D			

表 5-16 输入输出指令格式

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

（3）I/O指令。

表 5-17 输入输出指令格式

7 6 5 4	3 2	1 0
操作码	addr	Rd

（4）停机指令。

表 5-18 输入输出指令格式

7 6 5 4	3 2	1 0
操作码	00	00

5.5 含更多指令的模型机设计

表 5-19 含有移位功能 CPU 的指令系统

助记符号	指令格式	功能
CLR rd	0111 | 00 | rd	0 → rd
MOV rs, rd	1000 | rs | rd	rs → rd
ADC rs, rd	1001 | rs | rd	rs + rd + cy → rd
SBC rs, rd	1010 | rs | rd	rs - rd - cy → rd
INC rd	1011 | | rd	rd + 1 → rd
AND rs, rd	1100 | rs | rd	rs ∧ rd → rd
COM rd	1101 | | rd	$\overline{rd}$ → rd
RRC rs, rd	1110 | rs | rd	在 rs 中带进位右移后送 rd
RLC rs, rd	1111 | rs | rd	在 rs 中带进位左移后送 rd
LDA M, D, rd	00 | M | 00 | rd D	E → rs
STA M, D, rd	00 | M | 01 | rd D	rd → E
JMP M, D	00 | M | 10 | rd D	E → PC
BZC M, D	00 | M | 11 | rd D	当 CY=1 或 Z=1 时, E → PC
IN addr, rd	0100 | 01 | rd	addr → rd
OUT addr, rd	0101 | 10 | rd	rd → addr
HALT	0110 | 00 | 00	停机

5.5 含更多指令的模型机设计

5.5.2 微程序控制流程图设计

1、设计微操作流程图

2、确定微地址

3、确定微指令码

4、确定下地址

5、确定微指令

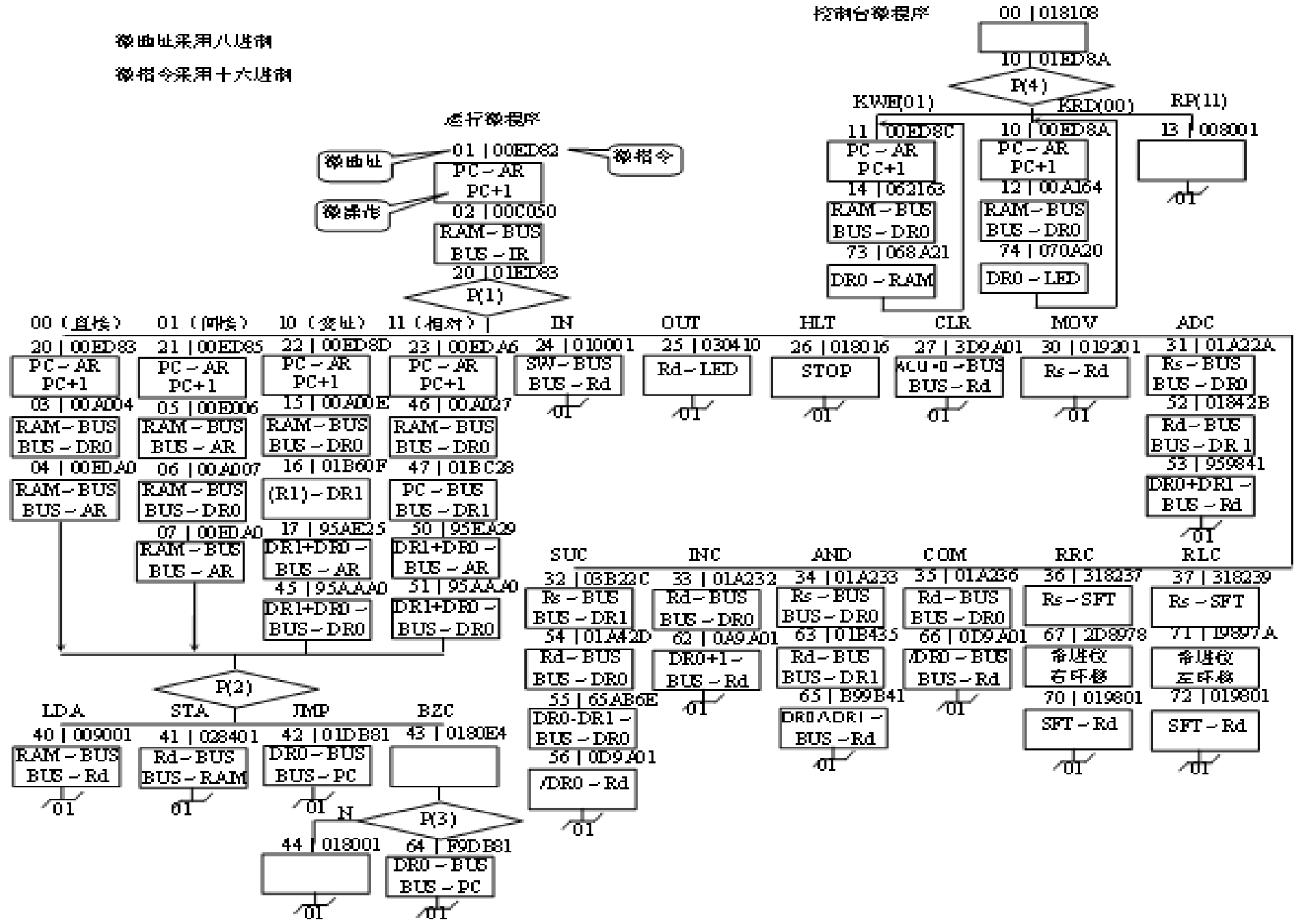

微地址采用八进制
微指令采用十六进制
检测台微程序
00 | 018108
10 | 01ED8A
P(4)
KWE(01)
KRD(00)
RP(11)
11 | 00ED8C
PC → AR
PC+1
14 | 062163
RAM → BUS
BUS → DR0
73 | 068A21
DR0 → RAM
10 | 00ED8A
PC → AR
PC+1
12 | 00A164
RAM → BUS
BUS → DR0
74 | 070A20
DR0 → LED
13 | 008001
01
执行微程序
01 | 00ED82
微地址
微指令
PC → AR
PC+1
02 | 00C050
RAM → BUS
BUS → IR
20 | 01ED83
P(1)
00（直接）
01（间接）
10（变址）
11（相对）
IN
OUT
HLT
CLR
MOV
ADC
20 | 00ED83
PC → AR
PC+1
03 | 00A004
RAM → BUS
BUS → DR0
04 | 00ED A0
RAM → BUS
BUS → AR
21 | 00ED85
PC → AR
PC+1
05 | 00ED06
RAM → BUS
BUS → AR
06 | 00A007
RAM → BUS
BUS → DR0
07 | 00ED A0
RAM → BUS
BUS → AR
22 | 00ED8D
PC → AR
PC+1
15 | 00A00E
RAM → BUS
BUS → DR0
16 | 01B60F
(R1) → DR1
17 | 95AE25
DR1+DR0 →
BUS → AR
45 | 95AAA0
DR1+DR0 →
BUS → DR0
23 | 00EDA6
PC → AR
PC+1
46 | 00A027
RAM → BUS
BUS → DR0
47 | 01BC28
PC → BUS
DR1 → DR1
50 | 95EA29
DR1+DR0 →
BUS → AR
51 | 95AAA0
DR1+DR0 →
BUS → DR0
24 | 010001
SW → BUS
BUS → Rd
01
25 | 030410
Rd → LED
01
26 | 018016
STOP
01
27 | 3D9A01
ALU=0 → BUS
BUS → Rd
01
30 | 019201
Rs → Rd
01
31 | 01A22A
Rs → BUS
BUS → DR0
52 | 01842B
Rd → BUS
BUS → DR1
53 | 959841
DR0+DR1 →
BUS → Rd
01
P(2)
LDA
STA
JMP
BZC
40 | 009001
RAM → BUS
BUS → Rd
01
41 | 028401
Rd → BUS
BUS → RAM
01
42 | 01DB81
DR0 → BUS
BUS → PC
01
43 | 0180E4
N
P(3)
44 | 018001
01
64 | F9DB81
DR0 → BUS
BUS → PC
01
SUC
INC
AND
COM
RRC
RLC
32 | 03B22C
Rs → BUS
BUS → DR1
54 | 01A42D
Rd → BUS
BUS → DR0
55 | 65AB6E
DR0-DR1 →
BUS → DR0
56 | 0D9A01
/DR0 → Rd
01
33 | 01A232
Rd → BUS
BUS → DR0
62 | 0A9A01
DR0+1 →
BUS → Rd
01
34 | 01A233
Rs → BUS
BUS → DR0
63 | 01B435
Rd → BUS
BUS → DR1
65 | B99B41
DR0∧DR1 →
BUS → Rd
01
35 | 01A236
Rd → BUS
BUS → DR0
66 | 0D9A01
/DR0 → BUS
BUS → Rd
01
36 | 318237
Rs → SFT
67 | 2D8978
带进位
右环移
70 | 019801
SFT → Rd
01
37 | 318239
Rs → SFT
71 | 1D8978
带进位
左环移
72 | 019801
SFT → Rd
01

图 5-31 微程序流程图

5.5 含更多指令的模型机设计

5.5.2 微程序控制流程图设计

表 5-20 微程序代码表

微地址	微指令	S3 S2 S1 S0 M CN WE A9 A8	A	B	C	UA5—UA0
0 0	0 1 8 1 0 8	0 0 0 0 0 0 0 1 1	0 0 0	0 0 0	1 0 0	0 0 1 0 0 0
0 1	0 1 E D 8 2	0 0 0 0 0 0 0 1 1	1 1 0	1 1 0	1 1 0	0 0 0 0 1 0
0 2	0 0 C 0 5 0	0 0 0 0 0 0 0 0 1	1 0 0	0 0 0	0 0 1	0 1 0 0 0 0
…略，详见本实验工程文件 \CPU7.bdf 中元件 LPM_ROM0 中的文件 rom_7.mif						
7 1	1 9 8 9 7 A	0 0 0 1 1 0 0 1 1	0 0 0	1 0 0	1 0 1	1 1 1 0 1 0
7 3	0 1 9 8 0 1	0 0 0 0 0 0 0 1 1	0 0 1	1 0 0	0 0 0	0 0 0 0 0 1
7 4	0 7 0 A 0 8	0 0 0 0 0 1 1 1 0	0 0 0	1 0 1	0 0 0	0 0 1 0 0 0
7 5	0 6 2 0 0 9	0 0 0 0 0 1 1 0 0	0 1 0	0 0 0	0 0 0	0 0 1 0 0 1

Addr	+0	+1	+2	+3	+4	+5	+6	+7
00	018108	00ED82	00C050	00E004	00B005	01A206	959A01	00E00F
08	00ED8A	00ED8C	00A008	008001	062009	062009	070A08	038201
10	001001	00ED83	00ED87	00ED99	00ED9C	31821D	31821F	318221
18	318223	00E01A	00A01B	070A01	00D181	21881E	019801	298820
20	019801	118822	019801	198824	019801	018110	000002	000003
28	000004	000005	000006	000007	000008	000009	00000A	00000B
30	00000C	00000D	00000E	00000F	000010	000011	000012	000013
38	000014	000015	000016	000017	000018	000019	00001A	00001C

图 5-32 rom_7.mif 中的数据

5.5 含更多指令的模型机设计

5.5.3 程序编辑与系统仿真

表 5-21 示例程序及代码表

RAM 地址	机器码	助记符	说　明
00H	44H	IN　01，R0	IN 端口数据→R0
01H	46H	IN　01，R2	IN 端口数据→R0
02H	98H	ADC　R2，R0	(R0)+(R2)→R0
03H	81H	MOV　R0，R1	(R0)→R1
04H	F5H	RLC　R1，R1	R1 带进位左移→R1
05H	0CH	BZC　00，00	Cy,Z 为 1 时，循环
06H	00H		
07H	08H	JMP　00	GOTO　00
08H	00H		

RAM_7.MIF

Addr	+0	+1	+2	+3	+4	+5	+6	+7
00	44	46	98	81	F5	0C	00	08
08	00	00	00	00	00	00	00	00
10	00	00	00	00	00	00	00	00
18	00	00	00	00	00	00	00	00
20	00	00	00	00	00	00	00	00

图 5-33 程序代码的 mif 文件窗

5.5 含更多指令的模型机设计

5.5.3 程序编辑与系统仿真

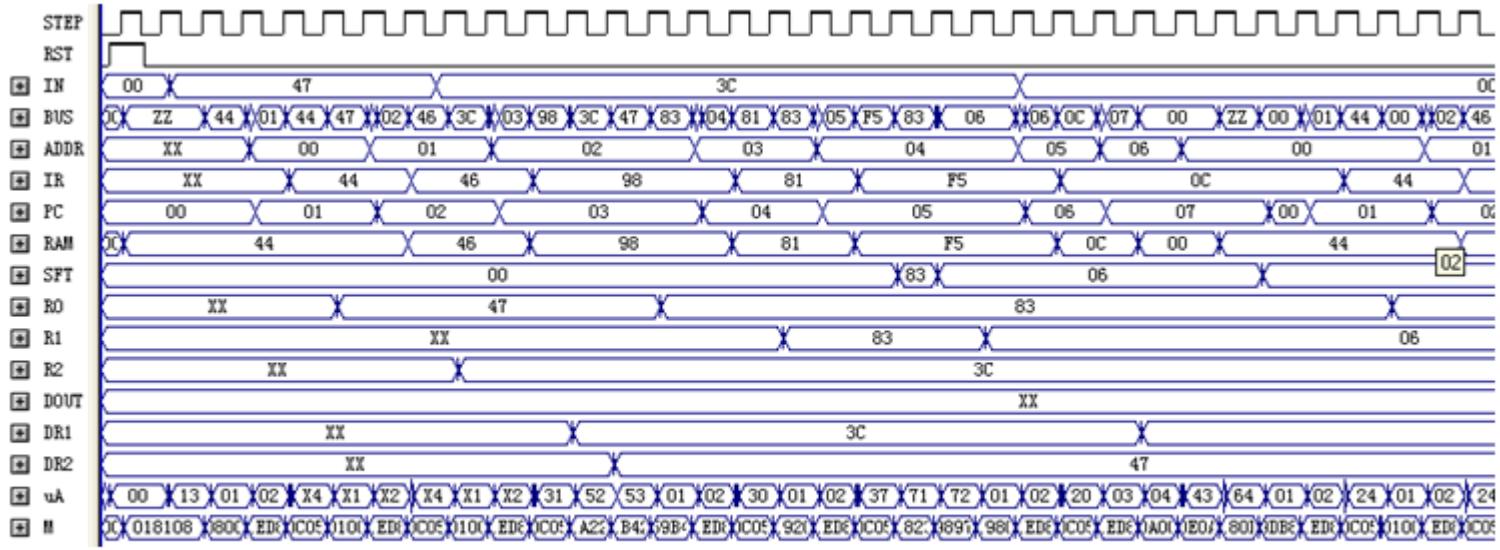


图 5-34 模型机运行表 5-21 程序的仿真波形图

实验与设计

5-1. 基本模型计算机设计与实现

表 5-22　LCD 液晶显示屏功能说明

名称	作 用	名称	作 用
IN	输入单元 INPUT	DR1	暂存器 DR1
OUT	输出单元 OUTPUT	DR2	暂存器 DR2
ALU	算术逻辑单元	PC	程序计数器
BUS	内部数据总线	AR	地址寄存器
R0	寄存器 R0	RAM	程序/数据存储器
R1	寄存器 R1	IR	指令寄存器
R2	寄存器 R2	MC	微程序控制器

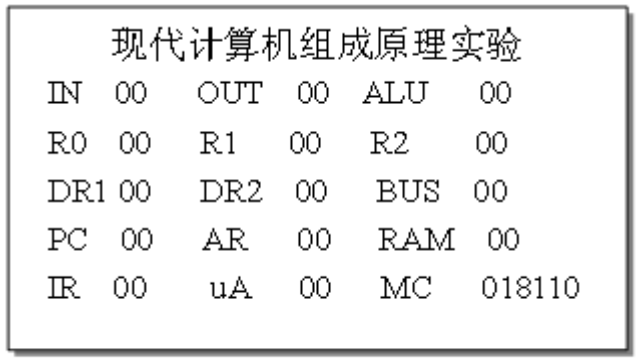


图 5-35　LCD 液晶显示屏

实验与设计

5-2. 带移位运算的模型机设计与实现

5-3. 含**16**条指令的**CPU**设计与实现

表 5-23 微指令执行流程表

STEP	后续 uA 微地址	MC 微指令	PC	IR 指令	完成功能	执行结果
1	00	018108	00	44	控制台（读/写/运行）功能判断	控制台操作，微指令从 00H 开始执行
2	13	01ED8A			SWB、SWA=（11）转 RP，分支转移	P(4)分支检测
3	01	008001			转程序执行方式	键 1、键 2 输入数据 45H
4	02	01ED82	01		执行第 1 条指令（输入 IN 01，R0）	PC→AR=00H 指向指令地址，PC+1=01H
5	24	00C050		44	取指令，将 RAM 中的指令送指令寄存器	RAM(00H)=44→BUS→IR=44H
6	01	001001			接收 IN 输入端口的数据，送寄存器 R0	R0=45H，
7	02	01ED82	02		执行第 2 条指令（输入 IN 01，R2）	PC→AR=01H 指向指令地址，PC+1=02H
8	24	00C050		46	取指令，将 RAM 中的指令送指令寄存器	AR=01H，RAM=46H→BUS→IR=46H
9	01	001001			接收 IN 输入端口的数据，送寄存器 R2	R2=3CH，**键 1、键 2 输入数据 3CH**
10	02	01ED82	03		执行第 3 条指令（ADC R2，R0）	PC→AR=02H 指向指令地址，PC+1=03H
11	31	00C050		98	取指令，将 RAM 中的指令送指令寄存器	AR=02H，RAM=98H→BUS→IR=98H
12	52	01A22A			取源操作数	R0=45H→BUS→DR0=45H
13	53	01B42B			取目的操作数	R0=3CH→BUS→DR1=3CH
14	01	959B41			R2+R0→R0	R2+R0=81H→BUS→R0=81H
15	02	01ED82	04		执行第 4 条指令（MOV R0，R1）	PC→AR=03H 指向指令地址，PC+1=04H
16	30	00C050		81	取指令，将 RAM 中的指令送指令寄存器	AR=03H，RAM=81H→BUS→IR=81H
17	01	019201			R0R1	R0=81H→BUS→R1=81H
18	02	01ED82	05		执行第 5 条指令（RLC R1，R1）	PC→AR=04H 指向指令地址，PC+1=05H
19	37	00C050		F5	取指令，将 RAM 中的指令送指令寄存器	AR=04H，RAM=F5H→BUS→IR=F5H
20	71	318239			R1 的数据送移位寄存器 SFT	R1=81→BUS→SFT=81H
21	72	19897A			带进位循环左移	SFT 带进位循环左移=02H
22	01	019801			移位运算后的结果送 R1	SFT=03H→BUS→R1=02H
23	02	01ED82	06		**执行第 6 条指令**（BZC 00，00）	PC→AR=05H 指向指令地址，PC+1=06H
24	14	00C050		0C	取指令，将 RAM 中的指令送指令寄存器	AR=05H，RAM=0CH→BUS→IR=0CH
25	03	01ED83	07		直接地址转移，从 RAM 中取转移地址	PC→AR=06H 指向指令地址，PC+1=07H
26	04	00A004			转移地址→DR0	RAM=00H→BUS→DR0=00H
27	43	00E0A0			转移地址→地址寄存器 AR	RAM=00H→BUS→AR=00H
28	44	0180E4			分支转移失败，顺序执行	PC=07
29	01	018001	08		**执行第 7 条指令**（JMP 00）	PC→AR=07H 指向指令地址，PC+1=08H
30	02	01ED82		08	取指令，将 RAM 中的指令送指令寄存器	AR=07H，RAM=08H→BUS→IR=08H
31	14	00C050	09		从 RAM 中取转移地址	PC→AR=08H 指向指令地址，PC+1=09H
32	03	01ED83			转移地址→DR0	AR=08H，RAM=00H→BUS→DR0=00H
33	04	00A004			转移地址→地址寄存器 AR	RAM=00H→BUS→AR=00H
34	42	00E0A0	00		无条件转移 GOTO 00	DR0=00→BUS→PC=00
35	01	01DB81	00		**执行第 1 条指令**（IN 01，R0）	PC→AR=00H 指向指令地址，PC+1=01H
36	02	00C048		44	取指令，将 RAM 中的指令送指令寄存器	AR=0AH，RAM=D1H→BUS→IR=D1H
37	…					

实验与设计

5-4. 较复杂CPU应用程序设计实验

表 5-24 实验程序

	汇编语言源程序：	功　能
LP0：	IN　　R0	从开关输入任意一个整数 n—>R0
	MOV　R1，1	将立即数 1—>R1（R1 存放参与运算的奇数）
	MOV　R2，0	将立即数 0—>R2（R2 存放累加和）
LP1：	CMP　R0，R1	将 R0 中的整数 n 与 R1 中的奇数进行比较
	JB　　LP2	若 R1<R0，则转到 LP2 处执行
	ADD　R1，R2	否则，累加求和
	INC　R1	R1 的内容加 2，形成下一个奇数
	INC　R1	
	JMP　LP1	跳转到 LP1 继续执行
LP2：	OUT　R2	输出累加和
	JMP　LP0	重新开始

实验与设计

5-4. 较复杂CPU应用程序设计实验

表 5-25 I/O 指令格式

7 6 5 4	3 2	1 0
操作码	Addr	目的寄存器

表 5-26 比较与加法指令格式

7 6 5 4	3 2	1 0
操作码	Rs	rd

表 5-27 寄存器对应的编码

Rs 或 rd	选定的寄存器
00	R0
01	R1
10	R2

表 5-28 转移等指令格式

7 6 5 4	3 2 1 0
操作码	X X X X
地 址	

表 5-29 MOV 指令格式

7 6 5 4	3 2	1 0
操作码	X X	R_d
立 即 数		

表 5-30 INC 指令格式

7654	3 2	1 0
操作码	X X	R_d

表 5-31 数据格式

7	6 5 4 3 2 1 0
符号位	尾 数

实验与设计

5-4. 较复杂CPU应用程序设计实验

表 5-32 指令系统

助记符号	指令格式	功 能
IN rd	1000 | XX | rd	input → rd 寄存器
OUT rd	1111 | XX | rd	rd → output
ADD rs, rd	1100 | rs | rd	rs + rd → rd
CMP rs, rd	1010 | rs | rd	rs −rd → rd
INC rd	1101 | XX | rd	Rd +1→ rd
MOV data, rd	1001 | XX | rd data	data rd
JMP addr	1110 | XXXX addr	Addr →PC
JB addr	1011 | XXXX addr	若小于，则 addr → PC

实验与设计

5-4. 较复杂CPU应用程序设计实验

表 5-26 实验程序

助记符		地址	机器代码	功能
LP0：	IN R0	00H	80H	Input → R0
	MOV R1，1	01H	91H	1→R1
		02H	01H	
	MOV R2，0	03H	92H	0→R2
		04H	00H	
LP1：	CMP R0，R1	05H	A1H	R0-R1→R1
	JB L2	06H	B0H	(LP2) →PC
		07H	0DH	
	ADD R1，R2	08H	C6H	R1+R2→R2
	INC R1	09H	D1H	R1+1→R1
	INC R1	0AH	D1H	R1+1→R1
	JMP L1	0BH	E0H	(LP1)→PC
		0CH	05H	
LP2：	OUT R2	0DH	F2H	R2→output
	JMP LP0	0EH	E0H	(LP0)→PC
		0FH	00H	

实验与设计

5-4. 较复杂CPU应用程序设计实验

表 5-27 微地址和微指令表

微地址	微指令	S3	S2	S1	S0	M	CN	WE	A9	A8	A	B	C	uA5—uA0
0 0	0 1 8 1 1 0	0	0	0	0	0	0	0	1	1	0 0 0	0 0 0	1 0 0	0 1 0 0 0 0
0 1	0 1 E D 8 2	0	0	0	0	0	0	0	0	1	1 1 0	1 1 0	1 1 0	0 0 0 0 1 0
0 2	0 0 C 0 4 8	0	0	0	0	0	0	0	0	1	1 0 0	0 0 0	0 0 1	0 0 1 0 0 0
…略，详见本实验工程文件 rom_8.mif														
2 7	0 0 D 1 8 1	0	0	0	0	0	0	0	0	1	1 0 1	0 0 0	1 1 0	0 0 0 0 0 1
3 0	0 0 D 1 8 1	0	0	0	0	0	0	0	0	1	1 0 1	0 0 0	1 1 0	0 0 0 0 0 1
3 1	9 1 9 A 0 1	0	0	0	0	0	0	0	1	1	0 0 0	0 0 0	0 0 0	0 0 0 0 0 1
3 2	9 1 9 B 4 1	1	0	0	1	0	0	0	1	1	0 0 1	1 0 1	1 0 1	0 0 0 0 0 1

表 3-38 微指令执行情况

STEP	指令 ROM 单地址	ROM 微指令	PC	IR 指令	完成功能	执行结果
1	00	018120	00	80	控制Bar（读/写/运行）功能判断	控制Bar操作，微指令从 008 开始执行
2	15	018120			IRD，IRar（11）转 RP，分支转移	P(4)分支检测
3	01	008001			转程序执行方式	
4	02	018082	01		执行第 1 条指令，（输入 IR 80）	PC→AR=008 指向指令地址，PC+1=018
5	10	000048		80	取指令，将 RAM 中的指令送指令寄存器	RAM[008]=00→BUS→IR=80H
6	01	001001			操作 IR 输入端口的数据，送 R0 寄存器	R0=0FH，键 1、键 2 输入数据 0FH
7	02	018082	02		执行第 2 条指令，（MOV R1，1）	PC→AR=018 指向指令地址，PC+1=028
8	11	000048		01	取指令，将 RAM 中的指令送指令寄存器	AR=028，RAM=02H→BUS→IR=01H
9	03	018083			指向操作数地址	PC→AR=028 指向操作数地址，PC+1=038
10	01	008001			RAM 内容送 R1	RAM[028]=03H→BUS→R1=03H
11	02	018082	04		执行第 3 条指令（MOV R2，0）	PC→AR=038 指向指令地址，PC+1=048
12	11	000048		02	取指令，将 RAM 中的指令送指令寄存器	AR=038，RAM=03H→BUS→IR=02H
13	03	018083			指向操作数地址	PC→AR=048 指向操作数地址，PC+1=058
14	01	008001			RAM 内容送 R2	RAM[048]=00H→BUS→R2=00H
15	02	018082	06		执行第 4 条指令（CMP R0，R1）	PC→AR=058 指向指令地址，PC+1=068
16	12	000048		A1	取指令，将 RAM 中的指令送指令寄存器	AR=058，RAM=A1H→BUS→IR=A1H
17	04	01A204			取源操作数	R0→BUS→DR0=07H
18	05	01B405			取目的操作数	R1→BUS→DR1=01H
19	01	018841			R0-R1，标志位→PC	DR0-DR1=06H，PC=0
20	02	018082	07		执行第 5 条指令（JB LP2）	PC→AR=068 指向指令地址，PC+1=078
21	13	000048		B0	取指令，将 RAM 中的指令送指令寄存器	AR=068，RAM=B0→BUS→IR=B0H
22	25	018095			PC 指向转移地址存放单元	PC→AR=07H，PC+1=08H
23	07	018007			分支检测 PC=1?	P(2)检测 PC=1?
24	01	018001			PC=0，顺序执行	PC=0，PC 内容不变
25	02	018082	08		执行第 6 条指令（ADD R1，R2）	PC→AR=07H 指向指令地址，PC+1=08H
26	14	000048		C0	取指令，将 RAM 中的指令送指令寄存器	AR=07H，RAM=C0H→BUS→IR=C0H
27	08	01A208			取源操作数→DR0	R2→BUS→DR0=00H
28	31	01B419			取目的操作数→DR1	R1→BUS→DR1=01H
29	01	019A01			R2+R1→R2	DR0+DR1→BUS→R2=01H
30	02	018082	09		执行第 7 条指令（INC R1）	PC→AR=088 指向指令地址，PC+1=098
31	15	000048		D1	取指令，将 RAM 中的指令送指令寄存器	AR=088，RAM=11H→BUS→IR=11H
32	36	01A416			取源操作数→DR0	R1→BUS→DR0=01H
33	32	C0841A			取目的操作数→DR1	1→DR1
34	01	019841			R1+1→R1	DR0+DR1→BUS→R1=02H
35	02	018082	0B		执行第 8 条指令（INC R1）	PC→AR=0A8 指向指令地址，PC+1=0B8
36	15	000048		D1	取指令，将 RAM 中的指令送指令寄存器	AR=0A8，RAM=11H→BUS→IR=11H
37	36	01A416			取源操作数→DR0	R1→BUS→DR0=02H
38	32	C0841A			取目的操作数→DR1	1→DR1
39	01	019841			R1+1→R1	DR0+DR1→BUS→R1=03H
40	01	018082	0C		执行第 9 条指令（JMP LP1）	PC→AR=0B8 指向指令地址，PC+1=0C8
41	16	000048		B0	取指令，将 RAM 中的指令送指令寄存器	AR=0B8，RAM=E0H→BUS→IR=E0H
42	30	018080			指向转移地址	PC→AR=0C8 指向转移地址，PC+1=0D8
43	03	008081			转移地址→PC	RAM[0C8]=05H→BUS→PC=05H
44	02	018082	06		执行第 4 条指令（CMP R0，R1）	PC→AR=058 指向指令地址，PC+1=068
45	12	000048		A1	取指令，将 RAM 中的指令送指令寄存器	AR=058，RAM=A1H→BUS→IR=A1H
46	04	01A204			取源操作数	R0→BUS→DR0=07H
47	05	01B405			取目的操作数	R1→BUS→DR1=03H
48	01	018841			R0-R1，标志位→PC	DR0-DR1=04H，PC=0
49	02	018082	07		执行第 5 条指令（JB LP2）	PC→AR=068 指向指令地址，PC+1=078
50	13	000048		B0	取指令，将 RAM 中的指令送指令寄存器	AR=068，RAM=B0H→BUS→IR=B0H
51	25	018095			PC 指向转移地址存放单元	PC→AR=07H，PC+1=08H
52	07	018007			分支检测 PC=1?	P(3)检测 PC=1?
53	01	018001			PC=0，顺序执行	PC=0，PC 内容不变
54	02	018082	08		执行第 6 条指令（ADD R1，R2）	PC→AR=078 指向指令地址，PC+1=088
55	14	000048		C0	取指令，将 RAM 中的指令送指令寄存器	AR=07H，RAM=C0H→BUS→IR=C0H
56	08	01A208			取源操作数→DR0	R2→BUS→DR0=03H
57	31	01B419			取目的操作数→DR1	R1→BUS→DR1=03H
58	01	019A01			R2+R1→R2	DR0+DR1→BUS→R2=06H
59	02	...				

实验与设计

5-4. 较复杂CPU应用程序设计实验

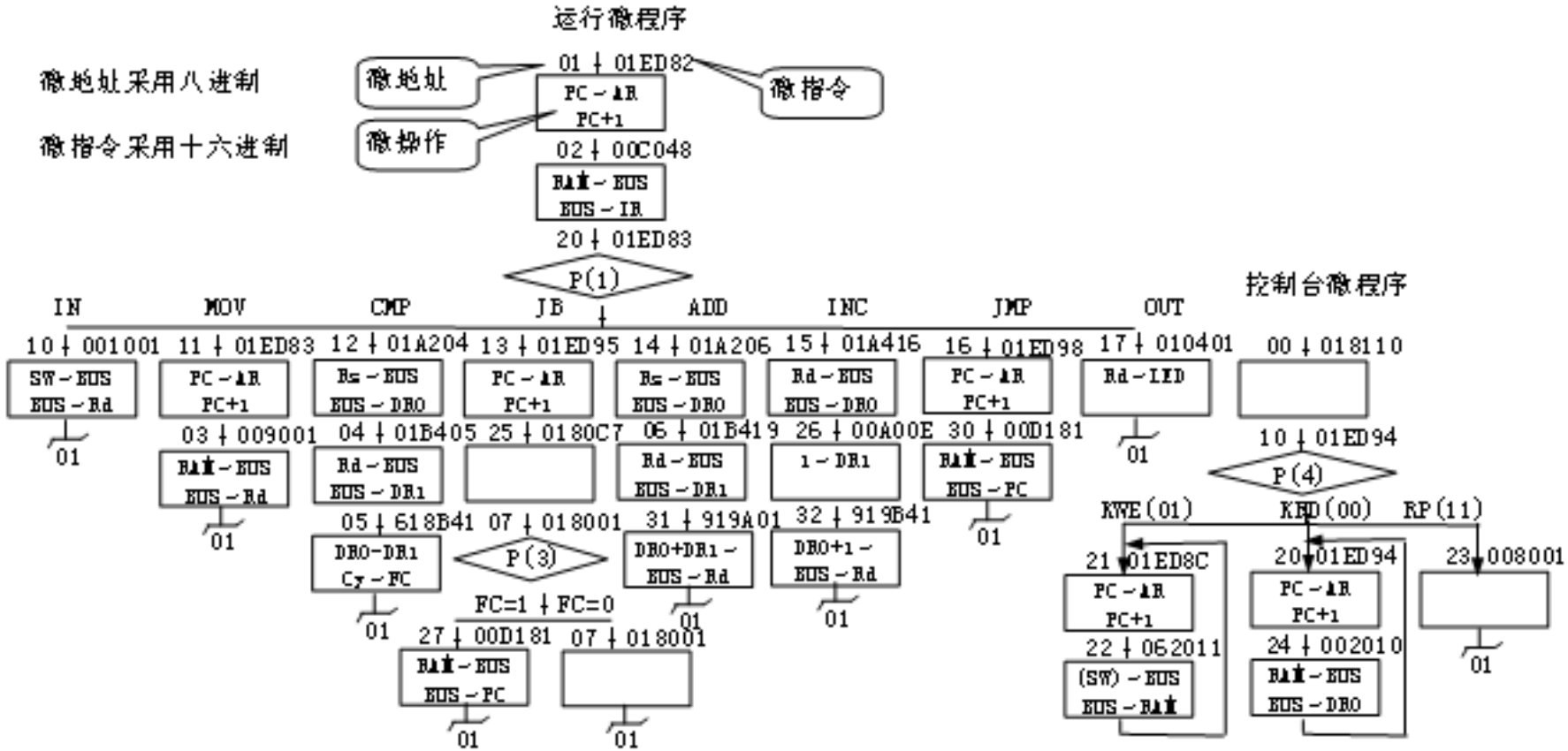


图 5-36 实验 5-4 的微程序流程图

实验与设计

5-4. 较复杂CPU应用程序设计实验

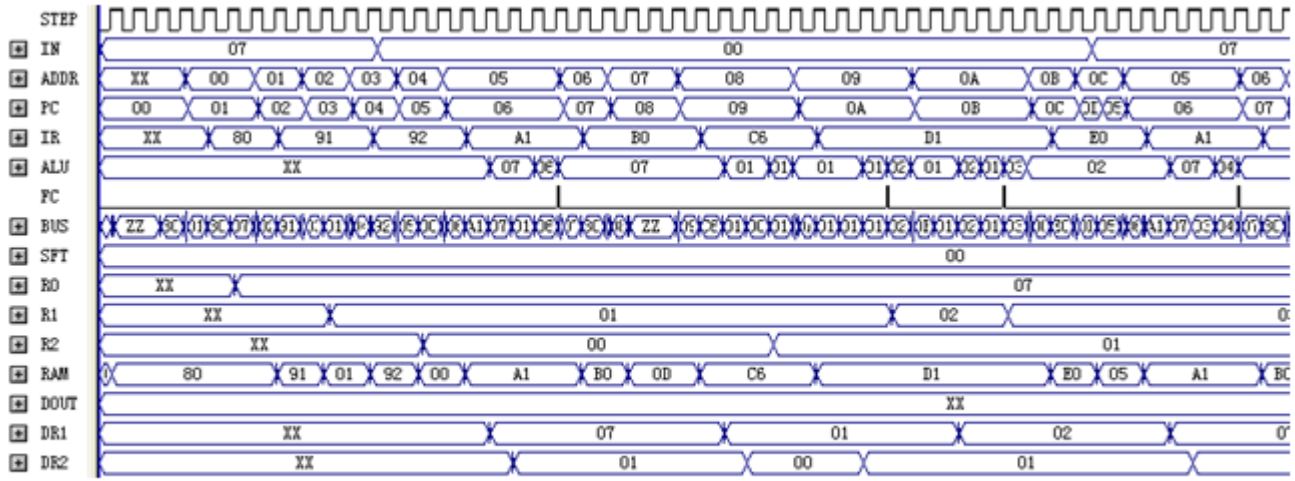


图 5-37 模型机时序仿真波形（截取其中部分）