现代计算机组成原理

潘 明　潘 松 编著

科学出版社

第 5 章

8位CISC计算机设计

5.1 8位CPU结构

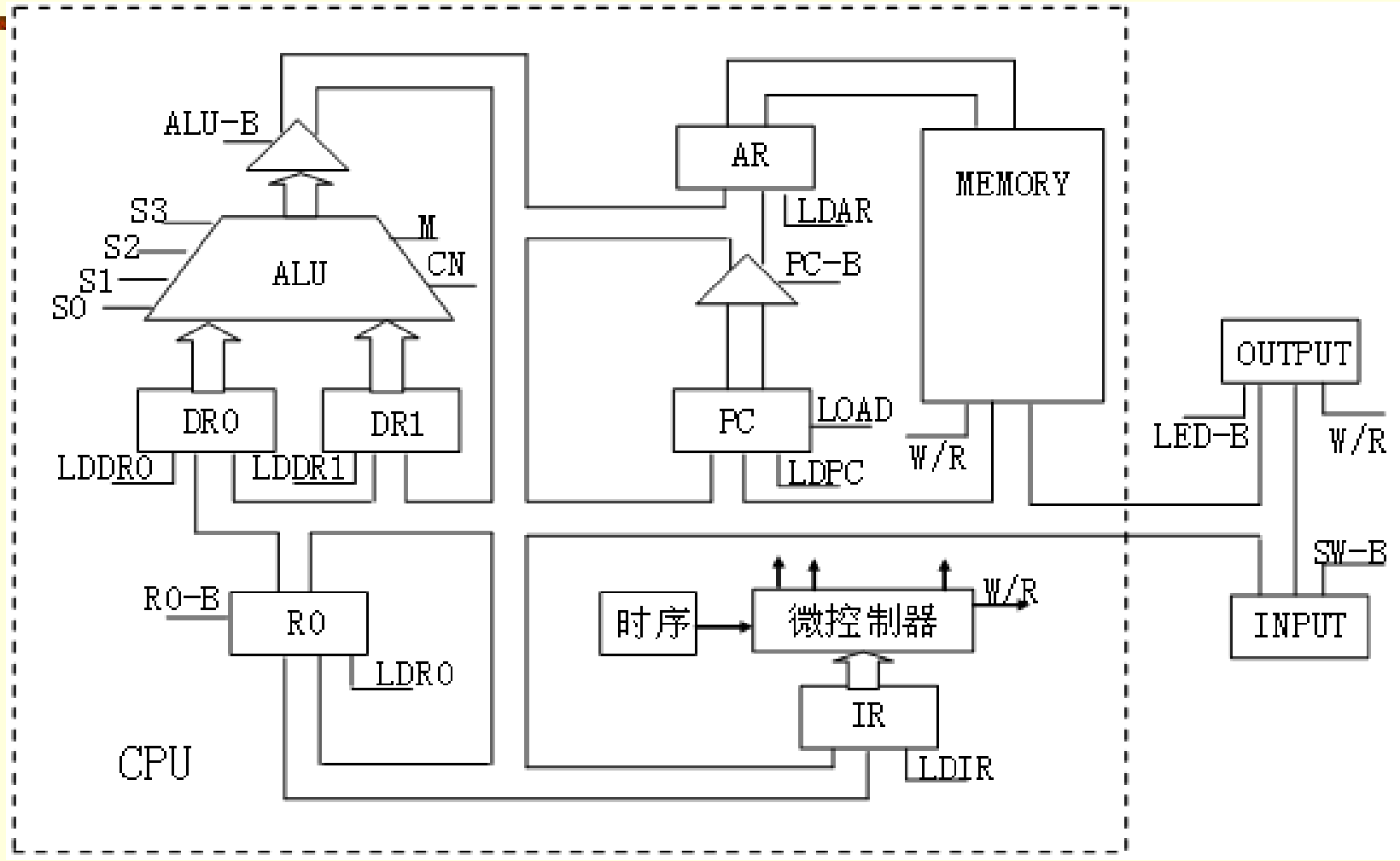


图5-1 8位CPU的结构

5.1 8位CPU结构

1．运算部件
- (1) 输入逻辑。
- (2) 算术/逻辑运算部件ALU。
- (3) 输出逻辑。

2．寄存器组 —— 由R0、R1、R2所组成

3．指令寄存器

4．程序计数器

5．地址寄存器

5.1 8位CPU结构

6．标志寄存器

- 进位位Fc：运算后如果产生进位，将Fc置为1；否则将Fc清为0。
- 零位Fz：运算结果为零，将Fz置为1，否则将Fz清为0。

7．微命令产生部件

8．时序系统

周期、节拍、脉冲等信号称为时序信号

5.2 指令系统的结构及功能的确

5.2.1. 模型机指令系统

一条指令必须包含下列信息

- 操作码
- 操作数的地址
- 操作结果的存储地址
- 下一条指令的地址

表5-1 指令的基本格式

位	7 6 5 4	32	10
功能	OP-CODE	rs	rd

5.2 指令系统的结构及功能的确

5.2.1. 模型机指令系统

表5-2 寄存器操作数

rs或rd	选定的寄存器
00	R0
01	R1
10	R2

5.2 指令系统的结构及功能的确

5.2.1. 模型机指令系统

表5-3模型机指令系统，及其指令编码形式

助记符	机器指令码	Addr地址码	功能说明
IN	0 0H		“INPUT”中的数据→R0
ADD addr	1 0H	XX H	R0+[addr] ->R0
STA addr	2 0H	XX H	R0 -> [addr]
OUT addr	3 0H	XX H	[addr] -> BUS
JMP addr	4 0H	XX H	addr →PC

5.2 指令系统的结构及功能的确

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

微命令和微操作是一一对应的

（2）微指令、微地址

微操作码字段，又称操作控制字段，该字段指出微指令执行的微操作；

微地址码字段，又称顺序控制字段，指出下一条要执行的微指令的地址。

（3）微周期

（4）微程序

5.2 指令系统的结构及功能的确

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

- 微操作码字段，又称操作控制字段，该字段指出微指令执行的微操作；
- 微地址码字段，又称顺序控制字段，指出下一条要执行的微指令的地址。

（3）微周期

（4）微程序

5.2 指令系统的结构及功能的确

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

控制字段	判别测试字段	下址字段

（2）垂直型微指令

（3）水平型微指令与垂直型微指令的比较

水平型微指令 VS **垂直型微指令**

- 并行操作能力强，效率高，灵活性强
- 执行一条指令的时间短
- 解释指令的微程序，微指令字比较长，但微程序短
- 难度较大

- 较差
- 执行时间长
- 微指令字比较短而微程序长
- 机器指令比较相似，相对容易设计

5.2 指令系统的结构及功能的确

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					

5.2 指令系统的结构及功能的确

5.2.2. 拟定指令流程和微命令序列

3. 模型机的微指令

表5-5　　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

5.2 指令系统的结构及功能的确

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

5.2 指令系统的结构及功能的确

5.2.3微程序设计

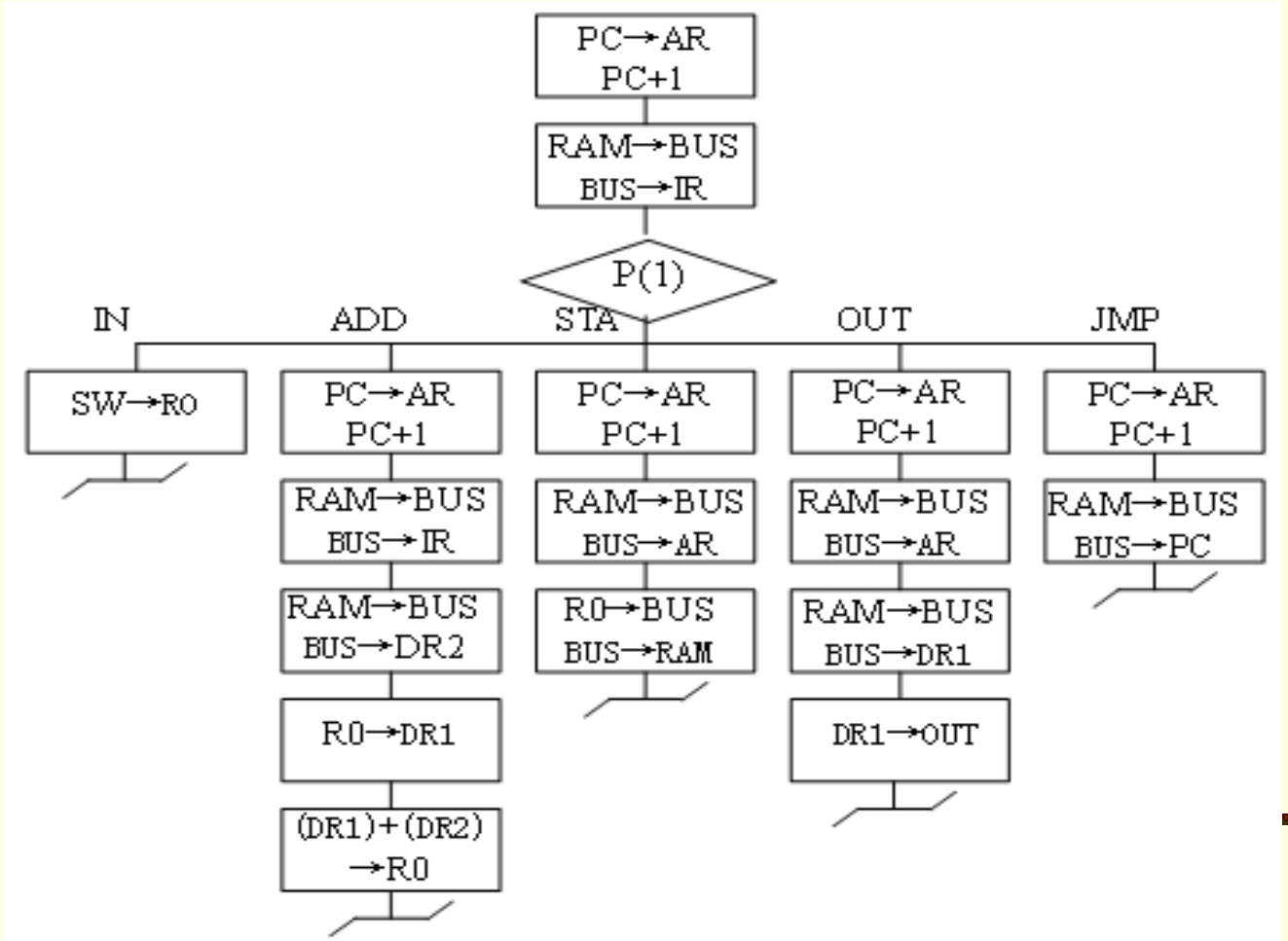


图5-2 微程序流程图

5.2 指令系统的结构及功能的确

5.2.3微程序设计

1. IN指令　BUS←SW ；　R0←BUS　　R0←SW

2. ADD指令　R0←R0+（MEM）

AR←PC，PC←PC+1 ；以AR的内容作为取操作数的地址

BUS←RAM，AR←BUS ；AR指向存放操作数的RAM单元

BUS←RAM，DR2←BUS ；RAM中的数据通过BUS送DR2

DR1←R0

R0←（DR1）+（DR2）

5.2 指令系统的结构及功能的确

5.2.3微程序设计

3．STA指令

AR←PC，PC←PC+1　　；以PC的内容作为存数据的地址

BUS←RAM，AR←BUS　；AR指向存放操作数的RAM单元

BUS←R0，RAM←BUS

4．OUT指令

AR←PC，PC←PC+1　　；以PC的内容作为存数据的地址

BUS←RAM，AR←BUS　；AR指向存放操作数的RAM单元

BUS←RAM，DR1←BUS

OUT←DR1

5．JMP指令

AR←PC，PC←PC+1　　；以PC的内容作为取数据的地址

BUS←RAM，PC←BUS　；将RAM内容送PC，实现程序转移

5.3 8位CPU的硬件系统设计

（1）用图形编辑工具设计模型CPU的顶层电路原理图。

（2）根据微程序的微操作，对于所需的控制信号，确定微指令，并确定微地址。

（3）微程序流程图按微指令格式转化为“二进制微代码表”。

（4）设计控制存储器LPM_ROM。

（5）对模型CPU的整机硬件电路进行编译、波形仿真和调试。

（6）根据仿真波形，查找故障原因，排除故障，重新编译。

（7）将编译通过的电路和应用程序下载到实验台上的FPGA中，在实验台上单步跟踪微程序的执行过程。

（8）最终完成模型CPU的硬件电路设计和应用程序及微程序的设计和调试。

5.3.1 CPU顶层设计

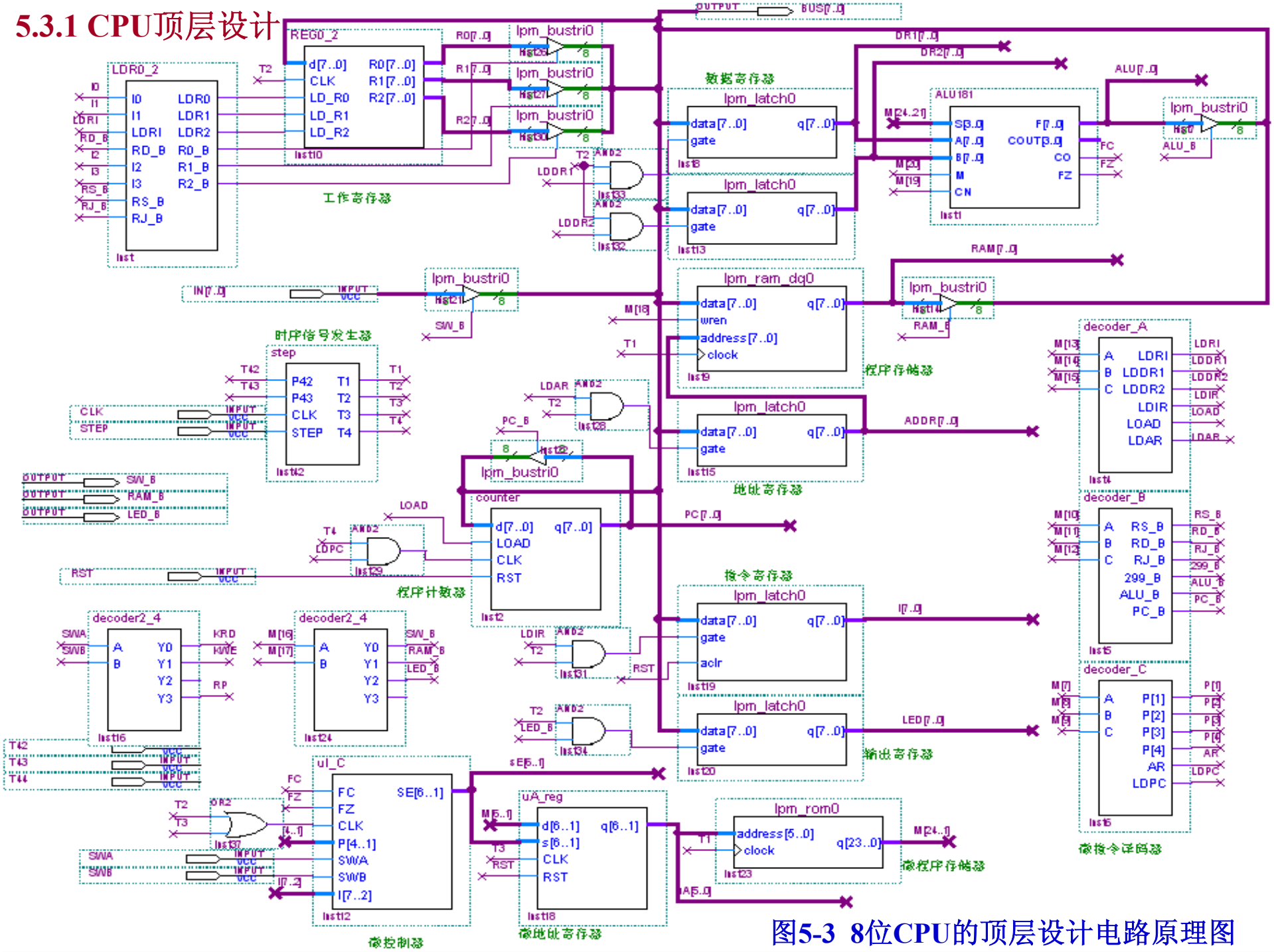
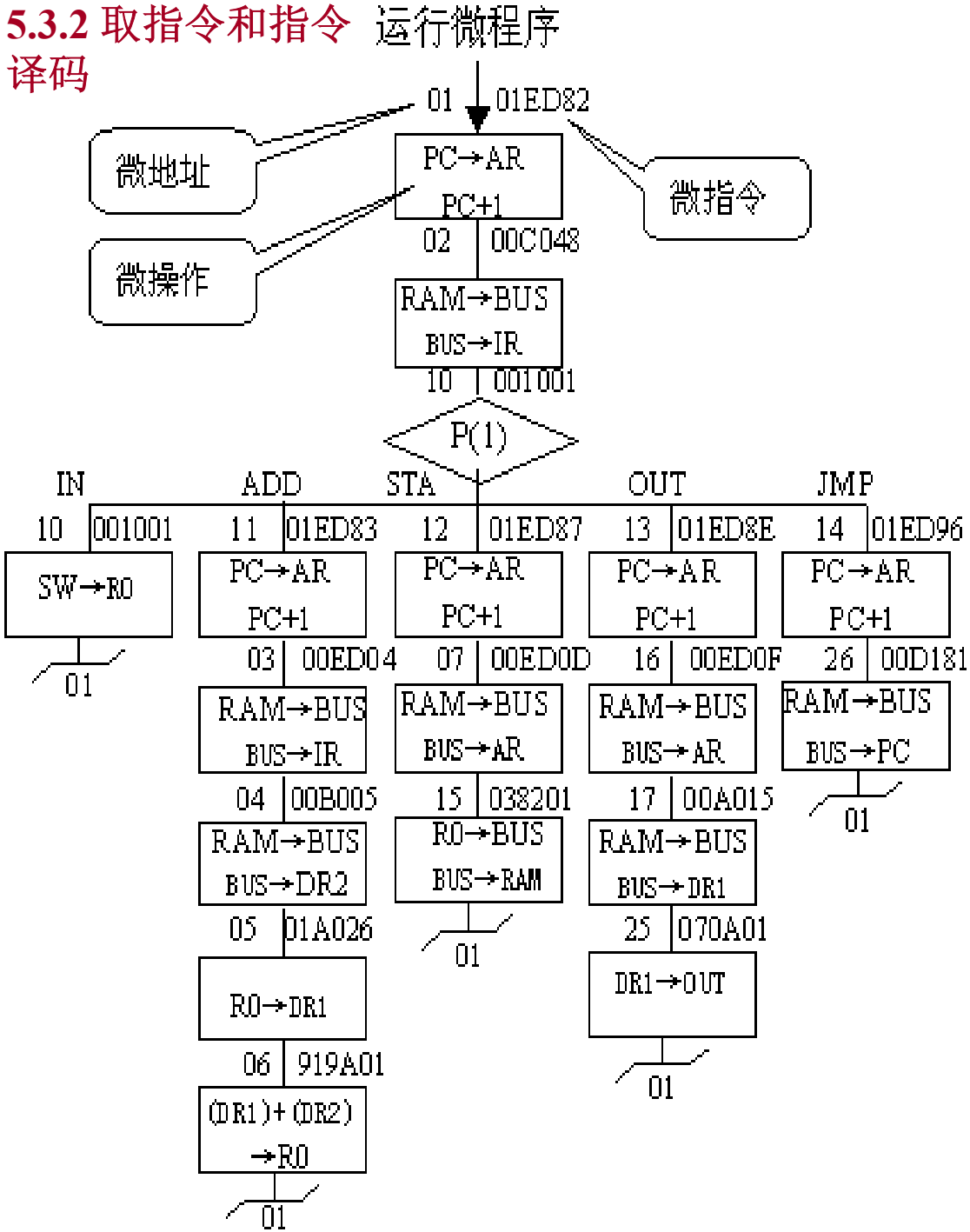


图5-3 8位CPU的顶层设计电路原理图

5.3.2 取指令和指令译码



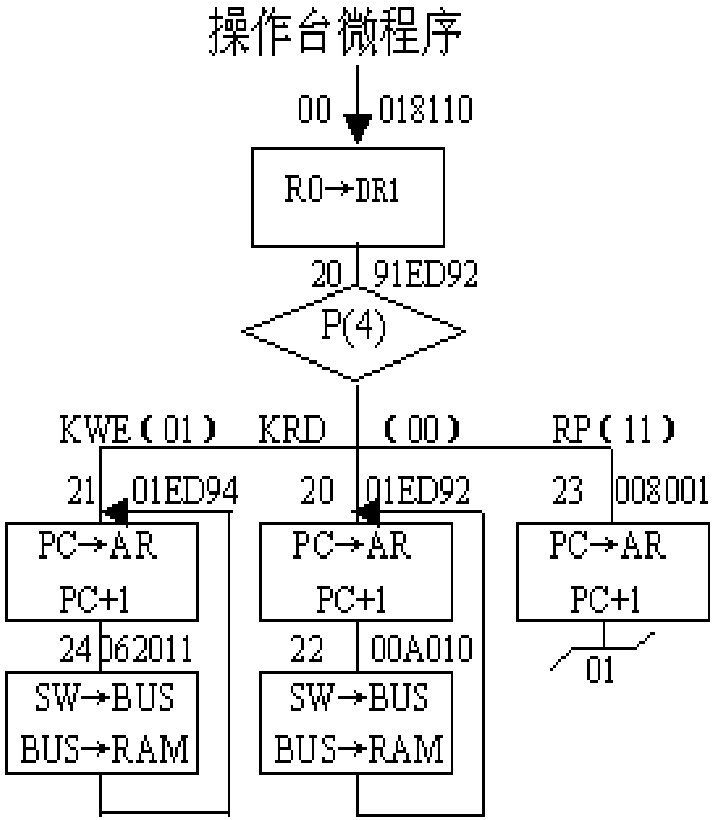


图5-4 微指令流程图

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
00	018110	0	0	0	0	0	0	0	1	1	000	000	100	010000
01	01ED82	0	0	0	0	0	0	0	1	1	110	110	110	000010
02	00C048	0	0	0	0	0	0	0	0	1	100	000	001	001000
03	00E004	0	0	0	0	0	0	0	0	1	110	000	000	000100
04	00B005	0	0	0	0	0	0	0	0	1	011	000	000	000101
05	01A206	0	0	0	0	0	0	0	1	1	010	010	000	000110
06	919A01	1	0	0	1	0	0	0	1	1	001	101	000	000001
07	00E00D	0	0	0	0	0	0	0	0	1	110	000	000	001101
10	001001	0	0	0	0	0	0	0	0	0	001	000	000	000001
11	01ED83	0	0	0	0	0	0	0	1	1	110	110	110	000011
12	01ED87	0	0	0	0	0	0	0	1	1	110	110	110	000111
13	01ED8E	0	0	0	0	0	0	0	1	1	110	110	110	001110
14	01ED96	0	0	0	0	0	0	0	1	1	110	110	110	010110
15	038201	0	0	0	0	0	0	0	1	1	000	001	000	000001
16	00E00F	0	0	0	0	0	0	0	0	1	110	000	000	001111
17	00A015	0	0	0	0	0	0	0	0	1	010	000	000	010101
20	01ED92	0	0	0	0	0	0	0	1	1	110	110	110	010010
21	01ED94	0	0	0	0	0	0	0	1	1	110	110	110	010100
22	01ED94	0	0	0	0	0	0	0	1	1	010	000	000	010000
23	018001	0	0	0	0	0	0	0	1	1	000	000	000	000001
24	062011	0	0	0	0	0	1	1	0	0	010	000	000	010001
25	010A01	0	0	0	0	0	0	0	1	0	000	101	000	000001
26	00D181	0	0	0	0	0	0	0	0	1	101	000	110	000001

5.3.4 建立数据通路

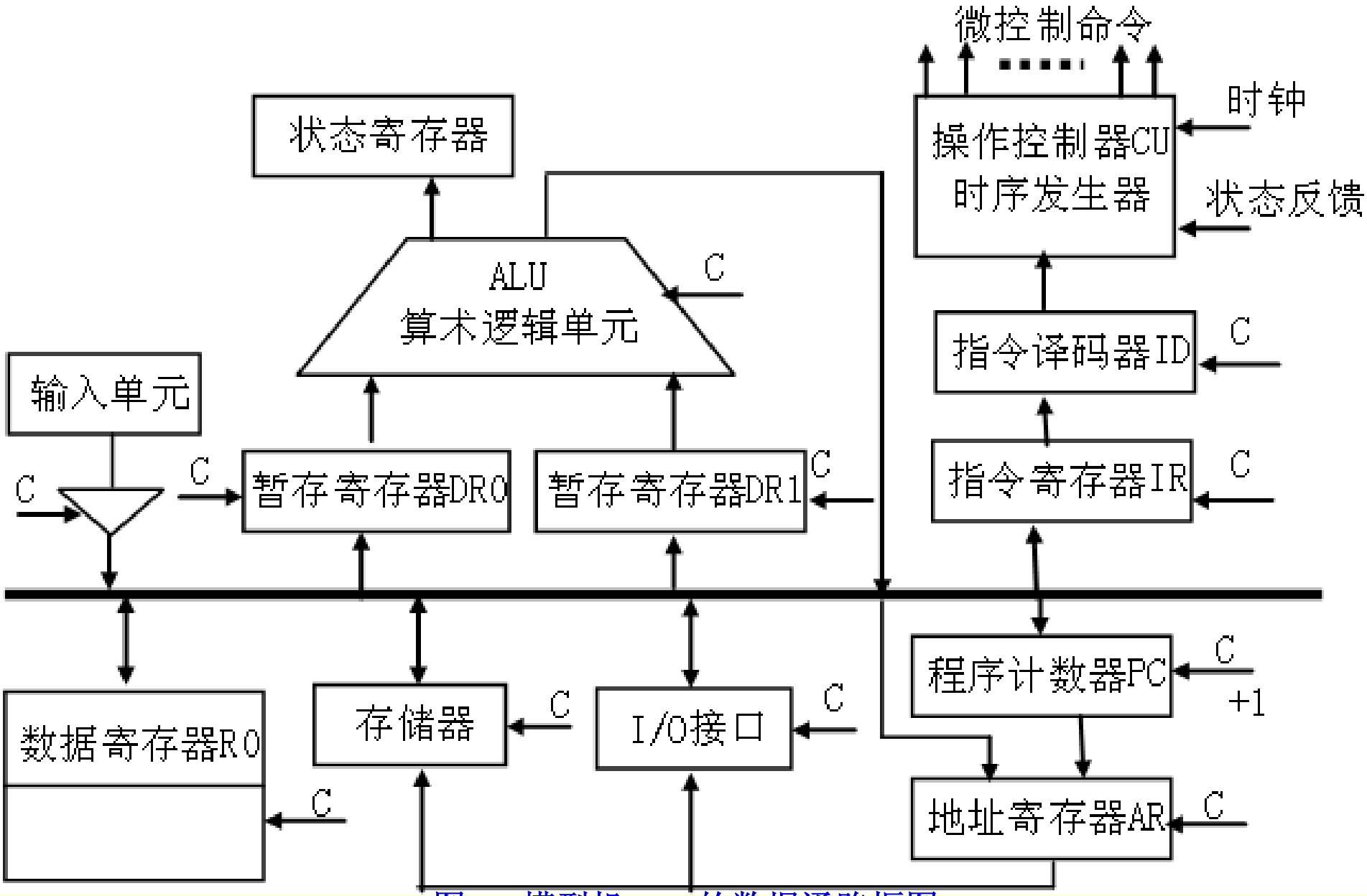


图5-5 模型机CPU的数据通路框图

5.3 8位CPU的硬件系统设计

5.3.5 运算器ALU的设计

5.3.6 控制执行单元

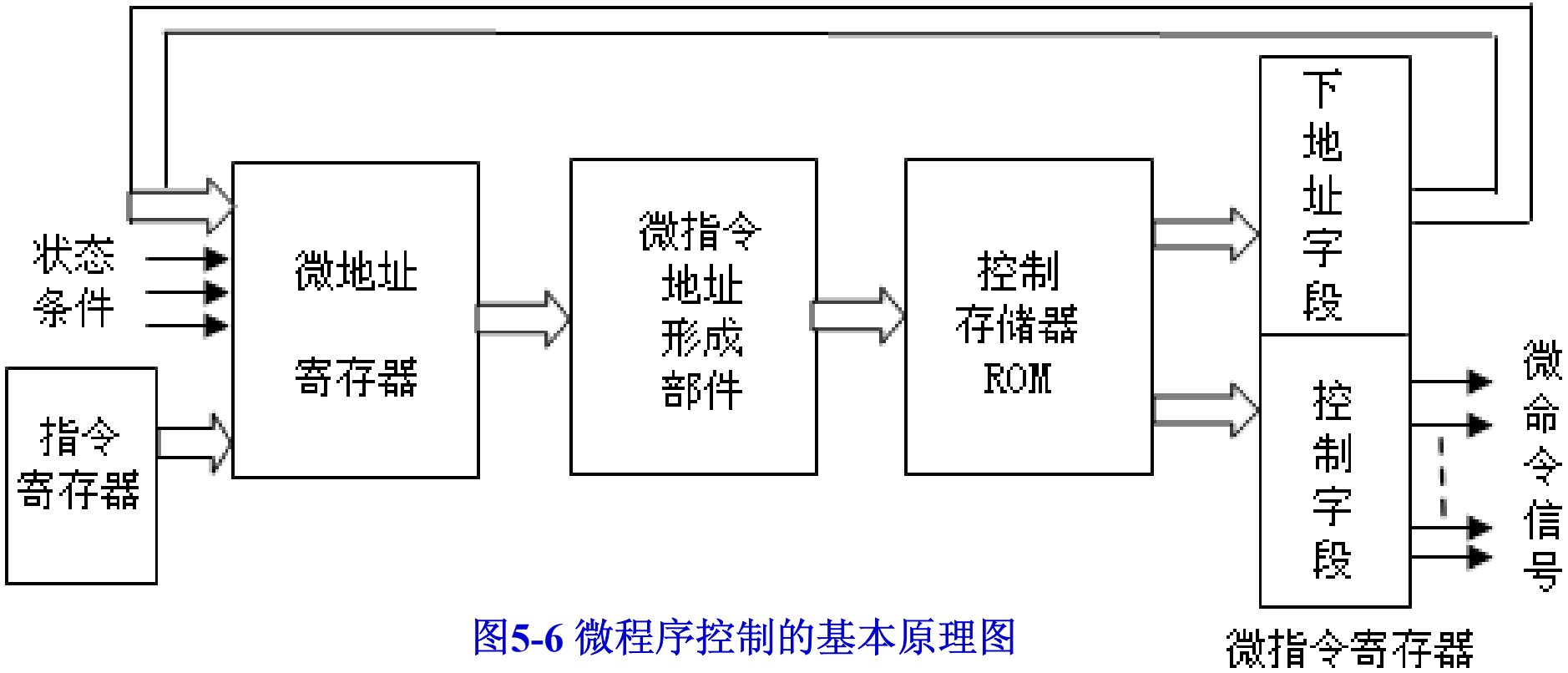


图5-6 微程序控制的基本原理图

5.3 8位CPU的硬件系统设计

5.3.7 在模型CPU中的软件执行

表5-7 示例程序 模型机的指令及编码形式

地址（16进制）	内容（16进制）	助记符	说明
00	00	IN	“INPUT”→ R0，键盘输入数据
01	10	ADD [0AH]	[R0]+[0AH] → R0
02	0A		
03	20	STA [0BH]	[R0] → [0BH]
04	0B		
05	30	OUT [0BH]	[0BH] “OUTPUT”，显示输出数据
06	0B		
07	40	JMP [08H]	[09H] → PC ，以[08H]内容为转移地址
08	00		
09	00		
0A	34	DB 34H	被加数（自定）
0B	XX		求和结果

5.3.7 在模型CPU中的软件执行

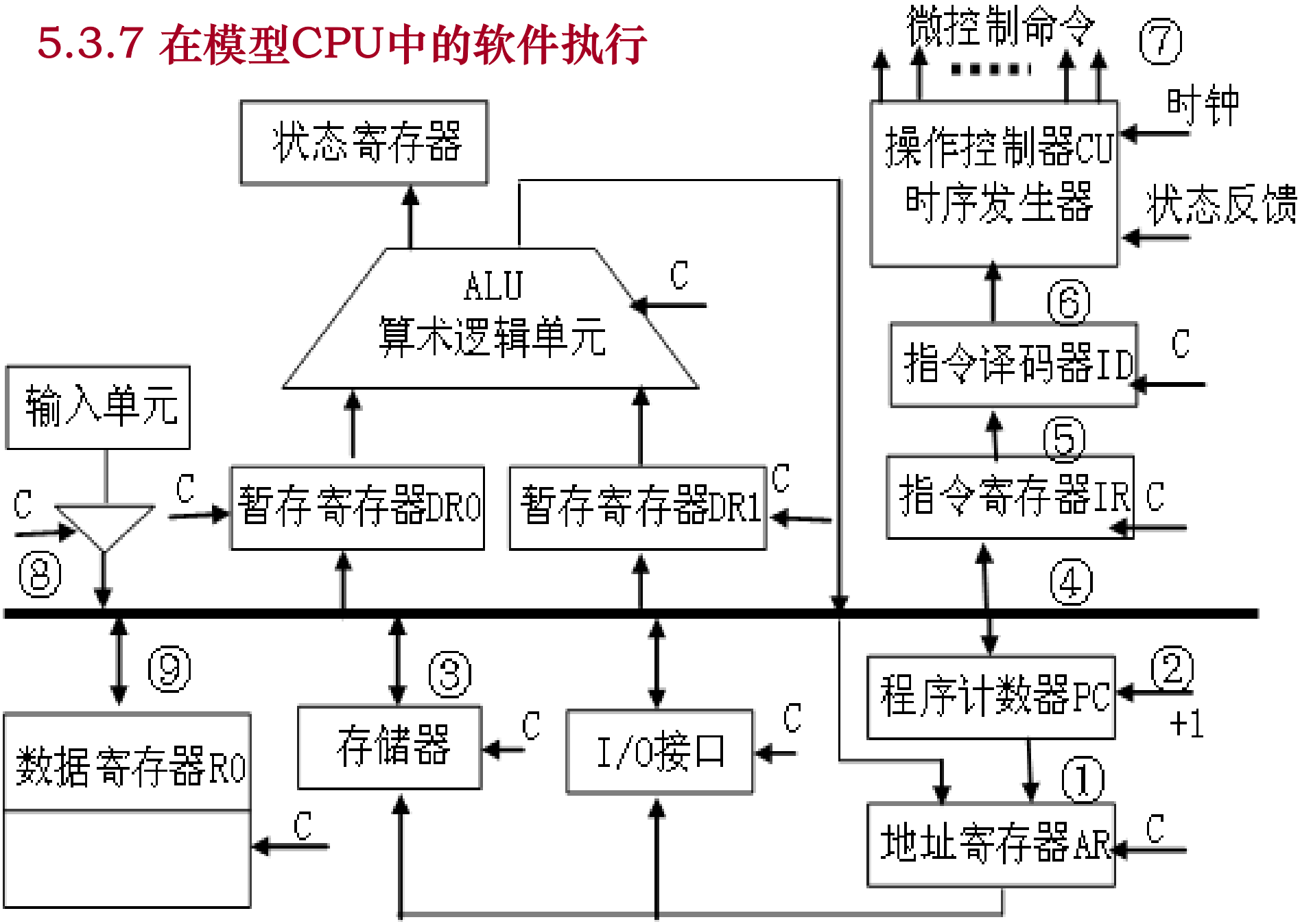


图5-7　输入指令IN的执行过程

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	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, AR 指向指令地址
5	10	00C048		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	02		接收键 1、2 输入的数据，送 R0 寄存器	R0=56H，键 1、键 2 输入数据 56H
7	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
8	09	00C048		10	取指令	RAM(01H)=10H→BUS→IR=10H
9	03	01ED83	03		间接寻址，AR 指向取数的间接地址	PC→AR=02H, PC+1=03H，RAM=10H
10	04	00E004			取数地址送 AR	RAM(02)=0AH→BUS→AR=0AH
11	05	00B005			从 RAM 中取数送 DR2	RAM(0AH)=34H→BUS→DR2=34H
12	06	01A206			将 R0 的数据送 DR1	(R0)=56H→BUS→DR1=56H
13	01	919A01			加法运算：(DR1)+(DR2)→R0	56H+34H=8AH→R0=8AH
14	02	01ED82	04		执行第 3 条指令（存储 STA）	PC→AR=03H，PC+1=04H
15	12	00C048		20	取指令	RAM(03H)=20H→BUS→IR=20H
16	07	01ED87	05		间接寻址，AR 指向存数的间接地址	PC→AR=04H, PC+1=05H
17	15	00E00D			存数的地址送 AR	RAM(04)=0BH→BUS→AR=0BH
18	01	038201			R0 的内容存入 RAM(0BH)单元	(R0)=8AH→BUS→RAM(0BH)=8AH
19	02	01ED82	06		执行第 4 条指令（输出 OUT）	PC→AR=05H，PC+1=06H
20	13	00C048		30	取指令	RAM(05H)=30H→BUS→IR=30H
21	16	01ED8E	07		间接寻址，AR 指向取数的间接地址	PC→AR=06H，PC+1=07H
22	17	00E00F			取数地址送 AR	RAM(06)=0BH→BUS→AR=0BH
23	25	00A015			从 RAM 中取数送 DR1	RAM(0BH)=8AH→BUS→DR1=8AH
24	01	010A01			DR1 的内容送输出单元 OUT	DR1=8AHH→BUS→OUT=8AH
25	02	01ED82	08		执行第 5 条指令（转移 JMP）	PC→AR=07H，PC+1=08H
26	14	00C048		40	取指令	RAM(07H)=40H→BUS→IR=40H
27	26	01ED96	09	40	间接寻址，AR 指向转移的间接地址	PC→AR=08H，PC+1=09H
28	01	00D181	00		转移地址送 PC，转到 00H。	RAM(08H)=00→BUS→PC=00H
29	02	01ED82	01		执行第 1 条指令——程序循环	PC→AR=00H，PC+1=01H
30	10	00C048	01	00	取指令	
ooo						

表5-8 微指令执行情况

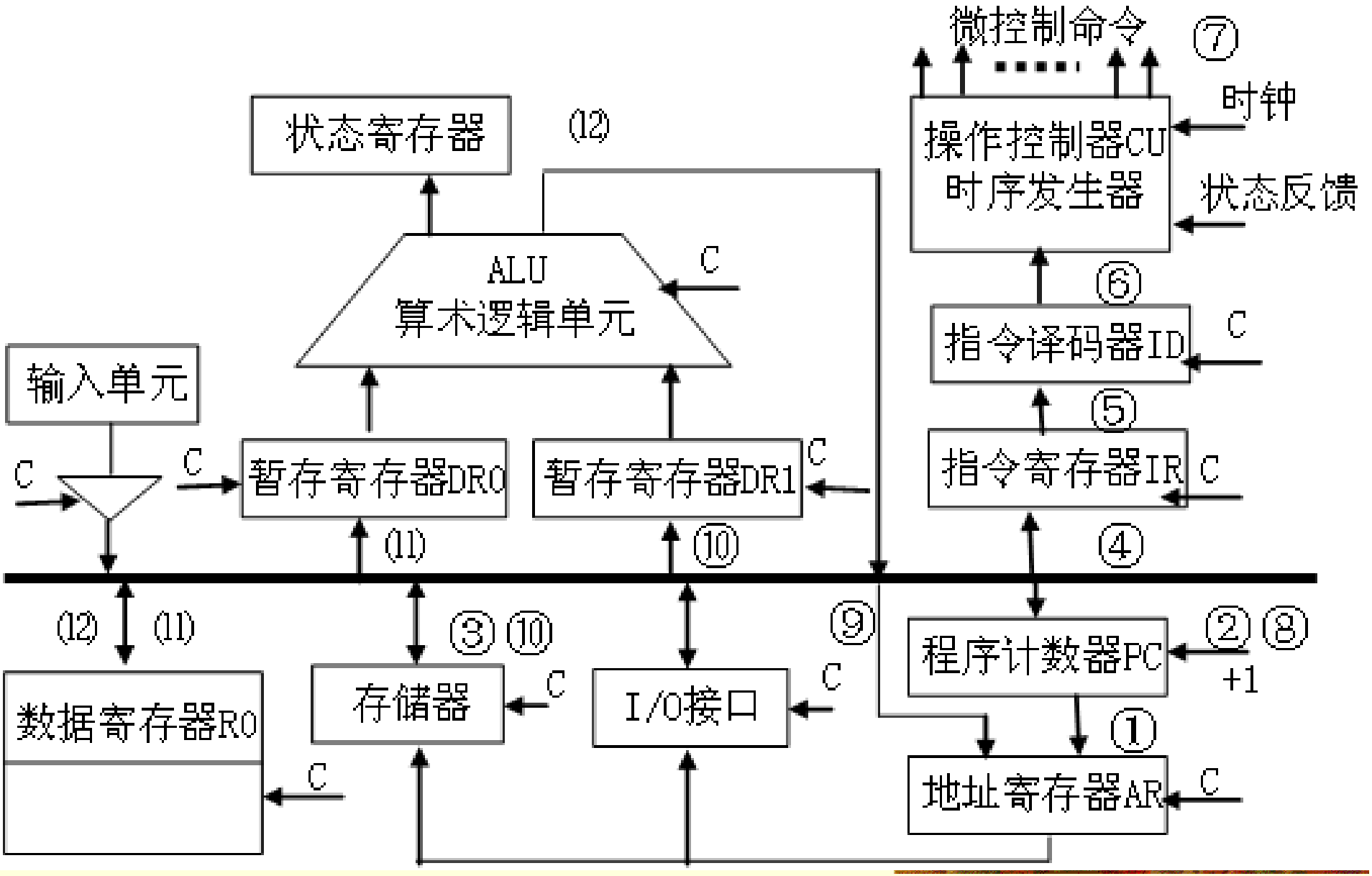


图5-8 加法指令ADD的执行过程

5.3.8 模型CPU的硬件仿真

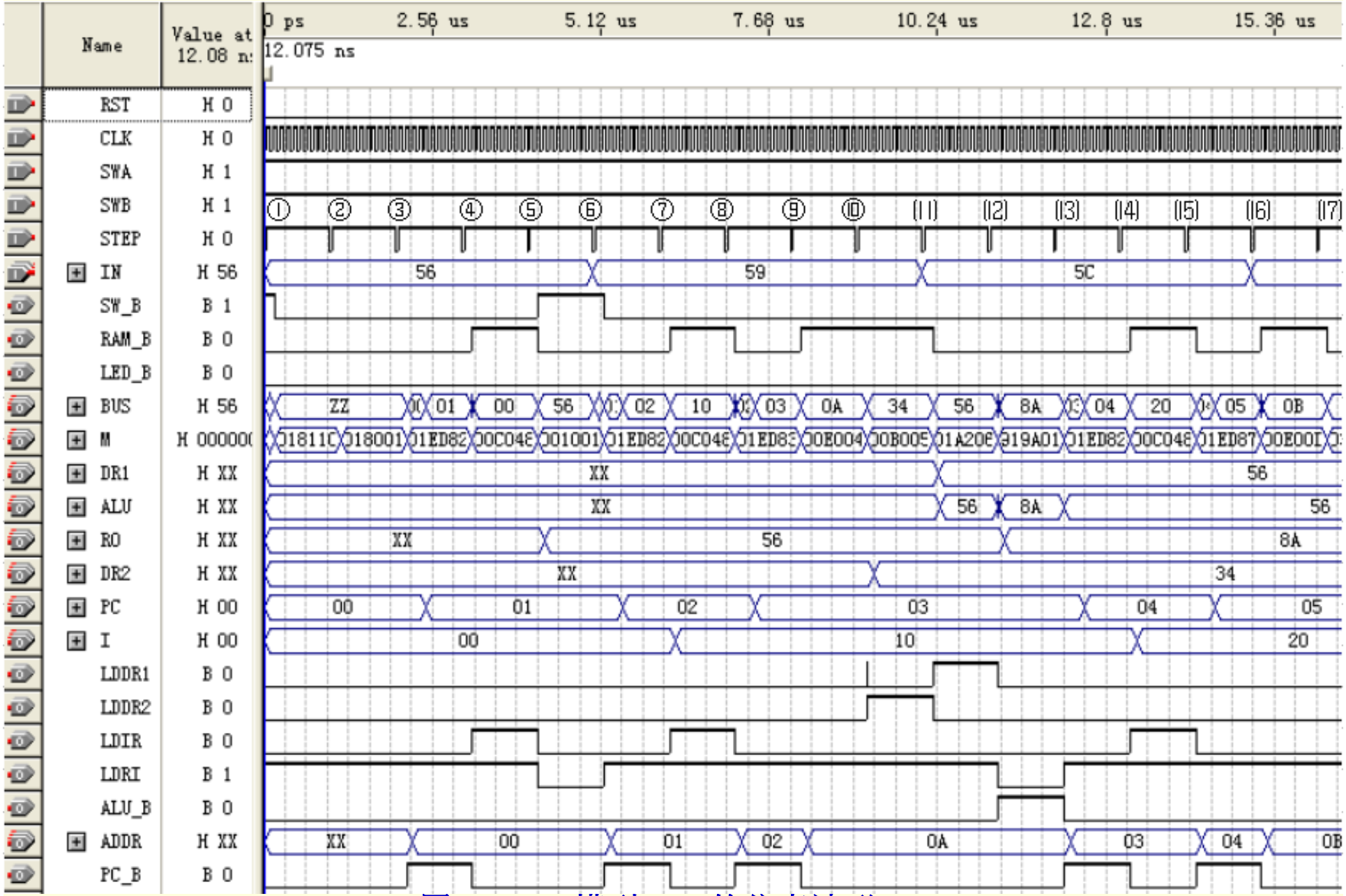


图5-9　　模型CPU的仿真波形

5.3.8 模型CPU的硬件仿真

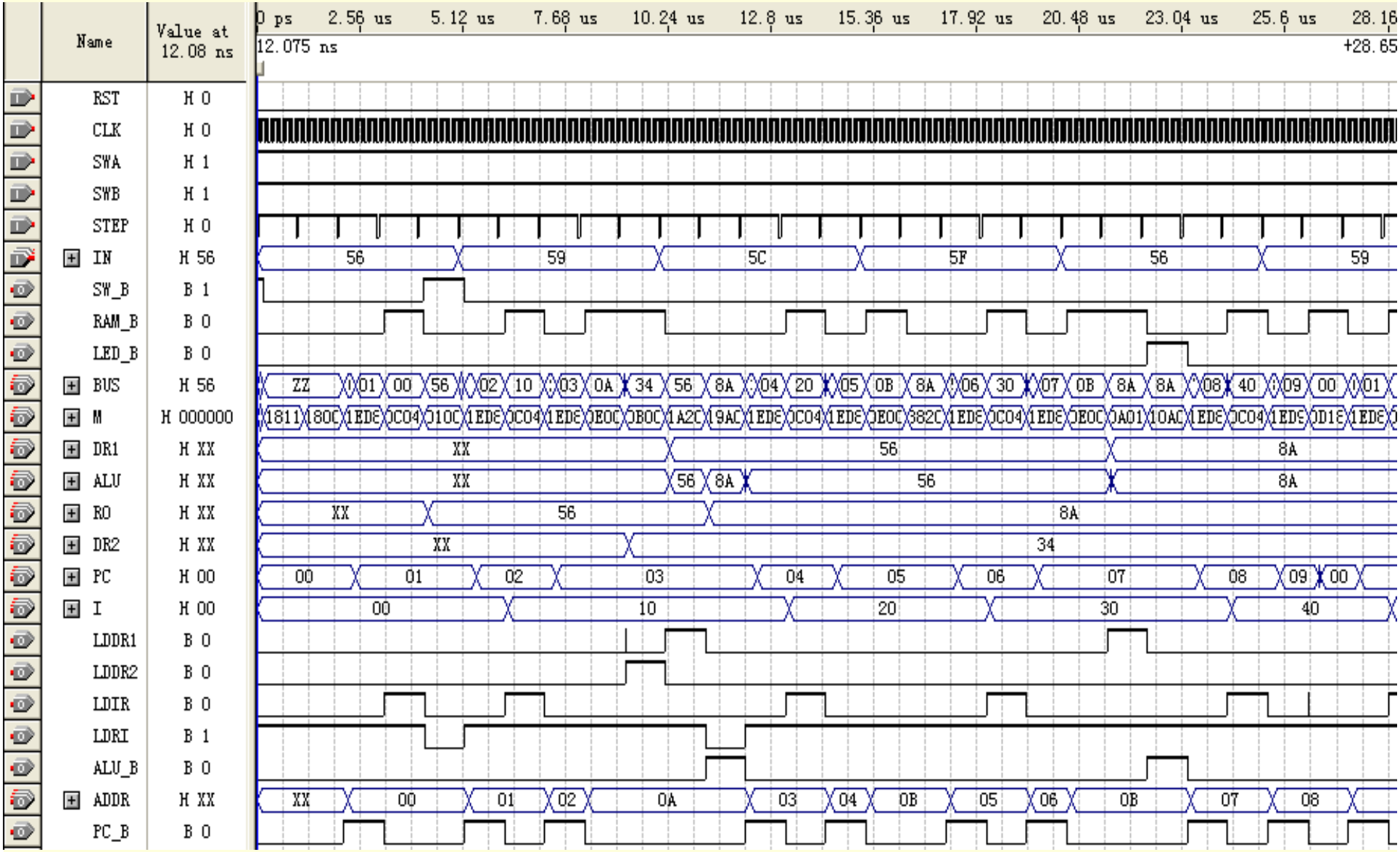


图5-10　　执行一个周期循环程序的全部仿真波形

5.3 8位CPU的硬件系统设计

5.3.8 模型CPU的硬件仿真

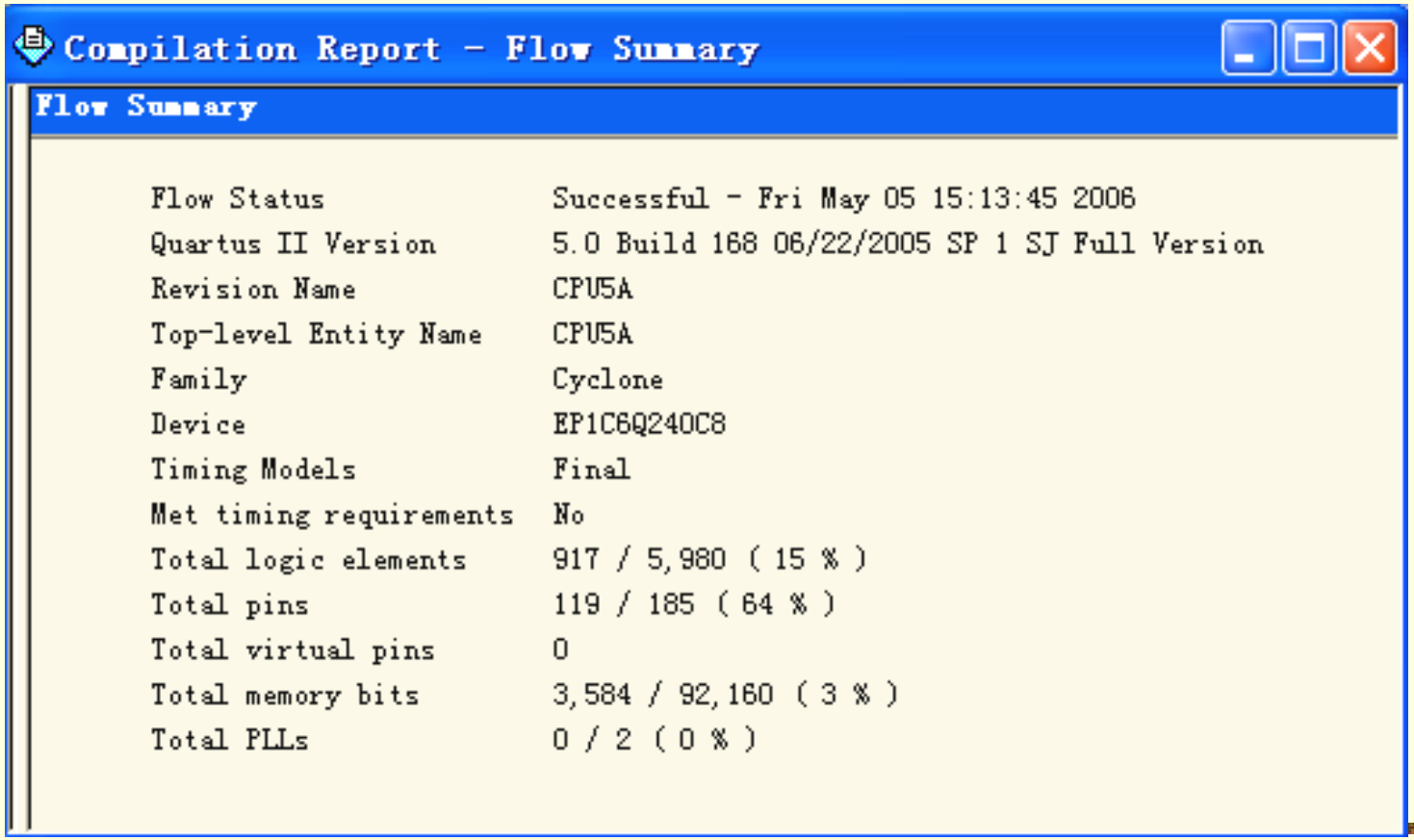


图5-11 模型CPU使用FPGA资源报告

5.4 具有移位功能的CPU设计

5.4.1 移位运算器的VHDL设计

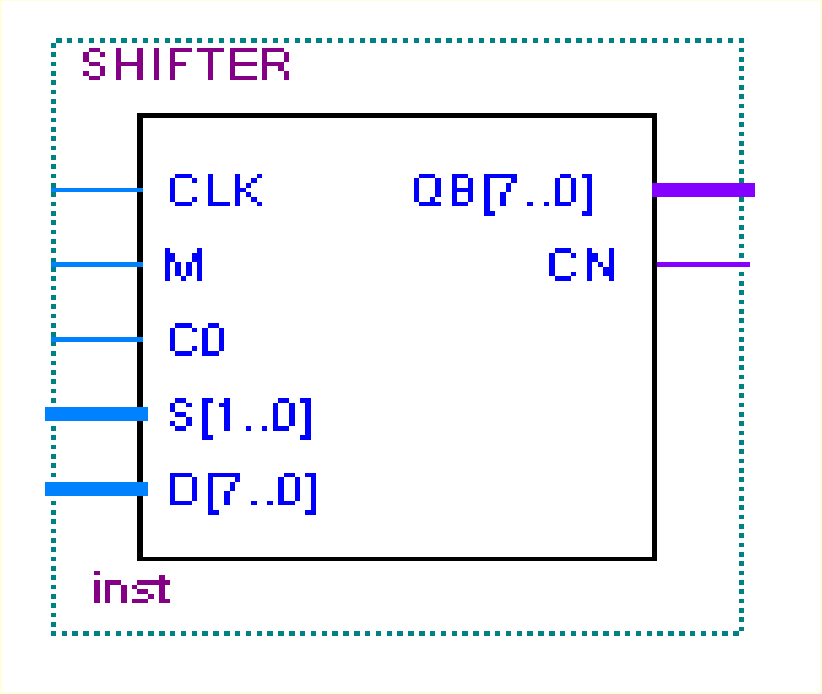


图5-12 移位运算器实体结构

5.4 具有移位功能的CPU设计

5.4.1 移位运算器的VHDL设计

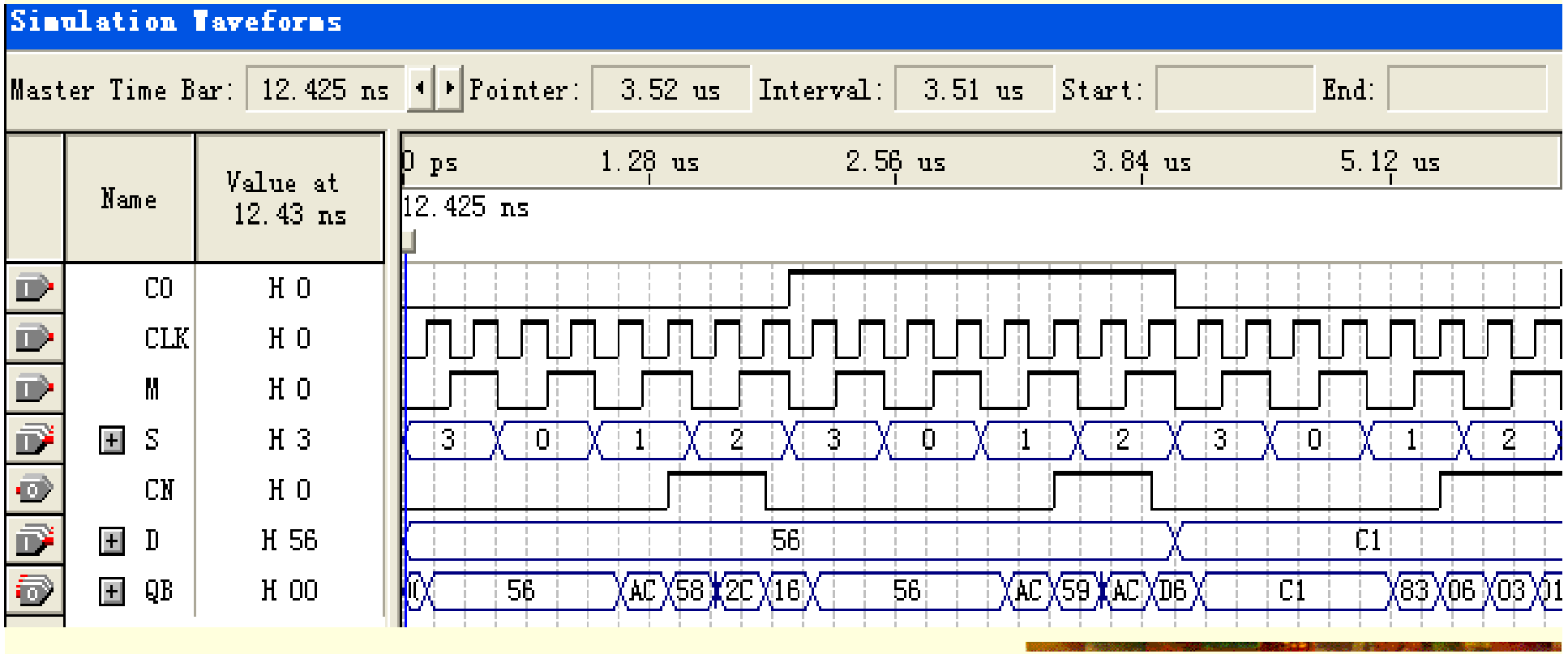


图5-13 移位运算器的仿真波形

5.4 具有移位功能的CPU设计

5.4.2 移位运算器与ALU的结合设计

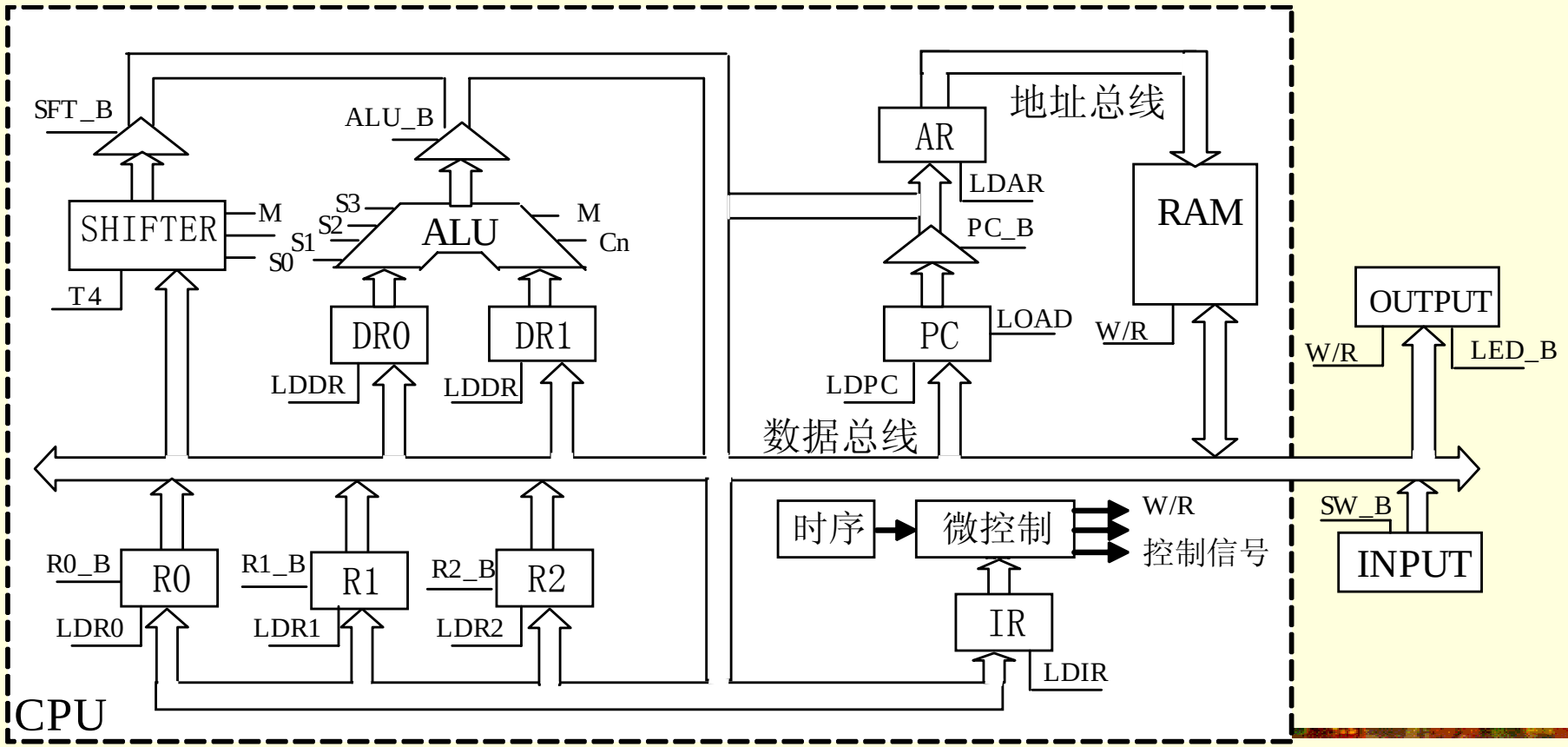


图5-14 带移位运算的CPU数据通路框图

5.4 具有移位功能的CPU设计

5.4.2 移位运算器与ALU的结合设计

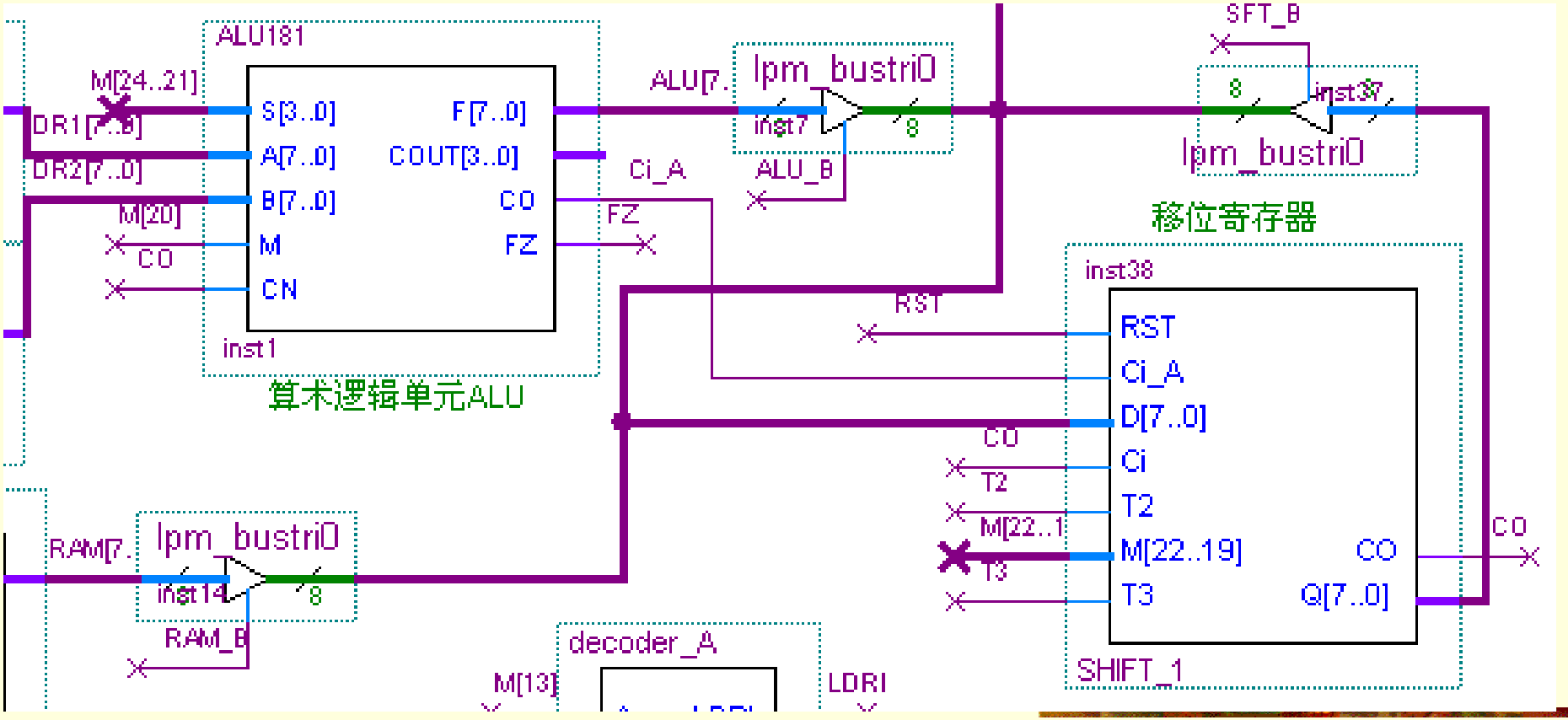
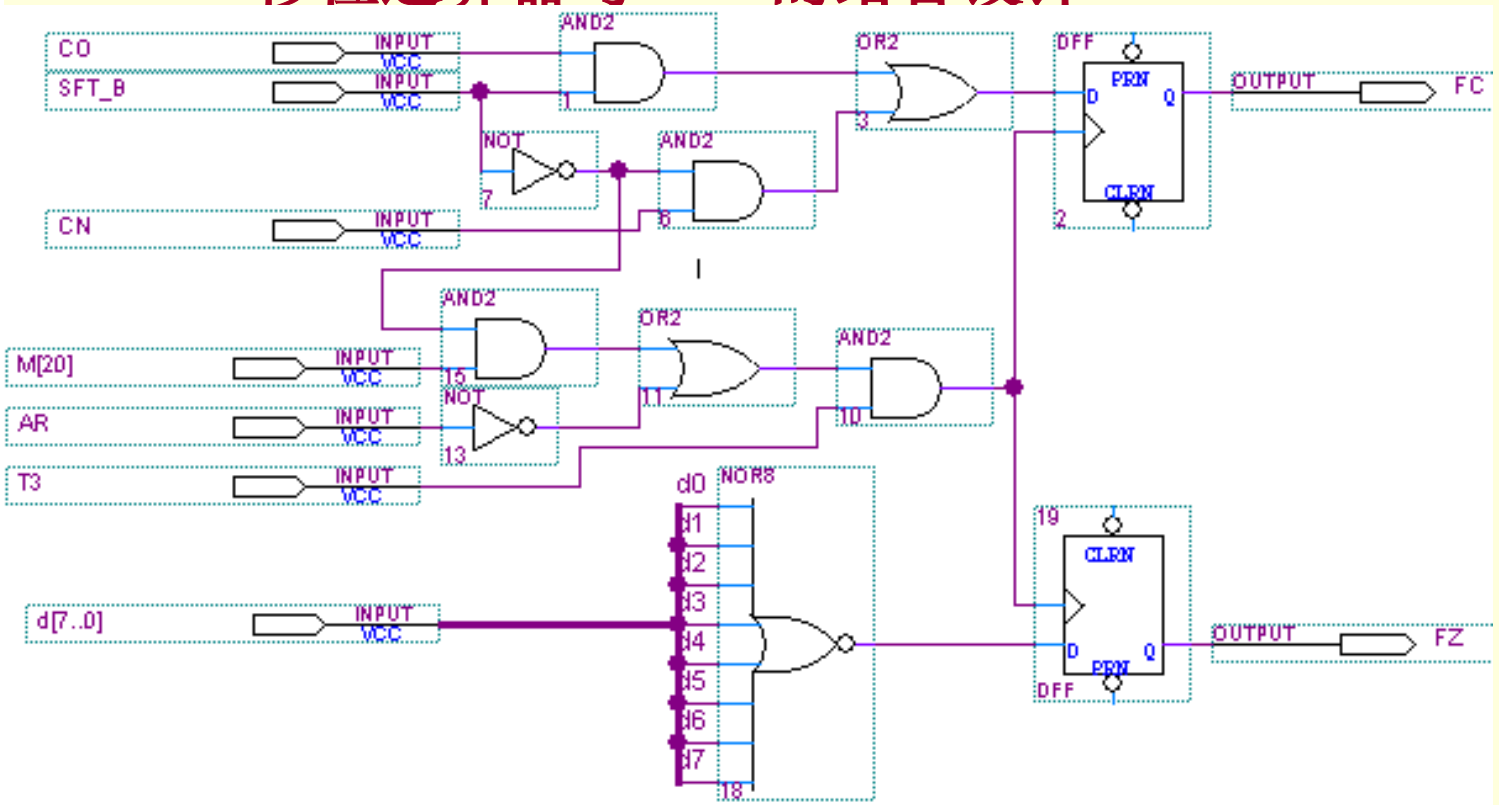


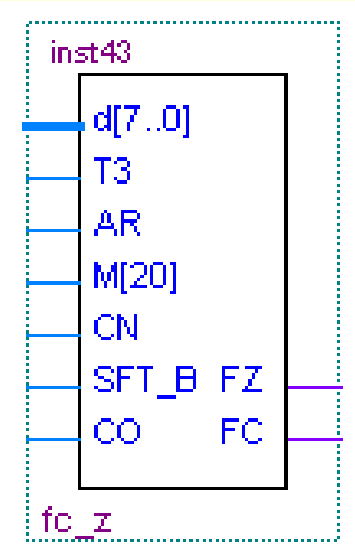
图5-15 移位寄存器的辅助电路

5.4 具有移位功能的CPU设计

5.4.2 移位运算器与ALU的结合设计



（a）Fc、Fz控制电路内部结构



（b）Fc、Fz组合后的模块

图5-16　Fc、Fz组合控制电路

5.5 含更多指令的CPU模型机设计

5.5.1 指令系统的格式

1. 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

7 6 5 4	3 2	1 0
OP-CODE	rs	Rd

Rs或rd	选定的寄存器
00 01 10	R0 R1 R2

5.5 含更多指令的CPU模型机设计

（2）访问指令及转移指令

76	54	32	10
00	M	OP-CODE	Rd
D			

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

5.5 含更多指令的CPU模型机设计

（3）I/O指令

7654	3 2	1 0
OP-CODE	addr	Rd

（4）停机指令

7 6 5 4	3 2	1 0
OP-CODE	00	00

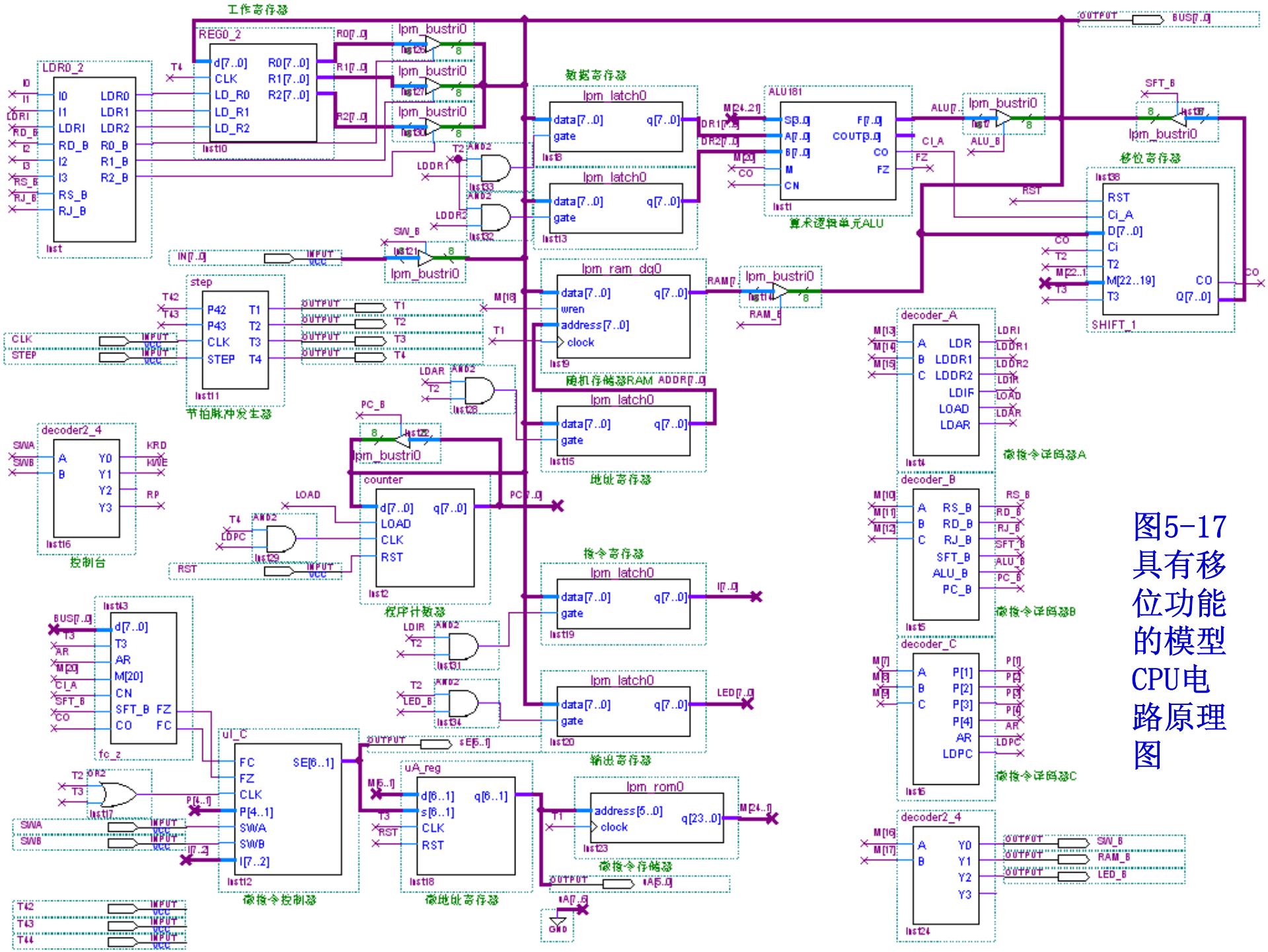
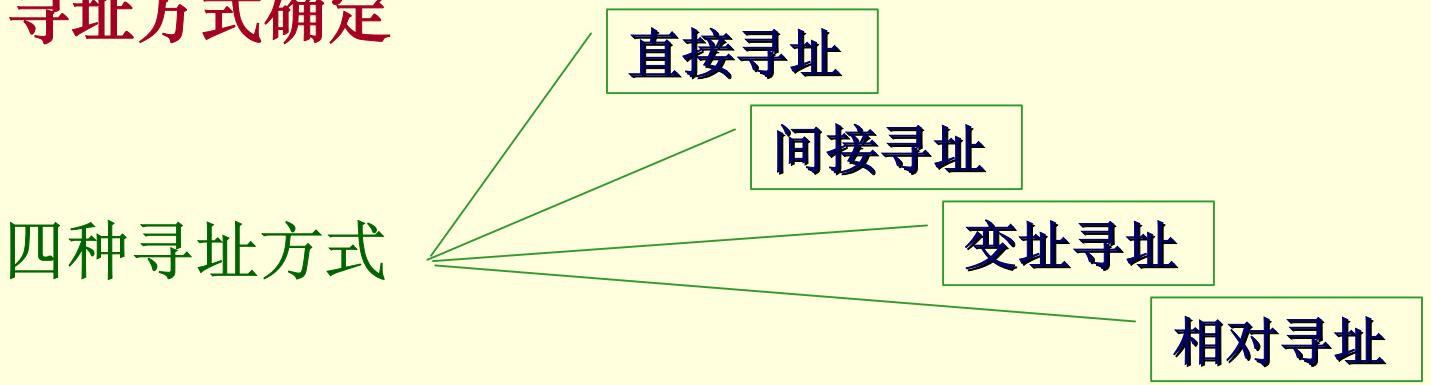
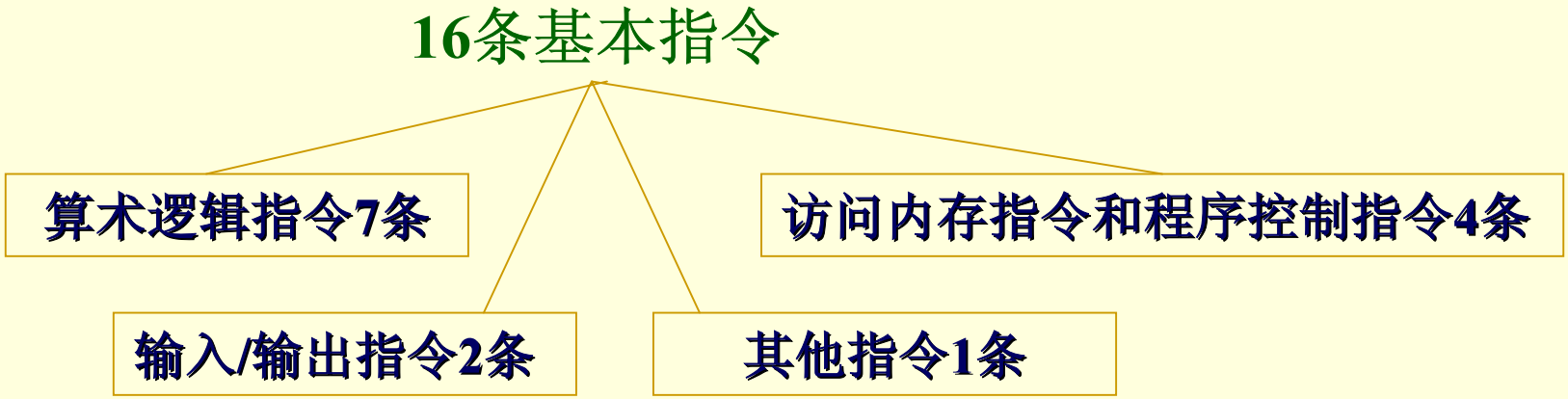


图5-17 具有移位功能的模型CPU电路原理图

5.5 含更多指令的CPU模型机设计

5.5.2 寻址方式确定





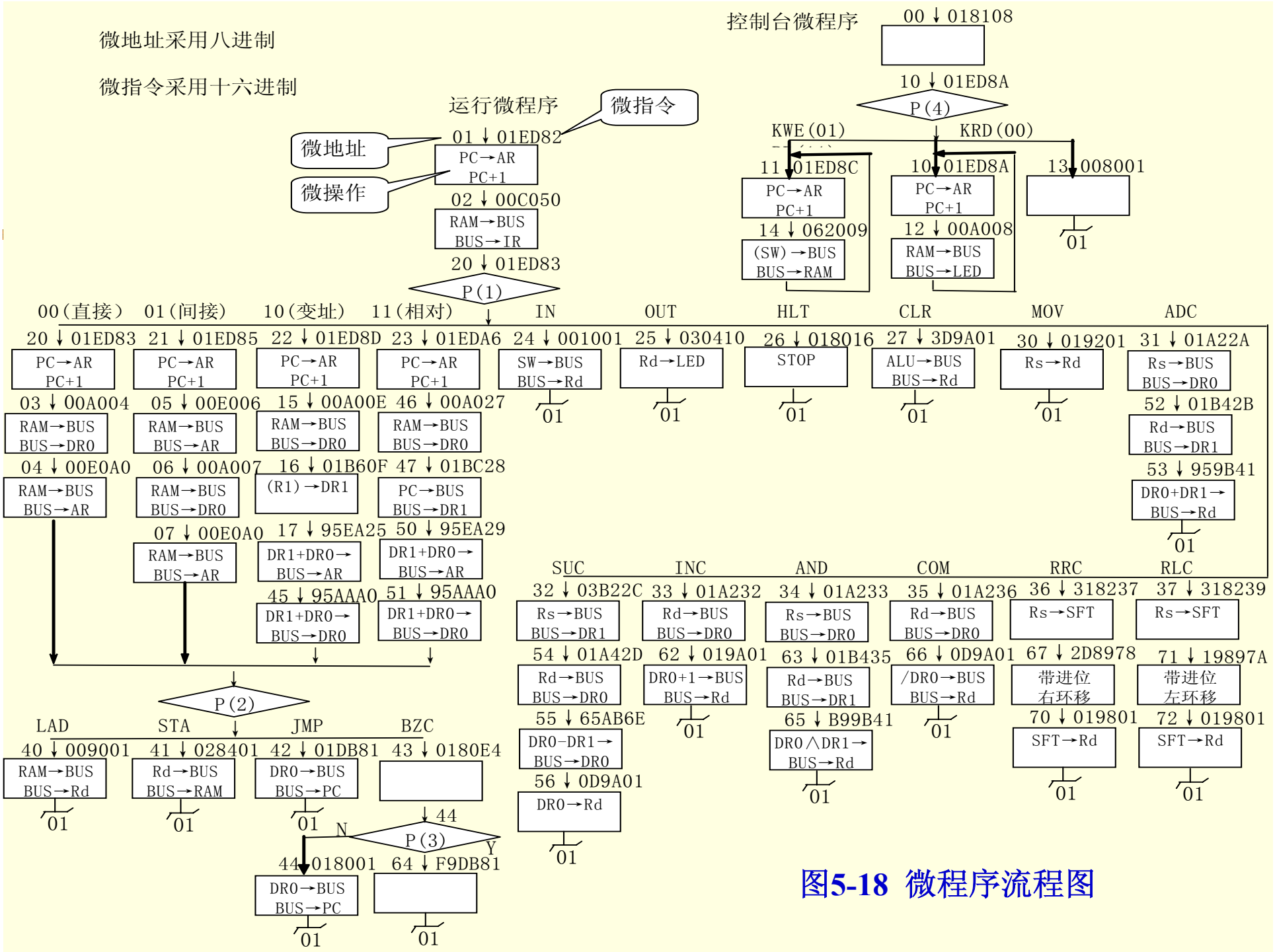

微地址采用八进制
微指令采用十六进制
运行微程序
微指令
微地址
微操作
01 ↓ 01ED82
PC→AR
PC+1
02 ↓ 00C050
RAM→BUS
BUS→IR
20 ↓ 01ED83
P(1)
控制台微程序
00 ↓ 018108
10 ↓ 01ED8A
P(4)
KWE(01)
KRD(00)
11 ↓ 01ED8C
PC→AR
PC+1
14 ↓ 062009
(SW)→BUS
BUS→RAM
10 ↓ 01ED8A
PC→AR
PC+1
12 ↓ 00A008
RAM→BUS
BUS→LED
13 ↓ 008001
01
00(直接)
01(间接)
10(变址)
11(相对)
IN
OUT
HLT
CLR
MOV
ADC
20 ↓ 01ED83
PC→AR
PC+1
21 ↓ 01ED85
PC→AR
PC+1
22 ↓ 01ED8D
PC→AR
PC+1
23 ↓ 01EDA6
PC→AR
PC+1
24 ↓ 001001
SW→BUS
BUS→Rd
25 ↓ 030410
Rd→LED
26 ↓ 018016
STOP
27 ↓ 3D9A01
ALU→BUS
BUS→Rd
30 ↓ 019201
Rs→Rd
31 ↓ 01A22A
Rs→BUS
BUS→DR0
03 ↓ 00A004
RAM→BUS
BUS→DR0
05 ↓ 00E006
RAM→BUS
BUS→AR
15 ↓ 00A00E
RAM→BUS
BUS→DR0
46 ↓ 00A027
RAM→BUS
BUS→DR0
52 ↓ 01B42B
Rd→BUS
BUS→DR1
04 ↓ 00E0A0
RAM→BUS
BUS→AR
06 ↓ 00A007
RAM→BUS
BUS→DR0
16 ↓ 01B60F
(R1)→DR1
47 ↓ 01BC28
PC→BUS
BUS→DR1
53 ↓ 959B41
DR0+DR1→
BUS→Rd
07 ↓ 00E0A0
RAM→BUS
BUS→AR
17 ↓ 95EA25
DR1+DR0→
BUS→AR
50 ↓ 95EA29
DR1+DR0→
BUS→AR
45 ↓ 95AAA0
DR1+DR0→
BUS→DR0
51 ↓ 95AAA0
DR1+DR0→
BUS→DR0
SUC
INC
AND
COM
RRC
RLC
32 ↓ 03B22C
Rs→BUS
BUS→DR1
33 ↓ 01A232
Rd→BUS
BUS→DR0
34 ↓ 01A233
Rs→BUS
BUS→DR0
35 ↓ 01A236
Rd→BUS
BUS→DR0
36 ↓ 318237
Rs→SFT
37 ↓ 318239
Rs→SFT
54 ↓ 01A42D
Rd→BUS
BUS→DR0
62 ↓ 019A01
DR0+1→BUS
BUS→Rd
63 ↓ 01B435
Rd→BUS
BUS→DR1
66 ↓ 0D9A01
/DR0→BUS
BUS→Rd
67 ↓ 2D8978
带进位
右环移
71 ↓ 19897A
带进位
左环移
55 ↓ 65AB6E
DR0-DR1→
BUS→DR0
65 ↓ B99B41
DR0∧DR1→
BUS→Rd
70 ↓ 019801
SFT→Rd
72 ↓ 019801
SFT→Rd
56 ↓ 0D9A01
DR0→Rd
P(2)
LAD
STA
JMP
BZC
40 ↓ 009001
RAM→BUS
BUS→Rd
41 ↓ 028401
Rd→BUS
BUS→RAM
42 ↓ 01DB81
DR0→BUS
BUS→PC
43 ↓ 0180E4
↓ 44
N
P(3)
Y
44 018001
DR0→BUS
BUS→PC
64 ↓ F9DB81

图5-18 微程序流程图

5.5 含更多指令的CPU模型机设计

5.5.4 微程序代码在LPM模块中的加载

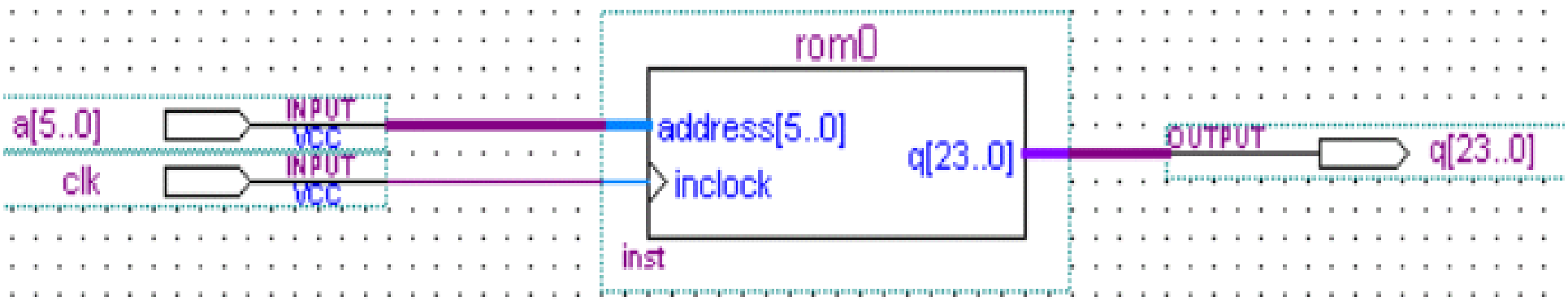


图5-19 lpm_ROM的结构图

5.5 含更多指令的CPU模型机设计

5.5.4 微程序代码在LPM模块中的加载

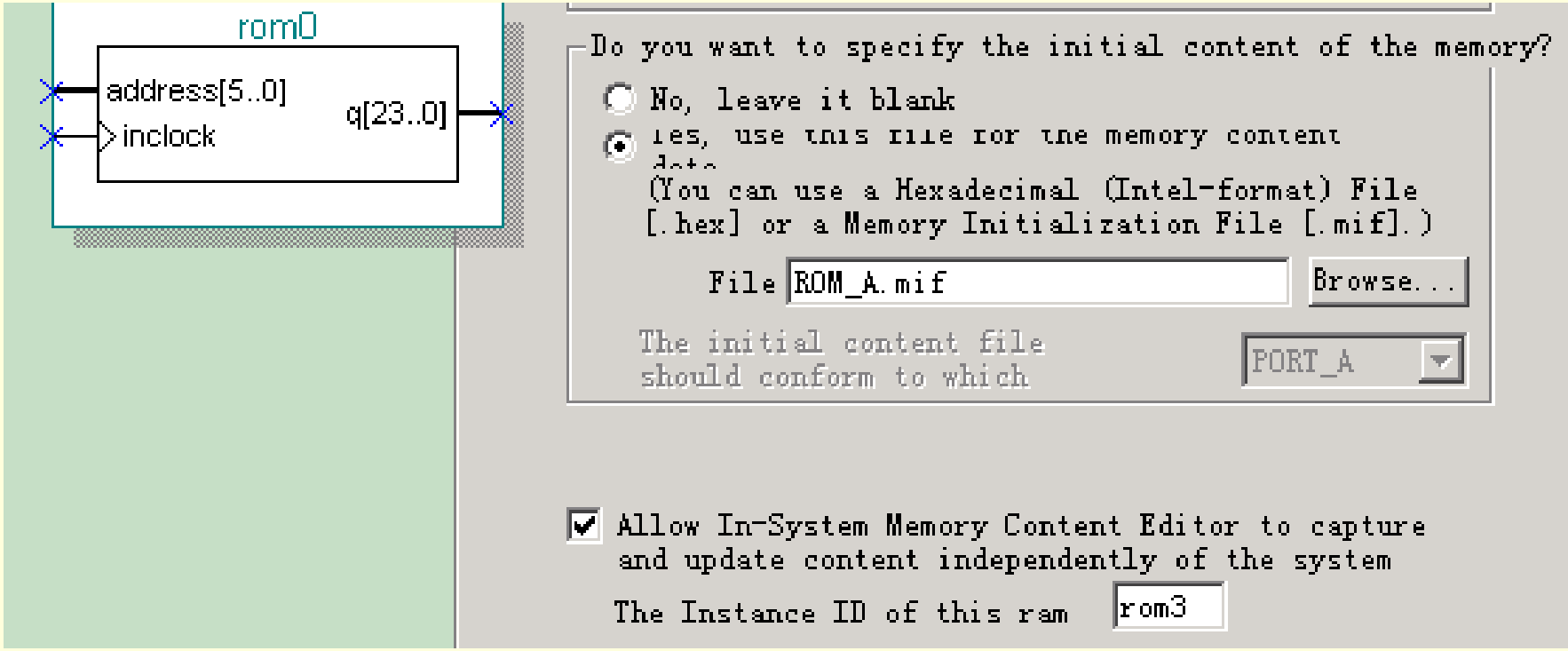


图5-20 设置在系统ROM/RAM读写允许

5.5 含更多指令的CPU模型机设计

5.5.4 微程序代码在LPM模块中的加载

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5-21 **rom_a.mif**中的数据

5.5 含更多指令的CPU模型机设计

5.5.4 微程序代码在LPM模块中的加载

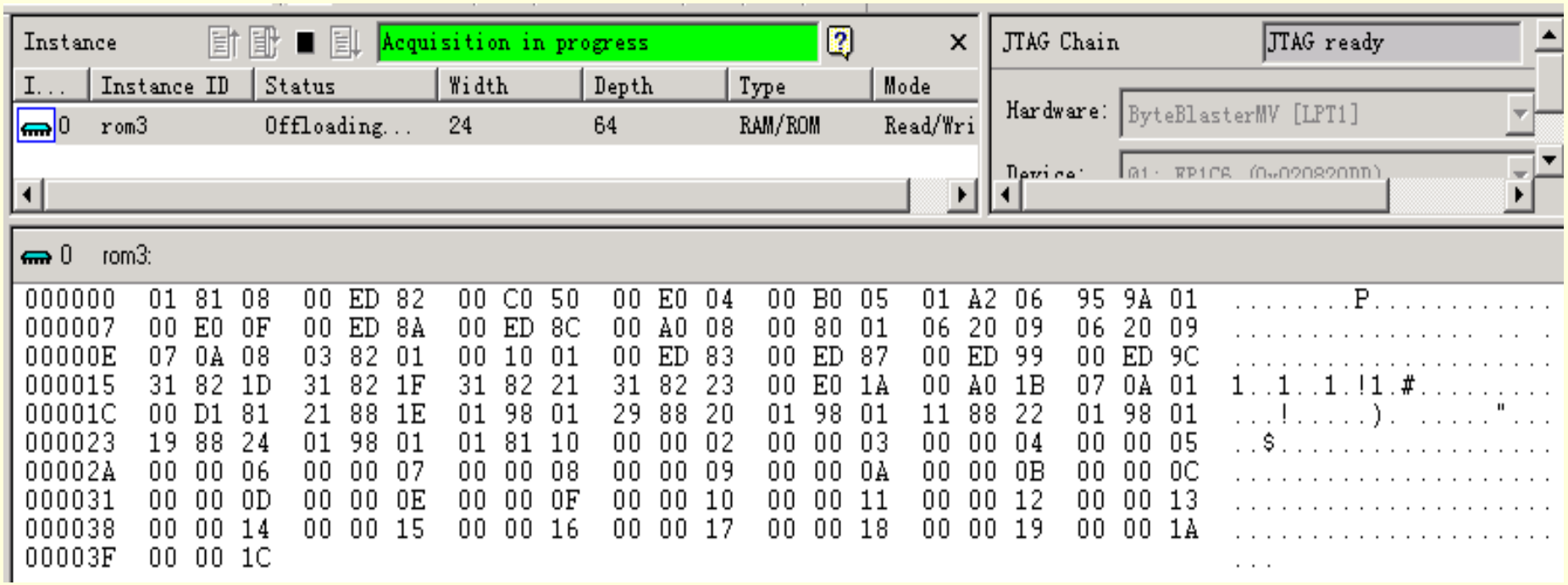


图5-22 在系统存储模块读写

5.6 8051单片机IP软核应用系统构建

5.6.1 K8051单片机软核基本功能和结构

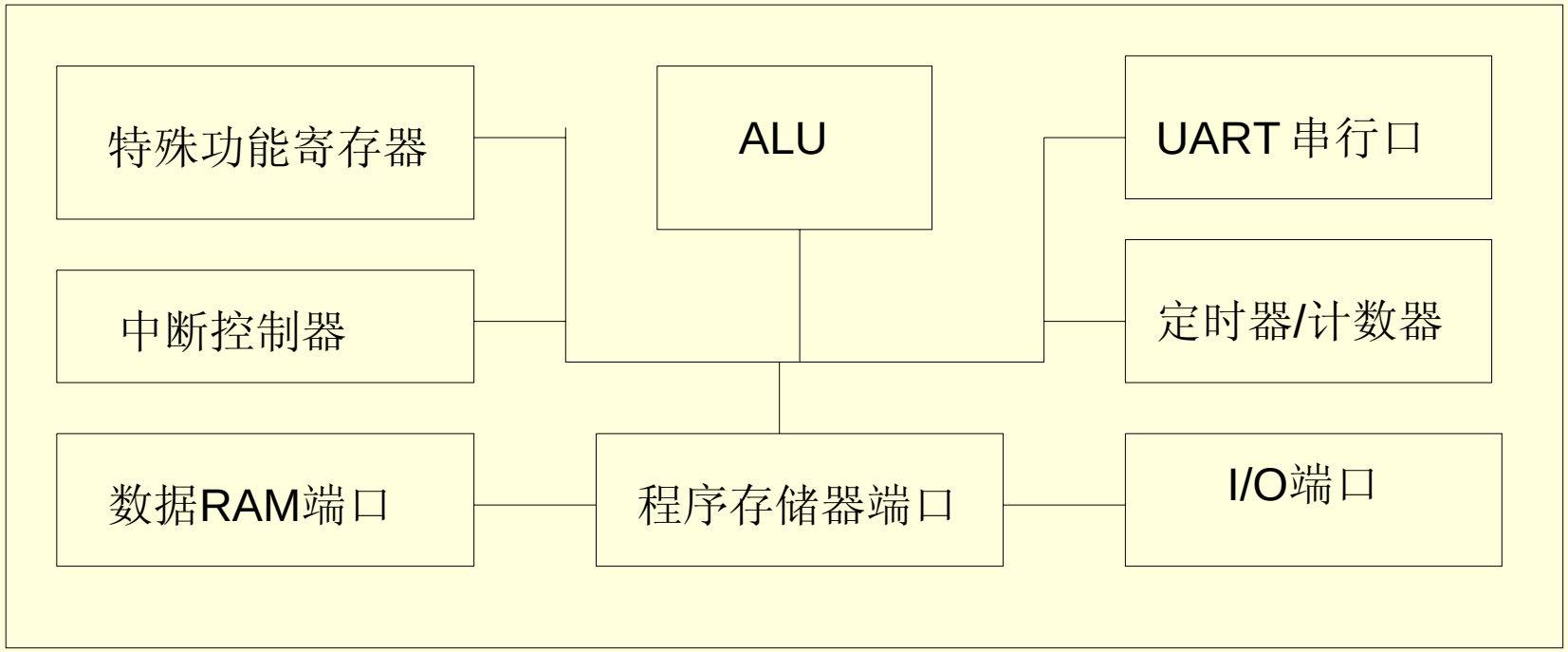


图5-23 K8051结构模块框图

5.6 8051单片机IP软核应用系统构建

5.6.1 K8051单片机软核基本功能和结构

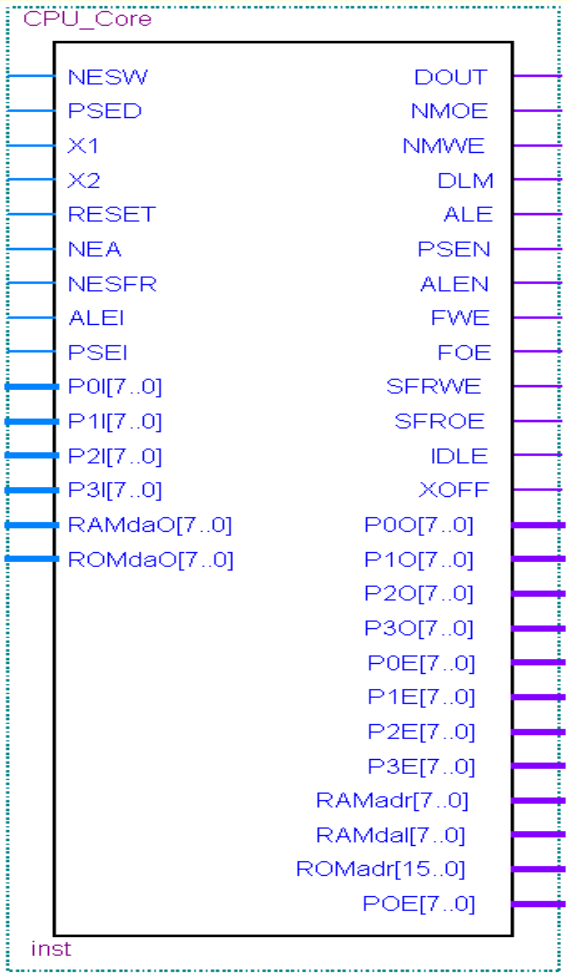


图5-24 K8051原理图元件

表5-10　　K8051单片机核信号端口功能明

单片机信号	端口类型	功能说明
ROMadr[15..0]	输　出	程序存储器地址总线
ROMdaO[7..0]	输　入	程序存储器数据总线
NMOE	输　出	程序存储器输出使能，低电平有效
RAMadr[7..0]	输　出	片内RAM地址总线
RAMdaI[7..0]	输　出	片内RAM数据输入总线（由单片机核输出）
RAMdaO[7..0]	输　入	片内RAM数据输出总线
FOE	输　出	片内RAM数据输出使能，低电平有效
FWE	输　出	片内RAM数据写入使能，低电平有效
SFROE	输　出	外部特殊寄存器输出使能，低电平有效
SFRWE	输　出	外部特殊寄存器写入使能，低电平有效
NESFR	输入	如果没有外部特殊寄存器，拉高此电平
P0O[7..0]	输出	P0口数据输出端，8位
P1O[7..0]	输出	P1口数据输出端，8位
P2O[7..0]	输出	P2口数据输出端，8位
P3O[7..0]	输出	P3口数据输出端，8位
P0I[7..0]	输入	P0口数据输入端，8位
P1I[7..0]	输入	P1口数据输入端，8位
P2I[7..0]	输入	P2口数据输入端，8位
P3I[7..0]	输入	P3口数据输入端，8位
P0E[7..0]	输出	P0口作为双向口的控制信号8位，执行输出指令时，为低电平
P1E[7..0]	输出	P1口作为双向口的控制信号8位，执行输出指令时，为低电平
P2E[7..0]	输出	P2口作为双向口的控制信号8位，执行输出指令时，为低电平
P3E[7..0]	输出	P3口作为双向口的控制信号8位，执行输出指令时，为低电平
NEA	输入	使能程序计数器的值进入P0和P2口
X1	输入	单片机工作时钟输入端
X2	输入	单片机工作时钟输入端，但在进入休闲状态时可控制停止
RESET	输入	复位信号线
ALE	输出	地址锁存信号
PSEN	输出	外部程序存储器使能，低电平有效
ALEN	输出	对ALE和PSEN信号的双向控制信号，低电平允许输出
XOFF	输出	振荡器禁止信号，用于省电模式
IDLE	输出	在休闲模式中，可通过外部控制NX2的时钟输入

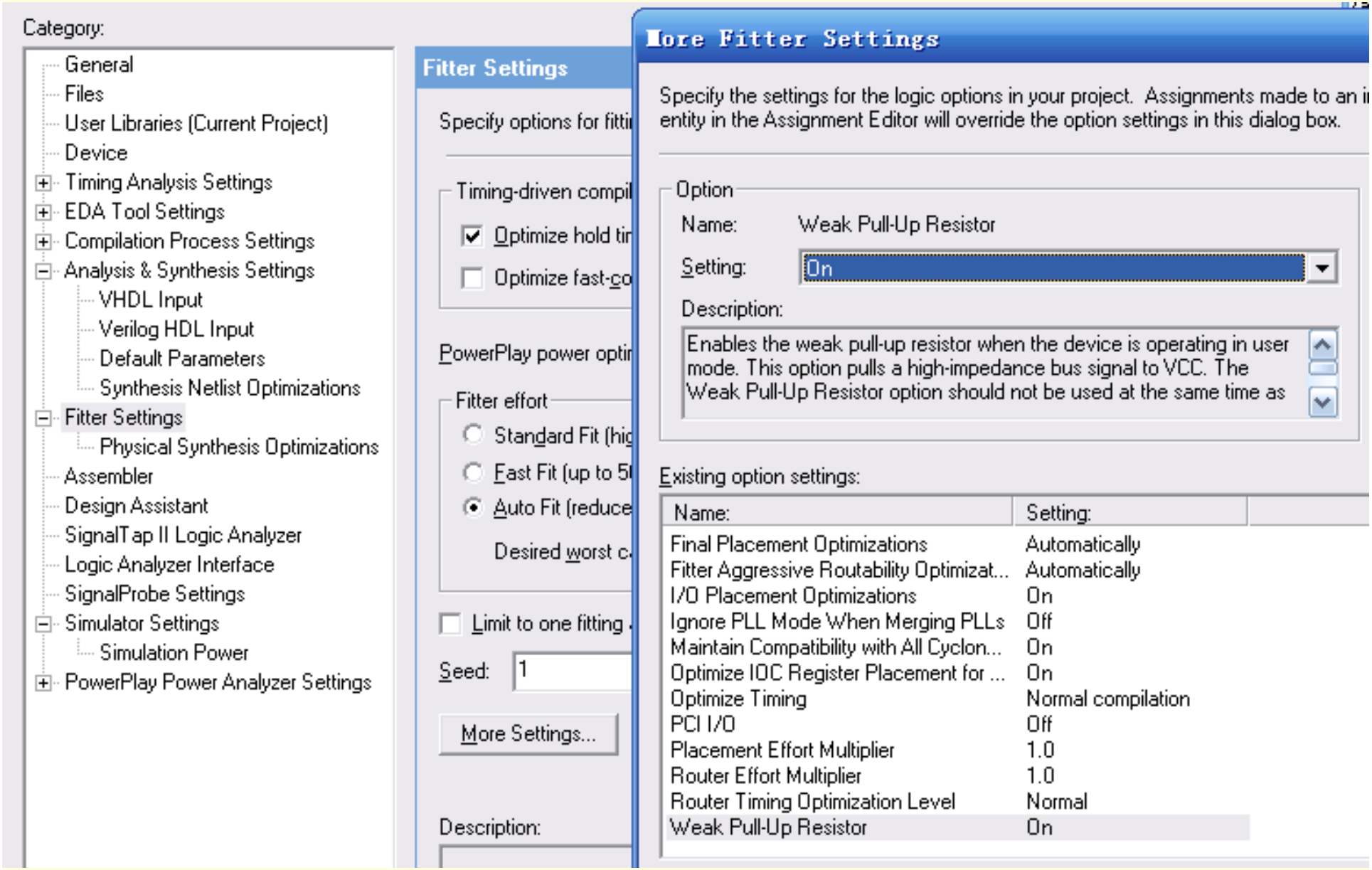


图5-25　设置FPGA的总线口输出为上拉

5.6 8051单片机IP软核应用系统构建

5.6.1 K8051单片机软核基本功能和结构

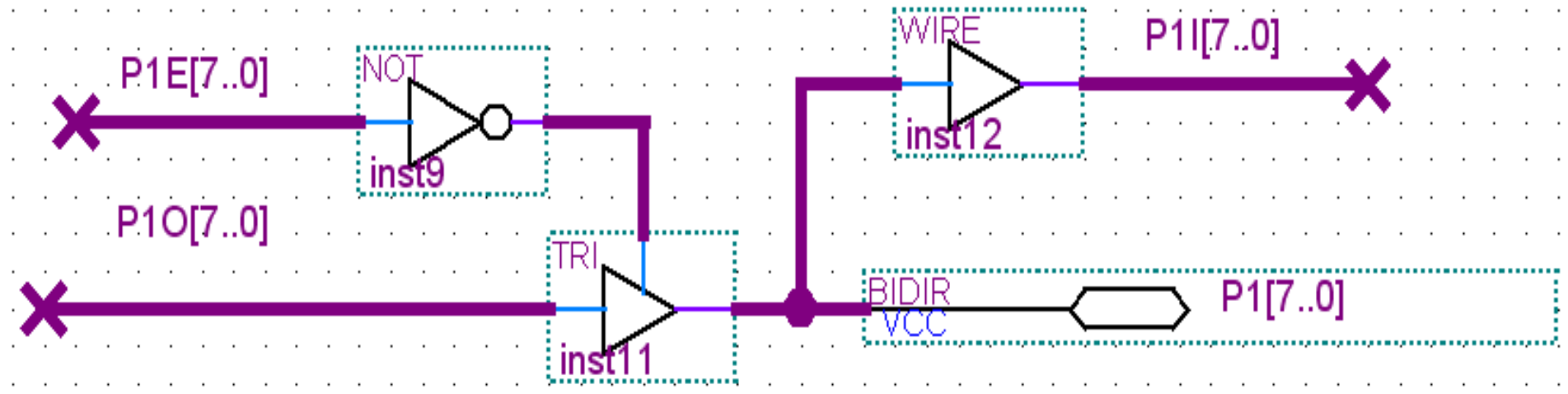


图5-26　K8051单片机I/O口设置成双向口的电路

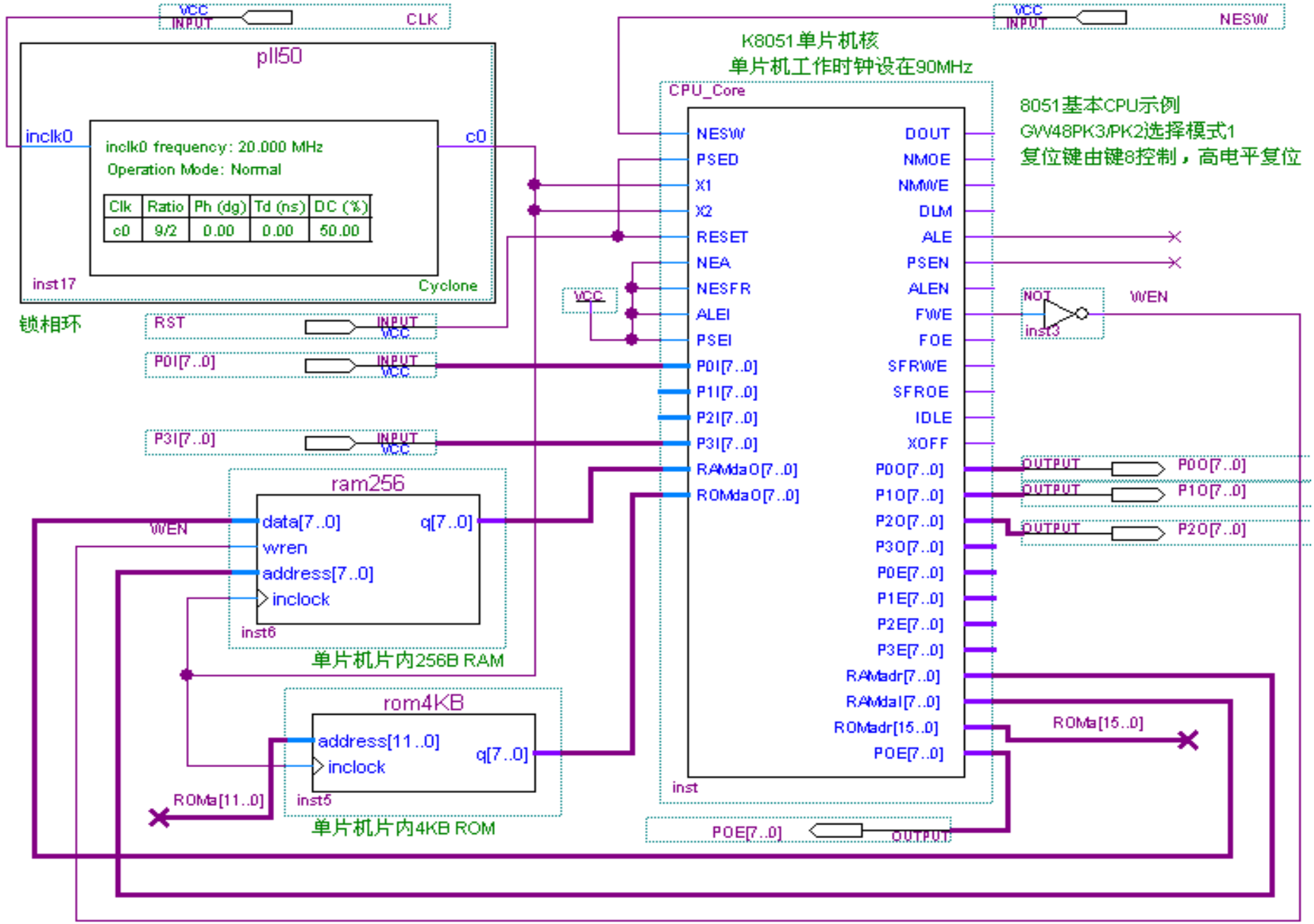


图5-27 K8051基本实用电路

```
            ORG  0000H
MAIN :      MOV    SP,#60H
            MOV    24H,#00H
            MOV    30H,#01H
ROUND:      LCALL   DELAY1
            MOV    A,24H
            INC    A
            MOV    24H,A
            MOV    P1,A
            MOV    A,30H
            RR     A
            MOV    P0,A
            MOV    30H,A
            NOP
            NOP
            MOV    A,P0
            MOV    B,P3
            ADD    A,B
            MOV    P2,A
            LCALL   DELAY1
            SJMP   ROUND
DELAY:      MOV    20H,#0FFH
   W1:      MOV    21H,#0FFH
   W2:      DJNZ   21H,W2
            DJNZ   20H,W1
            RET
DELAY1:     MOV    22H,#08H
    W3:     LCALL DELAY
            DJNZ   22H,W3
            RET
            END
```

图5-28 TEST1.asm汇编程序

5.6 8051单片机IP软核应用系统构建

5.6.2 K8051单片机软核实用系统构建和软件测试

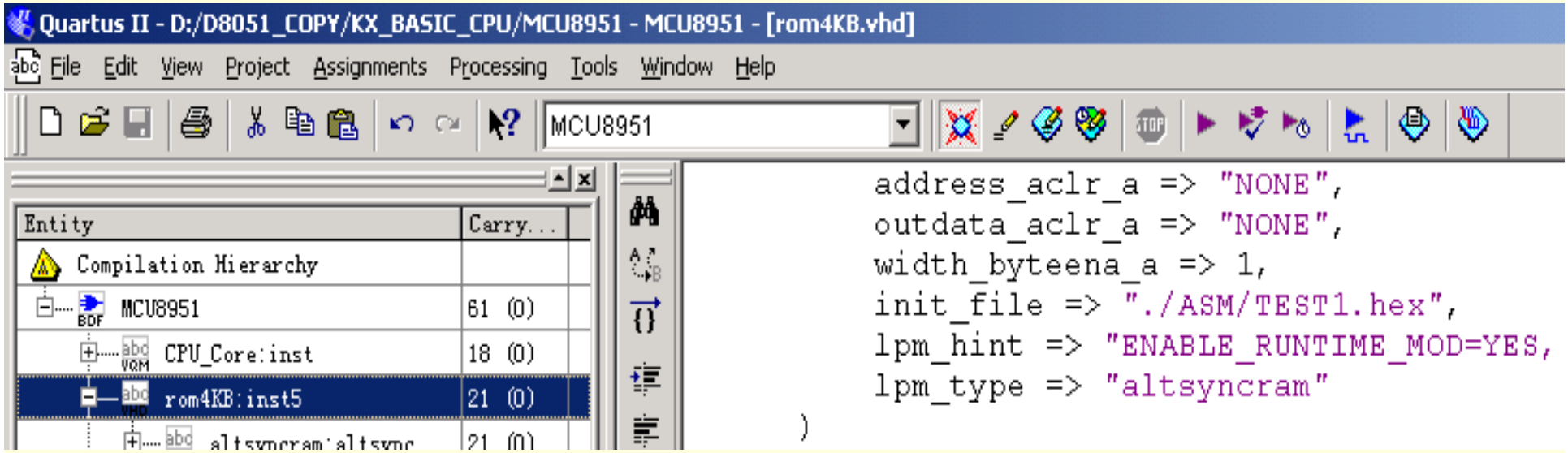


图5-29 ROM初始化文件路径

5.6 8051单片机IP软核应用系统构建

5.6.2 K8051单片机软核实用系统构建和软件测试

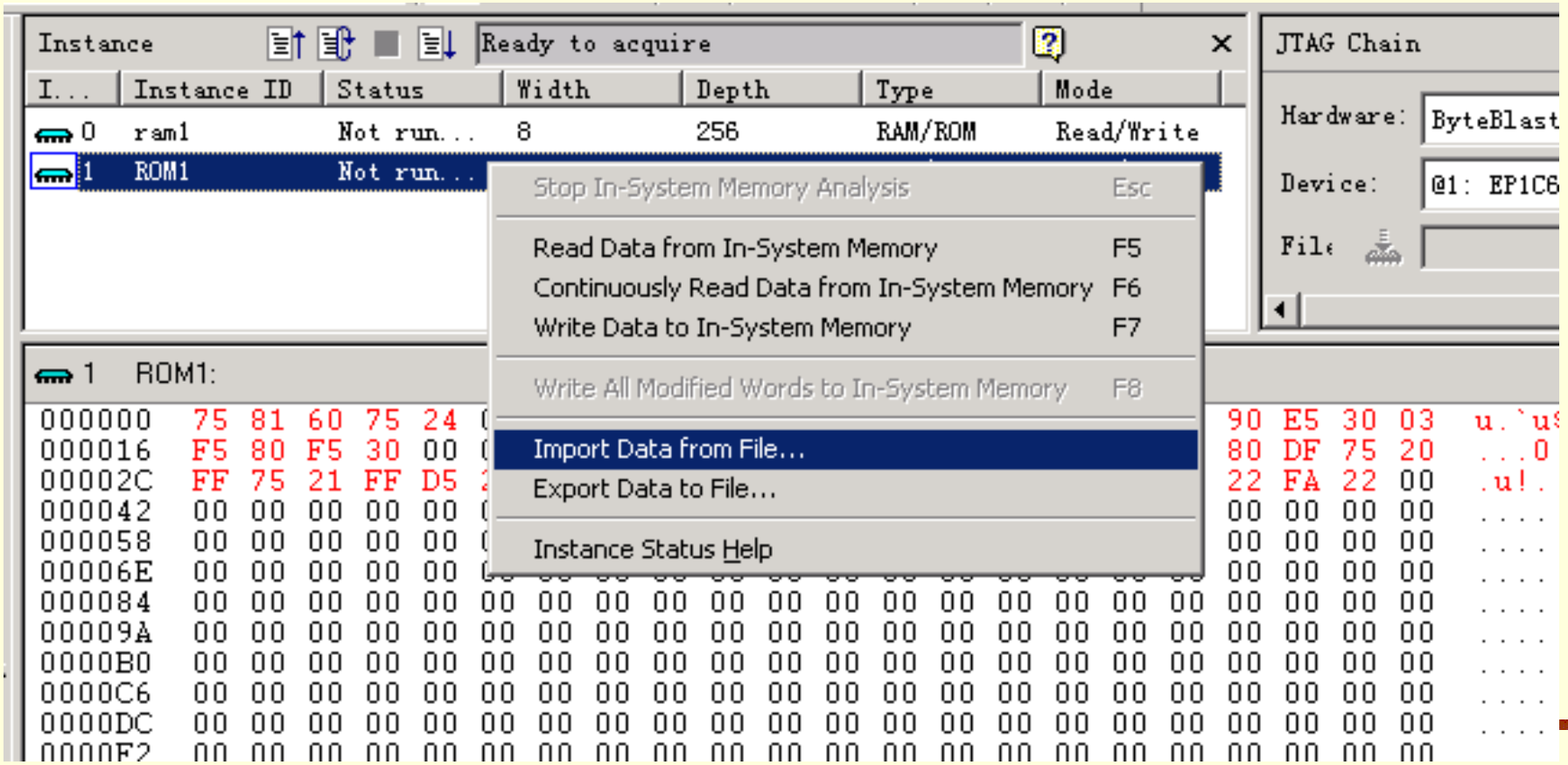


图5-30 利用In-System Memory Content Editor下载汇编程序HEX代码

习 题

5-1．简要解释下列名词术语：

通用寄存器，暂存器，指令寄存器IR，程序计数器PC，时序系统，微命令，组合逻辑控制，微程序控制，数据通路结构，指令周期，时钟周期，微指令周期，总线周期，微指令，微程序，控制存储器。

5-2．简要说明指令周期、时钟周期和操作节拍三种时间参数的含义及相互关系。

5-3．叙述微程序控制器，并解释执行一条加法指令的步骤(从取指令开始)。

5-4．简要说明硬布线控制器与微程序控制器组成的异同之处及两种控制器各自的优缺点。

5-5．说明控制器在计算机中的作用和地位。

5-6．说明微指令的下地址字段的组成和用法。

5-7．用模型计算机指令设计方法，设计并调试原码一位乘法和补码一位除法两个子程序。

习 题

5-8．用模型计算机指令设计，并调试子程序：输入正或负的10进制整数，输出它的2进制补码表示。

5-9．设计并调试一条完成两个内存单元内容相加并写回其中一个单元的指令格式和相应的微程序。

5-10．设计并调试一条完成两个寄存器内容相加并实现半字交换后写回其中一个寄存器的指令的指令格式和相应的微程序。

5-11．设计两个主存单元内容相加的指令，寄存器内容右移的指令，写出指令的格式和相应的微程序，说明微程序的执行过程。

5-12．说明微程序和硬布线两种控制器的组成和它们的异同之处。

5-13．写出在模型CPU上指令 LOAD　Rd，（mem）的执行过程，其含义是将存储单元mem中的数据送寄存器Rd。

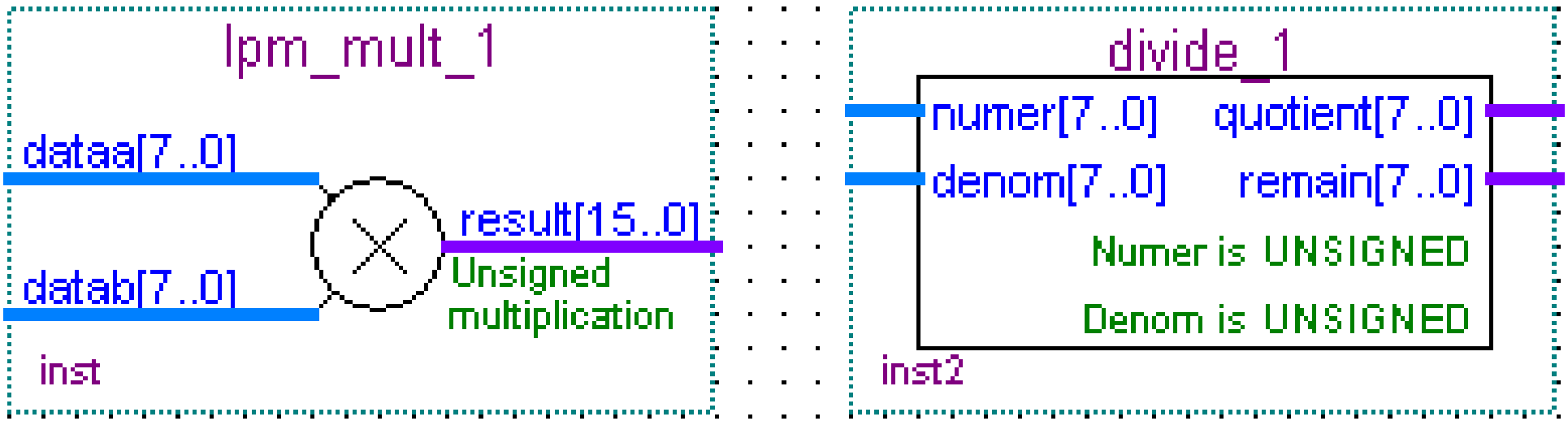
5-14．写出.在模型CPU上指令 LOAD　Rd，[（mem）]的执行过程，其含义是以存储单元mem的内容为地址，将该地址的存储单元中的数据送寄存器Rd。　.

习 题

5-15. 写出.在模型CPU上指令 STA Rs，（mem）的执行过程，其含义是将寄存器Rs中的数据存入存储单元mem中。

5-16. 写出.在模型CPU上指令 STA Rs，（Rd）的执行过程，其含义是将寄存器Rs中的数据存入以寄存器Rd的内容为地址的存储单元mem中。

5-17 在原模型CPU电路的基础上增加一个乘法器和一个除法器，使CPU能够完成加、减、乘除运算。乘法器和除法器可采用lpm_lib中的宏单元lpm_mul和lpm_div来实现。乘法器和除法器的形式如下。



习 题

在指令系统中目的操作数采用寄存器寻址，源操作数具有寄存器寻址、寄存器间接寻址和存储器间接寻址方式，试设计一个包含这些执行过程的流程图。例如：Op Rd，Rs ；其中操作码Op是加、减、乘、除运算（ADD、SUB、MUL、DIV）中的一种，源操作数Rs，目的操作数Rd。

直接寻址： Op Rd，Rs ；Rd Op Rs → Rd

寄存器间接寻址： Op Rd，[Rs] ；Rd Op [Rs] → Rd

存储器间接寻址方式： Op Rd，（mem） ；Rd Op （mem） → Rd

5-18. 条件与题5-17相同。在指令系统中源操作数采用寄存器寻址，目的操作数具有寄存器寻址、寄存器间接寻址和存储器间接寻址方式，试设计一个包含这些执行过程的流程图。

直接寻址： Op Rd，Rs ；Rd Op Rs → Rd

寄存器间接寻址： Op [Rd]，Rs ；[Rd] Op Rs → [Rd]

存储器间接寻址方式： Op （mem），Rs， ；（mem） Op Rs → （mem）

习 题

5-19. 写出.在模型CPU上指令 MOV mem1，mem2的执行过程，其含义是将存储单元mem2中的数据存入存储单元mem1中。

5-20 写出.在模型CPU上指令 ADD Rd，mem的执行过程，其含义是将寄存器Rd的数据与存储单元mem中的数据相加，结果存入寄存器Rd。

5-21 写出.在模型CPU上指令 ADD Rd，（Rs）的执行过程，其含义是将寄存器Rd的数据与以寄存器Rs中的内容为地址的存储单元中的数据相加，结果存入寄存器Rd。

实验与设计

实验5-1. 基本模型计算机设计与实现

参考实验示例和实验课件：/CMPUT_EXPMT/CH5_Expt / DEMO_51/CPU5A 和实验5_1.ppt

表5-11　　控制台操作电平

SWB	SWA	控制台指令
0	0	读内存（KRD）
0	1	写内存（KWE）
1	1	启动程序（RP）

实验与设计

表5-12 实验程序： 模型机指令及编码形式

地址（16进制）	内容（16进制）	助记符	说明
00	00	IN	“INPUT”→ R0，键盘输入数据
01	10	ADD [0AH]	[R0]+[0AH] → R0
02	0A		
03	20	STA [0BH]	[R0] → [0BH]
04	0B		
05	30	OUT [0BH]	[0BH] “OUTPUT”，显示输出数据
06	0B		
07	40	JMP [08H]	[09H] → PC ，以[08H]内容为转移地址
08	00		
09	00		
0A	34	DB 34H	被加数（自定）
0B	XX		求和结果

实验与设计

实验5-1. 基本模型计算机设计与实现

表5-13　　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

实验与设计

实验5-1. 基本模型计算机设计与实现

现代计算机组成原理实验

IN	00	OUT	00	ALU	00
R0	00	R1	00	R2	00
DR1	00	DR2	00	BUS	00
PC	00	AR	00	RAM	00
IR	00	uA	00	MC	018110

图5-31 LCD液晶显示屏

实验与设计

实验5-1. 基本模型计算机设计与实现

I...	Instance ID	Status	Width	Depth	Type	Mode
0	rom5	Not run...	24	64	RAM/ROM	Read/Write
1	ram5	Not run...	8	256	RAM/ROM	Read/Write

Hardware: ByteBlasterMV [LPT1]

Device: @1: EP1C6 (0x020820DD)

File:

0 rom5:

```
000000  01 81 10  01 ED 82  00 C0 48  00 E0 04  00 B0 05  01 A2 06  91 9A 01  .......H........
000007  00 E0 0D  00 10 01  01 ED 83  01 ED 87  01 ED 8E  01 ED 96  03 82 01  ................
00000E  00 E0 0F  00 A0 15  01 ED 92  01 ED 94  01 A0 10  01 80 01  06 20 11  ..............  .
000015  01 0A 01  00 D1 81  00 00 00  00 00 00  00 00 00  00 00 00  00 00 00  ................
00001C  00 00 00  00 00 00  00 00 00  00 00 00  00 00 00  00 00 00  00 00 00  ................
000023  00 00 00  00 00 00  00 00 00  00 00 00  00 00 00  00 00 00  00 00 00  ................
00002A  00 00 00  00 00 00  00 00 00  00 00 00  00 00 00  00 00 00  00 00 00  ................
000031  00 00 00  00 00 00  00 00 00  00 00 00  00 00 00  00 00 00  00 00 00  ................
000038  00 00 00  00 00 00  00 00 00  00 00 00  00 00 00  00 00 00  00 00 00  ................
00003F  00 00 00                                                              ...
```

1 ram5:

```
000000  00 10 0A 20 0B 30 0B 40 00 00 34 00 00 00 00 00 00 00 00 00 00 00 00 00  ... .0.@..4.....
000019  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000032  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
```

图5-32 用在系统EAB读写工具对FPGA中的ROM和RAM进行观察和改写

实验与设计

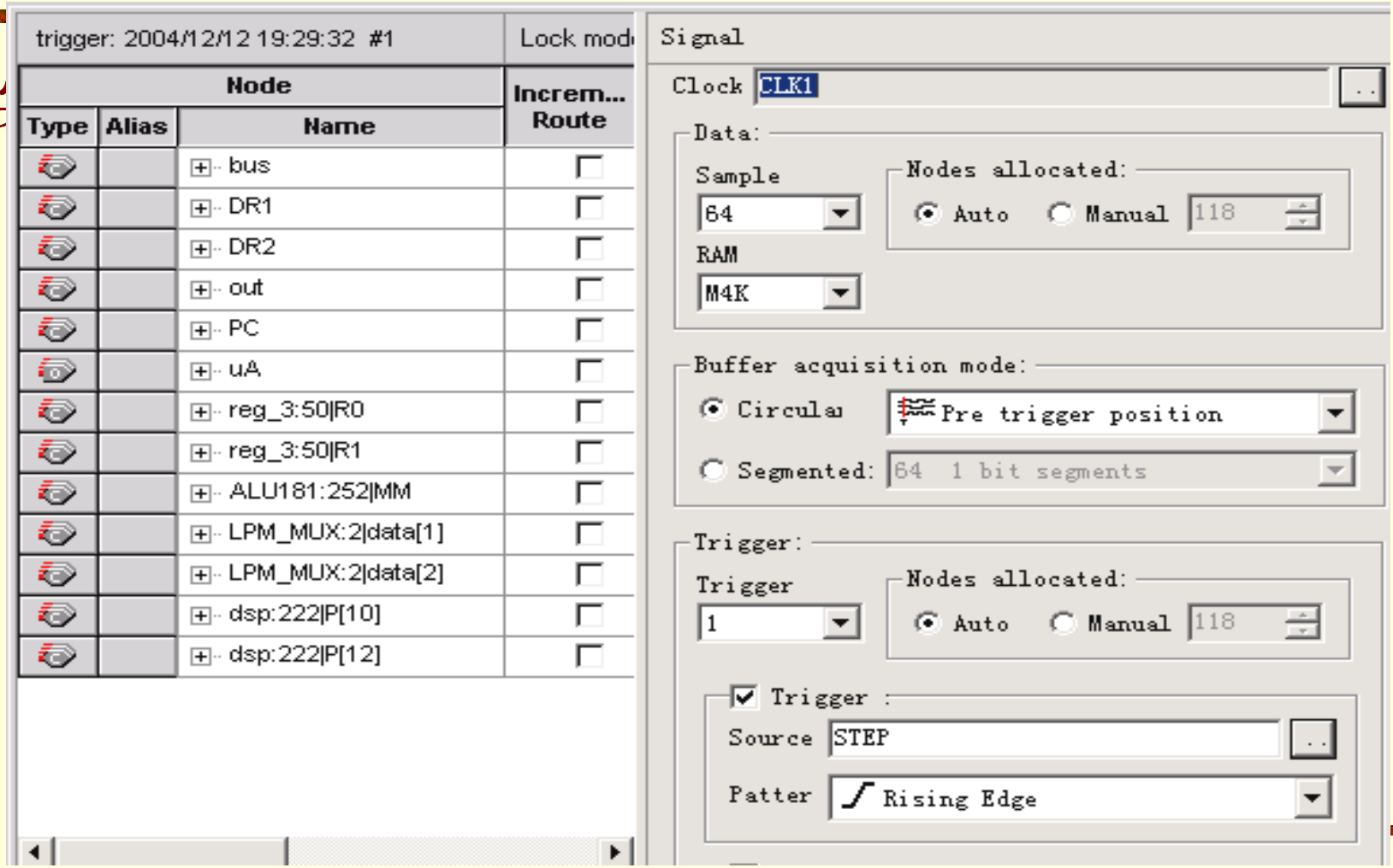


图5-33 嵌入式逻辑分析仪设置情况

实验与设计

实验5-1. 基本模型计算机设计与实现

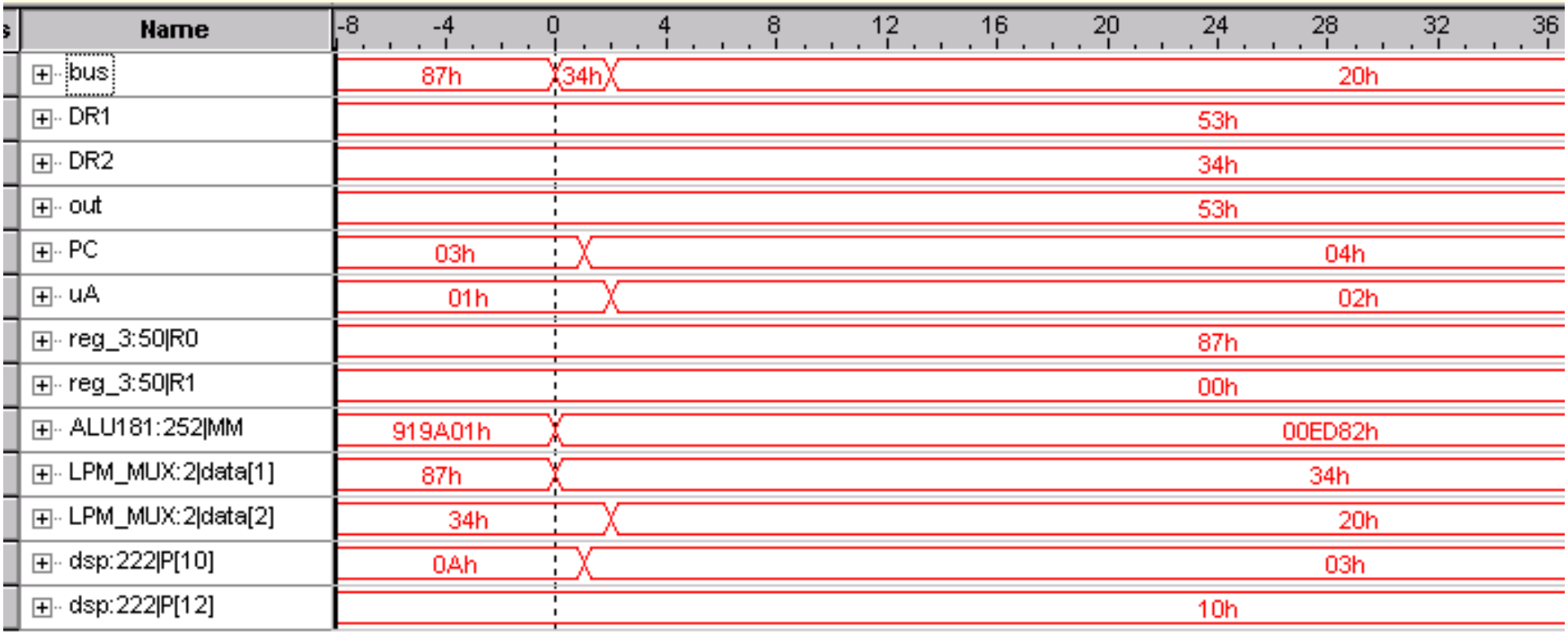


图5-34 嵌入式逻辑分析仪采样波形数据

实验与设计

实验5-1. 基本模型计算机设计与实现

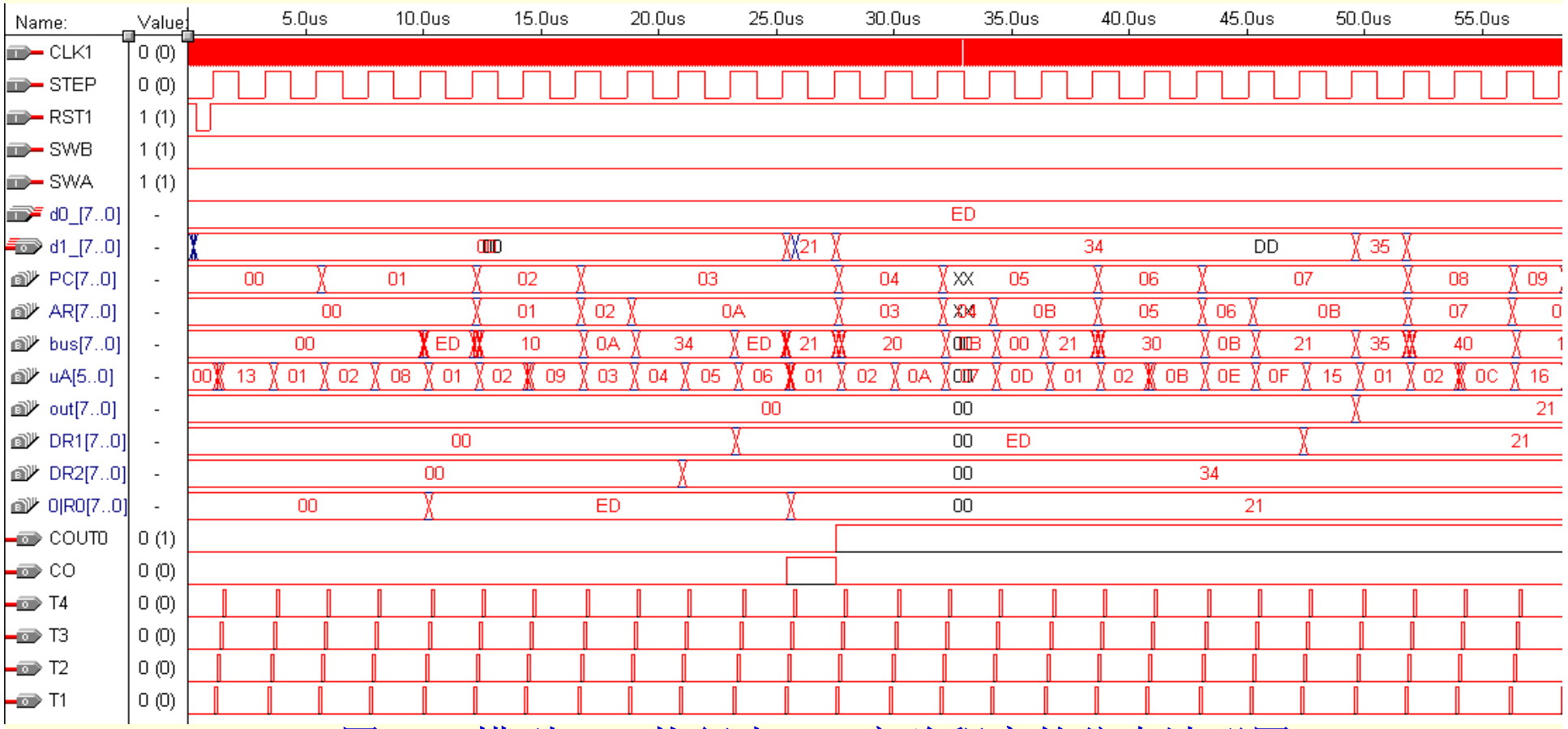


图5-35 模型CPU执行表5-12实验程序的仿真波形图

实验与设计

实验5-2. 带移位运算的模型机设计与实现

参考实验示例和实验课件： / CMPUT_EXPMT/CH5_Expt / DEMO_52/CPU6 和 实验5_2.ppt

表5-14　　带移位运算模型机微程序

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		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

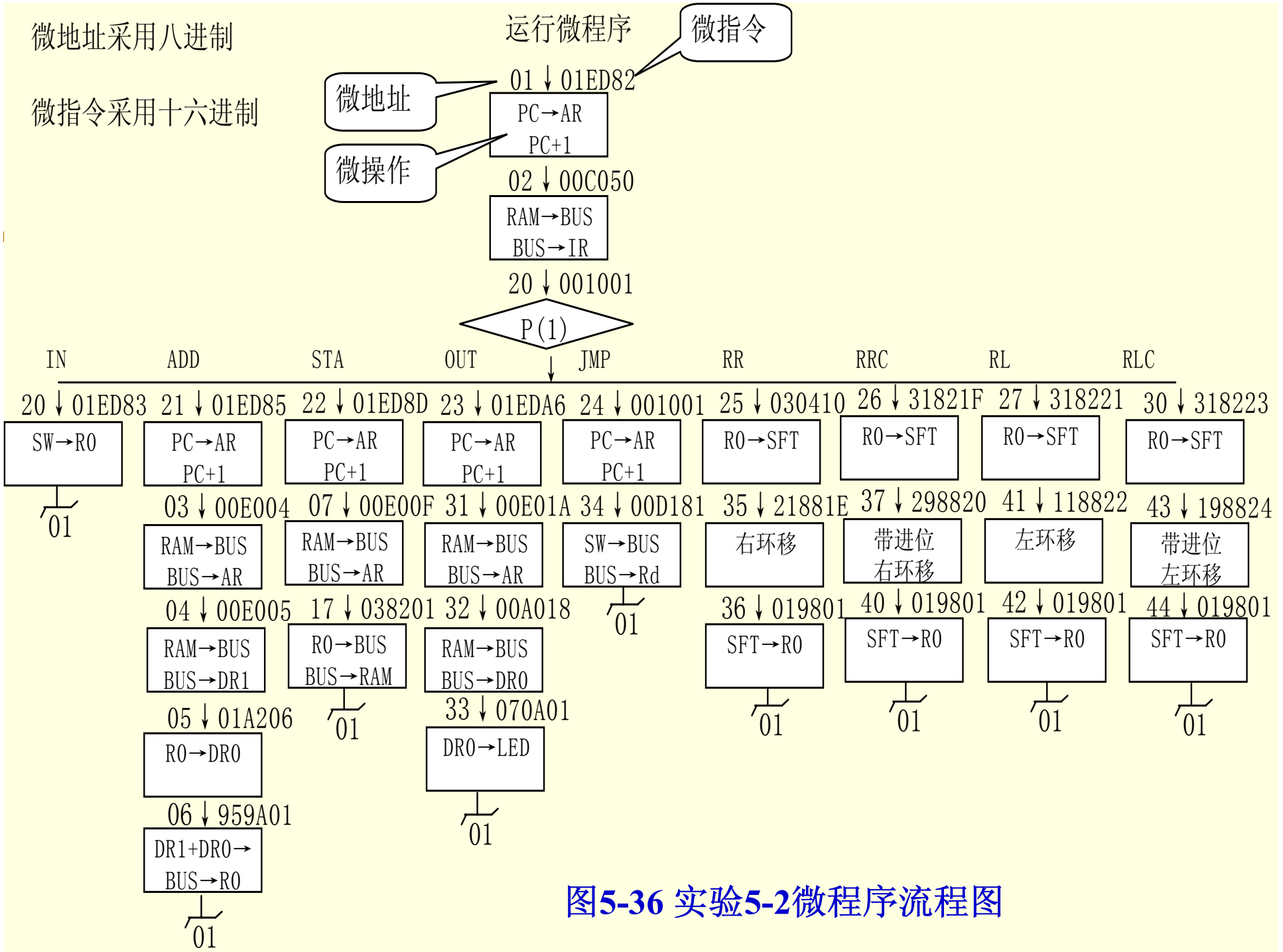


图5-36 实验5-2微程序流程图

实验与设计

实验5-2. 带移位运算的模型机设计与实现

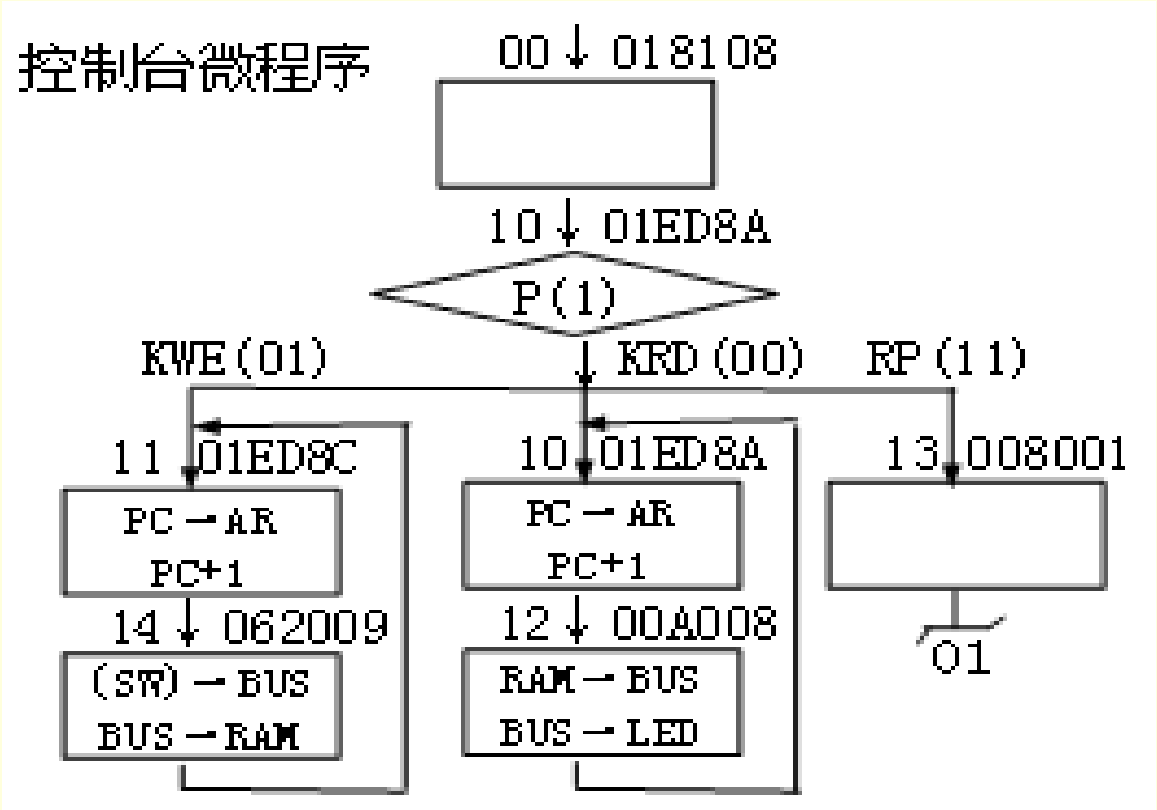


图5-37 控制台微程序

表5-15　　实验程序：指令及编码形式

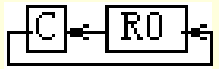
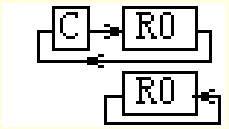
RAM地址（16进制）	内 容（16进制）	助记符	说 明
00	00	IN	“INPUT DEVICE”→R0
01	10	ADD [0DH]	R0+[0DH] →R0
02	0D		
03	80	RLC	“INPUT DEVICE”→R0
04	00	IN	
05	60	RRC	
06	70	RL	R0→[0EH]
07	20	STA [0EH]	
08	0E		[0EH] → OUTPUT
09	30	OUT [0EH]	
0A	0E		00→ PC
0B	40	JMP [addr]	自定
0C	00		存数单元
0D	45		
0E			

表5-16　　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	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，**键 1、键 2 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.**键入数据 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

表5-16　　微指令执行情况

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……	

实验与设计

实验5-2. 带移位运算的模型机设计与实现

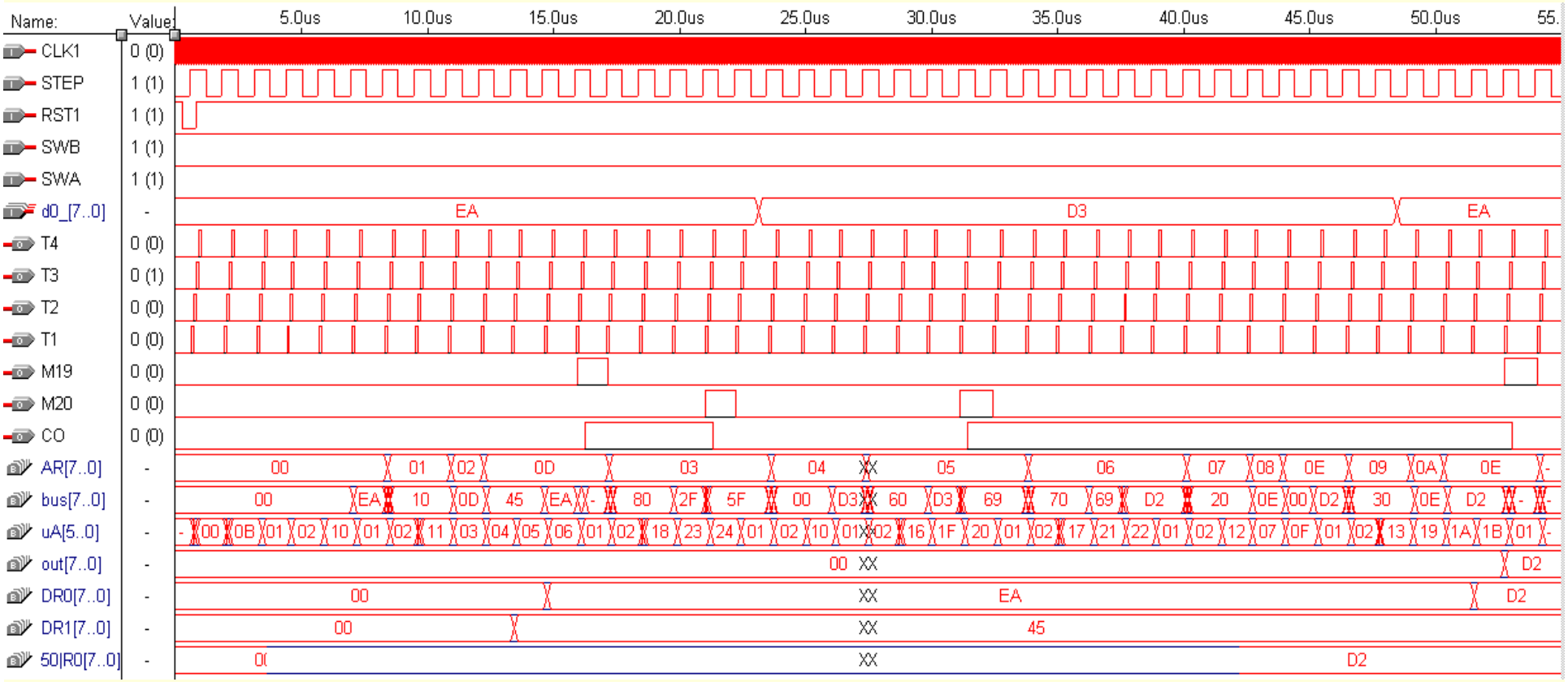


图5-38 模型CPU执行表5-16实验程序的仿真波形图

实验与设计

实验5-3. 含16条指令的CPU设计与实现

参考实验示例和实验课件：：/CMPUT_EXPMT/CH5_Expt/DEMO_53/CPU7 和 实验5_3.ppt

表5-17　　实验程序：指令及编码形式

程序	助记符	功能
$P00 44	IN　　01，R0	Input →R0
$P01 46	IN　　01，R2	Input →R2
$P02 98	ADC　　R2，R0	(R0)+(R2) →R0
$P03 81	MOV　　R0，R1	(R0) →R1
$P04 F5	RLC　　R1，R1	R1 带进位左移 →R1
$P05 0C　00	BZC　　00，00	Cy、Z 为 1 时，循环
$P07 08　00	JMP　　00	GOTO　00

实验与设计

实验5-3. 含16条指令的CPU设计与实现

表5-18　　微程序代码

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

实验与设计

实验5-3. 含16条指令的CPU设计与实现

表5-19　　微程序代码

微地址　代码	微地址　代码	微地址　代码	微地址　代码
$M00　018108 $M01　01ed82 $M02　00c050 …略 $M0E　01b60f $M0F　95ea25	$M10　01ed83 $M11　01ed85 …略 $M1D　01a236 $M1E　318237 $M1F　318239	$M20　009001 $M21　028401 $M22　01db81 …略 $M2E　0d9a01 $M2F　01aa30	…略 $M3A　019801 $M3B　070a08 $M3C　062009 $M3D　000000 $M3E　000000

表5-20　　实验5-3微指令详细执行情况

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	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

表5-20　　实验5-3微指令详细执行情况

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

实验5-4. 较复杂CPU应用程序设计实验

表5-21　　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

实验5-4. 较复杂CPU应用程序设计实验

表5-22　　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 rd
	data			
JMP　addr	1110	XXXX		Addr →PC
	addr			
JB　addr	1011	XXXX		若小于，则 addr → PC
	addr			

实验与设计

实验5-4. 较复杂CPU应用程序设计实验

表5-23　　实验程序

助记符		地址（十六进制）	机器代码（十六进制）	功能
LP0：	IN　R0	00	80	Input → R0
	MOV R1，1	01	91	1→R1
		02	01	
	MOV R2，0	03	92	0→R2
		04	00	
LP1：	CMP R0，R1	05	A1	R0-R1→R1
	JB　L2	06	B0	(LP2) →PC
		07	0D	
	ADD R1，R2	08	C6	R1+R2→R2
	INC R1	09	D1	R1+1→R1
	INC R1	0A	D1	R1+1→R1
	JMP L1	0B	E0	(LP1)→PC
		0C	05	
LP2：	OUT R2	0D	F2	R2→output
	JMP LP0	0E	E0	(LP0)→PC
		0F	00	

实验与设计

实验5-4. 较复杂CPU应用程序设计实验

表5-24　　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5-25 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	80	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	018110			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	
4	02	01ED82	01		执行第 1 条指令.（输入 IN RO）	PC→AR=00H 指向指令地址，PC+1=01H
5	10	00C048		80	取指令,将 RAM 中的指令送指令寄存器	RAM(00H)=00→BUS→IR=80H
6	01	001001			接收 IN 输入端口的数据，送 RO 寄存器	RO=07H，**键 1、键 2 输入数据 07H**
7	02	01ED82	02		执行第 2 条指令，（MOV R1，1）	PC→AR=01H 指向指令地址，PC+1=02H
8	11	00C048		91	取指令,将 RAM 中的指令送指令寄存器	AR=01H，RAM=91H→BUS→IR=91H
9	03	01ED83			指向操作数地址	PC→AR=02H 指向操作数地址，PC+1=03H
10	01	009001			RAM 内容送 R1	RAM(02H)=01H→BUS→R1=01H
11	02	01ED82	04		执行第 3 条指令（MOV R2，0）	PC→AR=03H 指向指令地址，PC+1=04H
12	11	00C048		92	取指令,将 RAM 中的指令送指令寄存器	AR=03H，RAM=92H→BUS→IR=92H
13	03	01ED83			指向操作数地址	PC→AR=04H 指向操作数地址，PC+1=05H
14	01	009001			RAM 内容送 R2	RAM(04H)=00H→BUS→R2=00H
15	02	01ED82	06		执行第 4 条指令（CMP RO，R1）	PC→AR=05H 指向指令地址，PC+1=06H
16	12	00C048		A1	取指令,将 RAM 中的指令送指令寄存器	AR=05H，RAM=A1H→BUS→IR=A1H
17	04	01A204			取源操作数	RO→BUS→DRO=07H
18	05	01B405			取目的操作数	R1→BUS→DR1=01H
19	01	618B41			RO-R1，标志位→FC	DRO-DR1=06H，FC=0
20	02	01ED82	07		执行第 5 条指令（JB LP2）	PC→AR=06H 指向指令地址，PC+1=07H
21	13	00C048		B0	取指令,将 RAM 中的指令送指令寄存器	AR=06H，RAM=B0→BUS→IR=B0H
22	25	01ED95			PC 指向转移地址存放单元	PC→AR=07H，PC+1=08H
23	07	0180C7			分支检测 FC=1?	P(3)检测 FC=1?
24	01	018001			FC=0，顺序执行	FC=0，PC 内容不变
25	02	01ED82	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	00C048		C6	取指令,将 RAM 中的指令送指令寄存器	AR=07H，RAM=C6H→BUS→IR=C6H
27	06	01A206			取源操作数→DRO	R2→BUS→DRO=00H
28	31	01B419			取目的操作数→DR1	R1→BUS→DR1=01H
29	01	919A01			R2+R1→R2	DRO+DR1→BUS→R2=01H

表5-25 微指令执行情况

30	02	01ED82	09		执行第 7 条指令（INC R1）	PC→AR=08H 指向指令地址，PC+1=09H
31	15	00C048		D1	取指令,将 RAM 中的指令送指令寄存器	AR=08H，RAM=D1H→BUS→IR=D1H
32	26	01A416			取源操作数→DR0	R1→BUS→DR0=01H
33	32	C9BA1A			取目的操作数→DR1	1→DR1
34	01	919B41			R1+1→R1	DR0+DR1→BUS→R1=02H
35	02	01ED82	0B		执行第 8 条指令（INC R1）	PC→AR=0AH 指向指令地址，PC+1=0BH
36	15	00C048		D1	取指令,将 RAM 中的指令送指令寄存器	AR=0AH，RAM=D1H→BUS→IR=D1H
37	26	01A416			取源操作数→DR0	R1→BUS→DR0=02H
38	32	C9BA1A			取目的操作数→DR1	1→DR1
39	01	919B41			R1+1→R1	DR0+DR1→BUS→R1=03H
40	02	01ED82	0C		执行第 9 条指令（JMP LP1）	PC→AR=0BH 指向指令地址，PC+1=0CH
41	16	00C048		E0	取指令,将 RAM 中的指令送指令寄存器	AR=01H，RAM=E0H→BUS→IR=E0H
42	30	01ED98			指向转移地址	PC→AR=0CH 指向转移地址，PC+1=0DH
43	01	00D181			转移地址→PC	RAM(0C)=05H→BUS→PC=05H
44	02	01ED82	06		执行第 4 条指令（CMP R0，R1）	PC→AR=05H 指向指令地址，PC+1=06H
45	12	00C048		A1	取指令,将 RAM 中的指令送指令寄存器	AR=05H，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	618B41			R0-R1，标志位→FC	DR0-DR1=04H，FC=0
49	02	01ED82	07		执行第 5 条指令（JB LP2 ）	PC→AR=06H 指向指令地址，PC+1=07H
50	13	00C048		B0	取指令,将 RAM 中的指令送指令寄存器	AR=01H，RAM=10H→BUS→IR=B0H
51	25	01ED95			PC 指向转移地址存放单元	PC→AR=07H，PC+1=08H
52	07	0180C7			分支检测 FC=1?	P(3)检测 FC=1?
53	01	018001			FC=0，顺序执行	FC=0，PC 内容不变
54	02	01ED82	08		执行第 6 条指令（ADD R1，R2）	PC→AR=07H 指向指令地址，PC+1=08H
55	14	00C048		C6	取指令,将 RAM 中的指令送指令寄存器	AR=07H，RAM=C6H→BUS→IR=C6H
56	06	01A206			取源操作数→DR0	R2→BUS→DR0=01H
57	31	01B419			取目的操作数→DR1	R1→BUS→DR1=03H
58	01	919A01			R2+R1→R2	DR0+DR1→BUS→R2=04H
59	02	...				

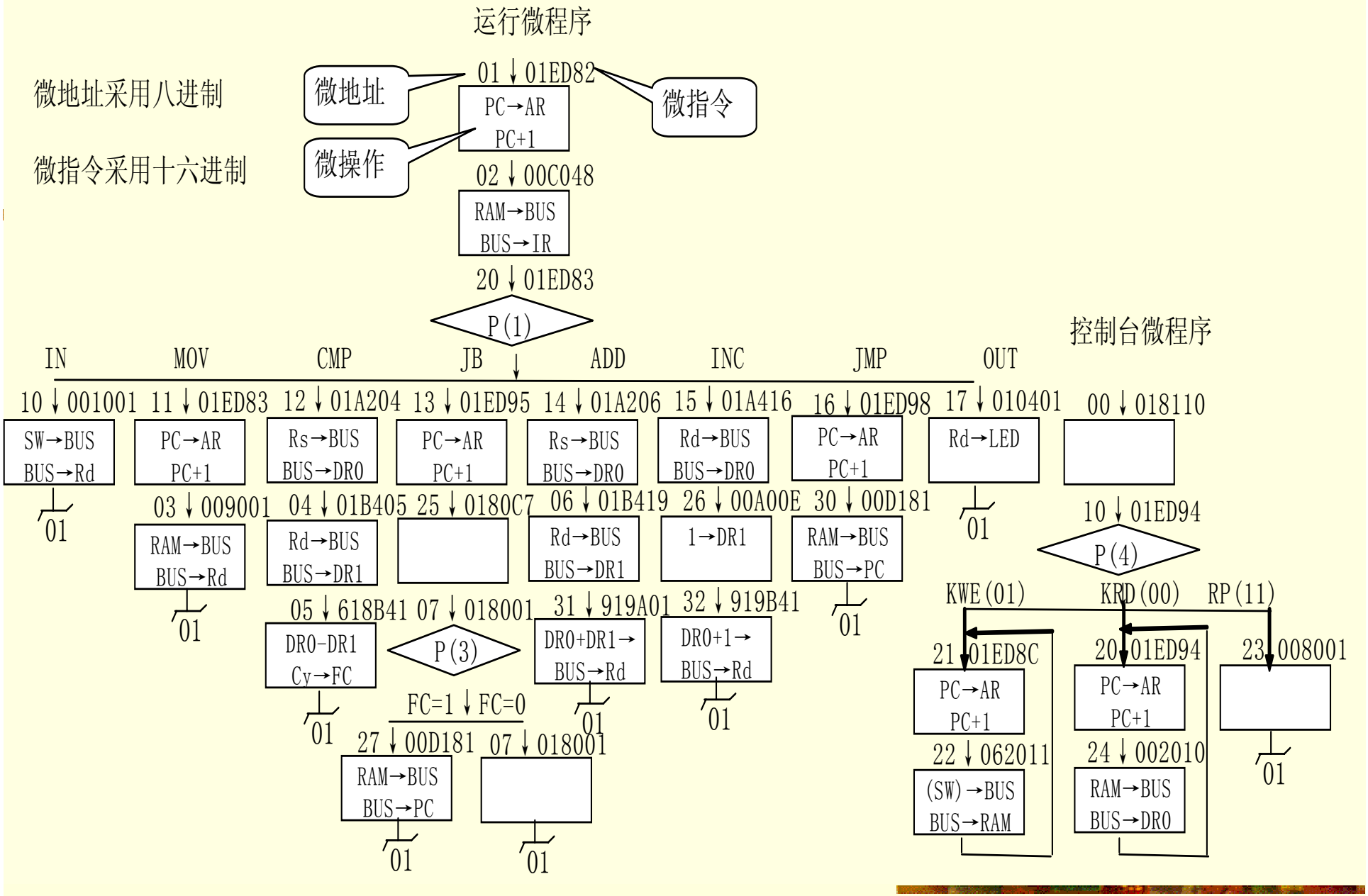


图5-39 实验5-4的微程序流程图

实验与设计

实验5-5. K8051单片机核基本系统构建和测试实验

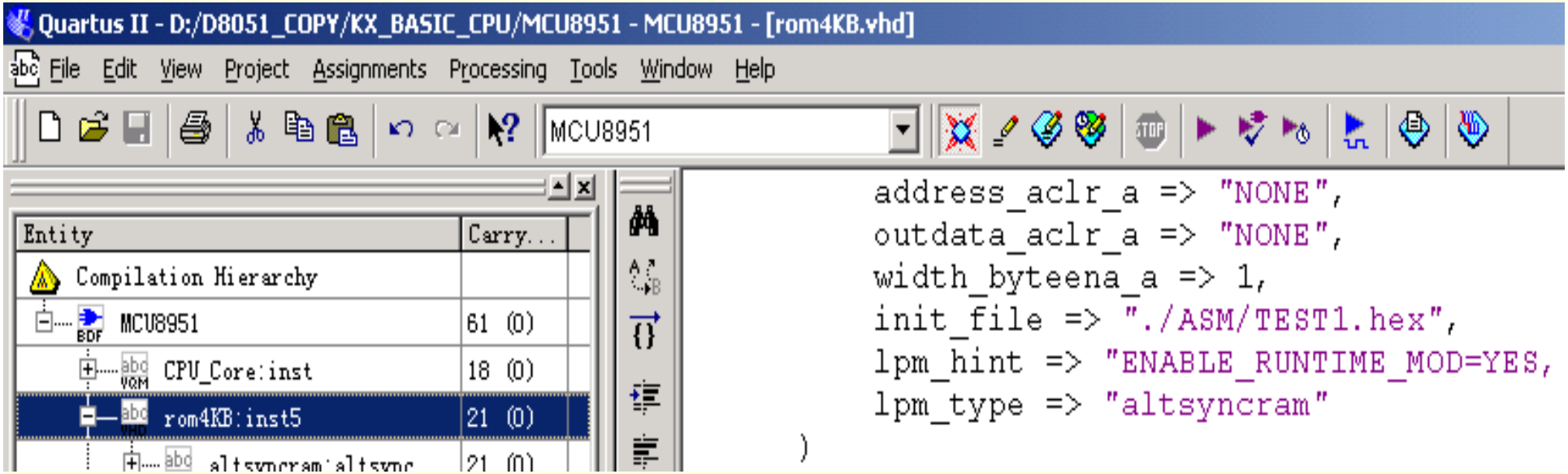


图5-40 ROM初始化文件路径

实验与设计

实验5-5. K8051单片机核基本系统构建和测试实验

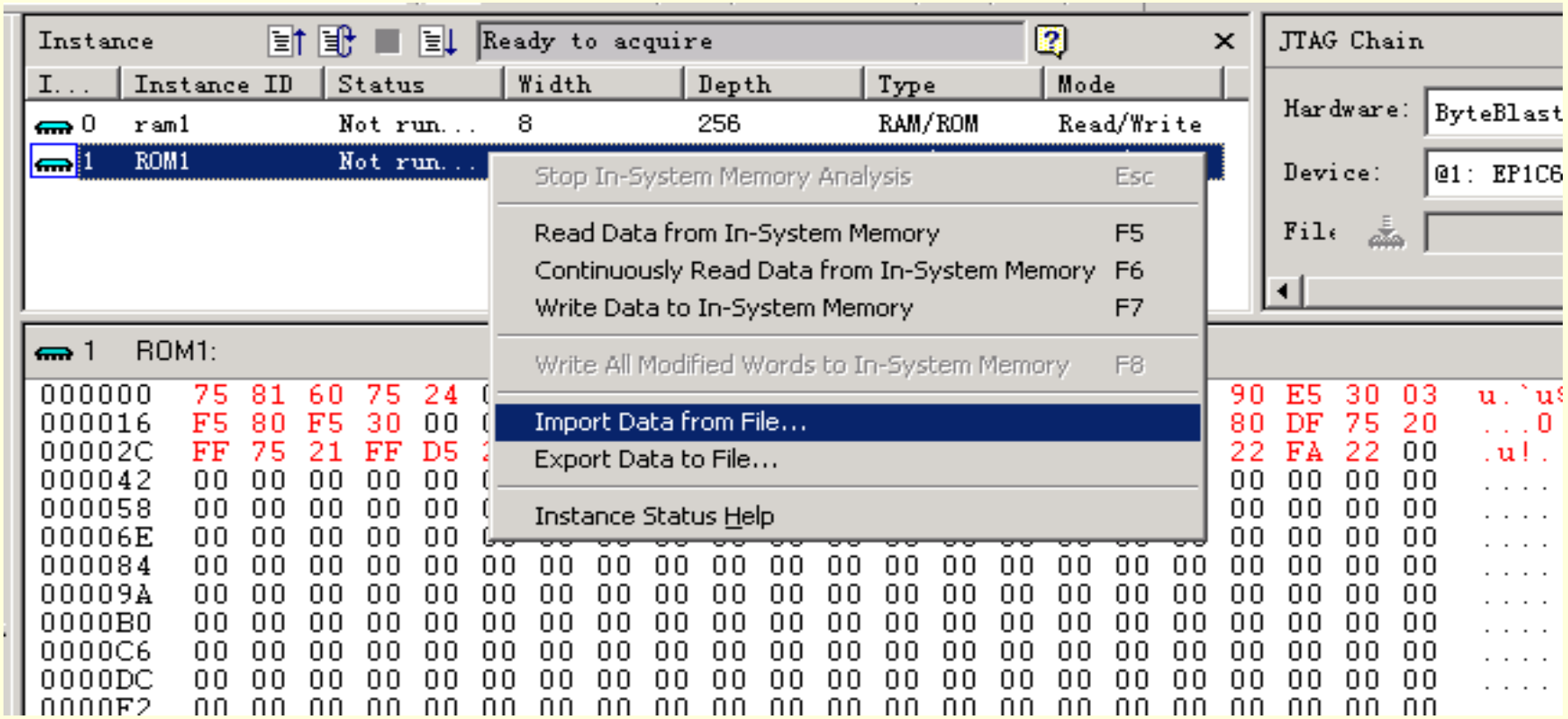


图5-41 下载汇编程序HEX代码

实验与设计

实验5-6. 基于K8051核的液晶显示与等精度频率测试系统设计

(1) 实验目的

掌握单片机核与其他硬件功能模块（等精度频率计采样模块）接口设计和调试方法，以及**K8051**控制液晶显示技术。

(2) 实验原理

此实验示例工程路径：/ **CMPUT_EXPMT/CH5_Expt/DEMO56_K8051/MCU8951**，用**QuartusII**打开工程**MCU8951**，其顶层原理图如图**5-43**所示，这是一个含有等精度频率计测试模块的**K8051**单片机系统。该模块由**VHDL**描述。单片机时钟设在**40MHz**上。本实验选择模式**3**，示例程序路径和程序名是：/ **CMPUT_EXPMT/CH5_Expt/DEMO56_K8051/ASM/PK_LCD8K.ASM**

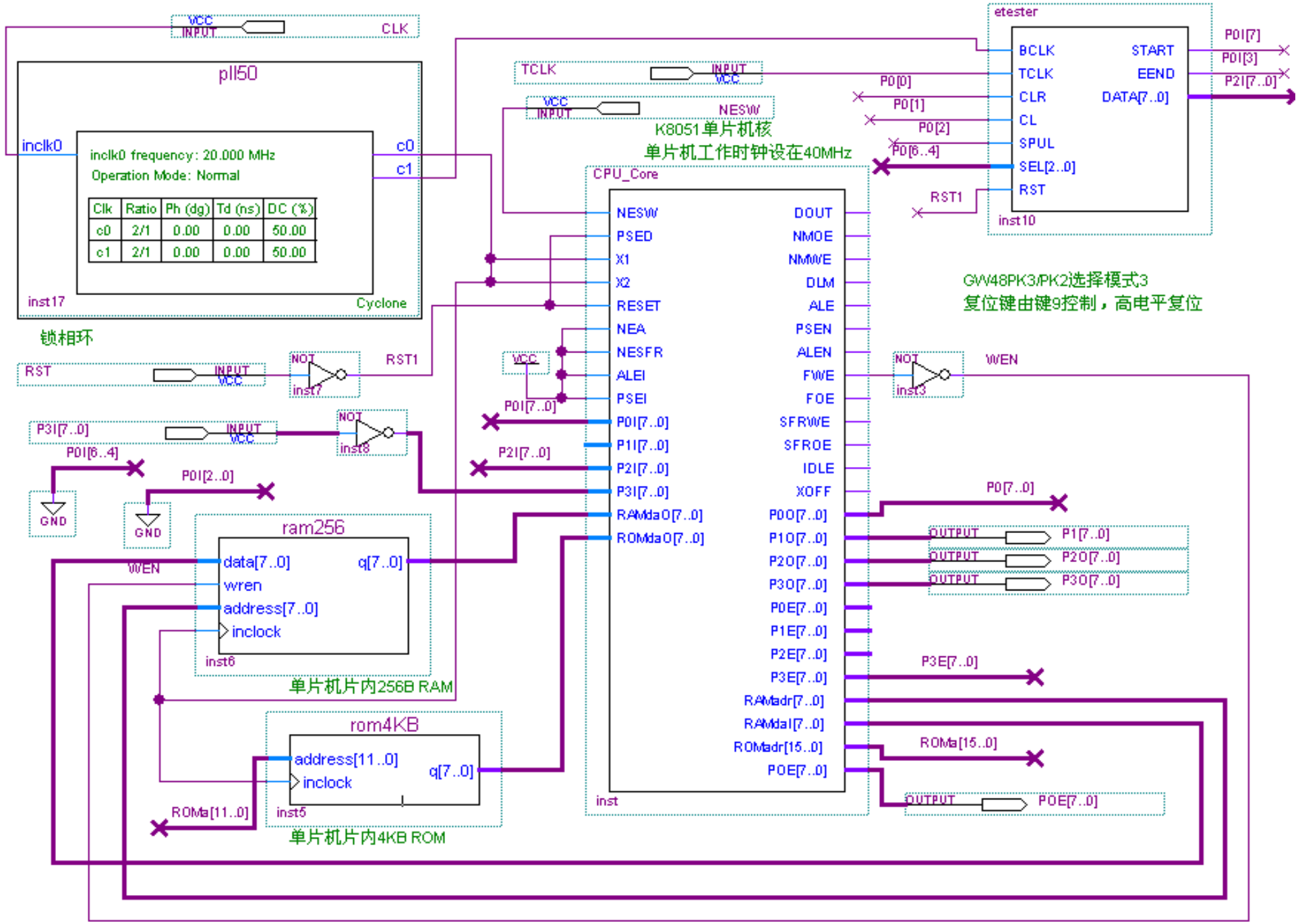


图5-42 含有等精度频率计测试模块的8051单片机系统

实验与设计

实验5-7．基于K8051核的数码管显示与等精度频率测试系统设计

此实验示例工程路径：

/CMPUT_EXPMT/CH5_Expt /DEMO57_K8051LED/MCU8951。

此设计工程中也含有等精度频率测试模块，但用数码管显示测试结果。注意顶层原理图中有一个**P3**口的双向口连接电路。单片机时钟设在**60MHz**上，等精度频率计测试模块的标准频率设在**100MHz**上。

单片机软件示例程序路径和程序名是：

/CMPUT_EXPMT/CH5_Expt /DEMO57_K8051LED/ASM/GW50M2.hex 。