
实践教学

兰州理工大学

计算机与通信学院

2021 年春季学期

计算机组成原理课程设计

题 目：	模型机设计-4
专业班级：	
姓 名：	
学 号：	
指导教师：	包仲贤
成 绩：	

摘 要

本模型机的设计主要涉及运算器、存储系统、微程序控制模块、模型机指令系统的设计。其中，运算器采用八位移位运算器，使用模型机的器件，由 4 片 74LS181 组成带有片间串行进位 16 位算数逻辑运算功能的运算器，存储器设计为移位存储器，使用了一片 74LS299 作为移位发生器；微控制器由 E2PROM 2816、8D 触发器 74LS273、4D 触发器 74LS175 和双 D 触发器 74LS74 构成；指令系统设计了 IN、OUT、STA、LDA、JMP、BZC、CLR、MOV、NOT、OR、AND、HLT 指令的格式。

关 键 词：运算器； 存储器； 控制器； 模型机

目 录

摘 要	2
第一章 设计目的及设计原理	3
1.1 设计目的	3
1.2 设计原理	3
第二章 总体设计	4
2.1 模型机数据通路图	4
2.2 硬件总体框图	4
2.3 模型机微程序流程图	6
第三章 详细设计	9
3.1 运算器的物理结构	9
3.2 存储器系统的组成与说明	11
3.3 指令系统的设计与指令格式分析	14
3.4 微程序控制器的逻辑结构及功能	16
3.5 微程序的设计与实现	21
第四章 系统调试报告	27
4.1 实验接线	27
4.2 联机读/写程序	27
4.3 运行程序	27
4.4 解决调试中遇到的问题	28
参考文献	30
设计总结	31
致谢	32

前 言

计算机组成原理是计算机科学与技术及相关专业的一门专业基础课，是工程性、技术性和实践性都非常强的一门课程，在理论教学课中学到计算机的基本组成与工作原理的基础知识的同时，还要重视理论与实践相结合，以增强我们的硬件实践动手能力，从而更深入地理解计算机的工作原理。加深我们对计算机的组成原理和指令在计算机中运行过程的理解，本次课程设计是设计一个简单计算机模型，通过微指令、微程序的设计实现计算机的基本功能、不断调试最终达到设计要求的全过程。进而更好地掌握计算机中的运算器、寄存器、译码电路、存储器、和存储微指令的控制存储器等硬件组成的相关知识，实现知识融会贯通的目的。

本次课程设计在 DVCC 系列计算机组成原理系统下进行。DVCC 实验机能很好地完成计算机硬件系统各功能部件的实验，它包括运算器部件、控制器部件、主存储器部件、总线和几种最重要的外设接口实验，包括中断、定时计数器、输入/输出接口等，本次课程设计是在相应软件的配合下，将各功能部件有机的结合起来，完成计算机整机的实验，建立带有 8 移位运算指令的整机模型。通过 DVCC 系列计算机顺利完成本次课程设计任务。

第一章 设计目的及设计原理

1.1 设计目的

该设计根据计算机组成原理课程所学知识，设计、开发一套简单的模型计算机。

通过对一个简单计算机的设计，以达到对计算机的基本组成、部件的功能与设计、微程序控制器的设计、微指令和微程序的编制与调试等过程有更深入的了解，加深对理论课程的理解。通过模型机的设计和调试，连贯运用计算机组成原理课程学到的知识，建立计算机整机概念，加深计算机时间和空间概念的理解。

1.2 设计原理

部件实验过程中，各部件单元的控制信号是人为模拟产生的，而本次设计将能在微程序控制下自动产生各部件单元控制信号，实现特定指令的功能。

计算机数据通路的控制将由微程序控制器来完成，CPU 从内存中取出一条机器指令到指令执行结束的一个指令周期全部由微指令组成的序列来完成，即一条机器指令对应与一个微程序。

本设计采用 12 条机器指令：IN（输入）、ADD（二进制加法）、SUB（二进制减法）、LDA（取数）、STA（存数）、OUT（输出）、JMP（无条件转移）、BZC（为零或有进位转移）、MOV（数据传送）、CLR（位清零）、NOT（逻辑非）、OR（逻辑或）、AND（逻辑与）、HLT（停机指令）。

为了向 RAM 中装入程序和数据，检查写入是否正确，并能启动程序执行，还必须设计三个控制台操作微程序。

存储器读操作（KRD）：拨动总清开关 CLR 后，控制台开关 SWB、SWA 为“00”时，按 START 微动开关，可对 RAM 连续手动读操作。

存储器写操作（KWE）：拨动总清开关 CLR 后，控制台开关 SWB、SWA 为“01”时，按 START 微动开关，可对 RAM 进行连续手动写入。

启动程序：拨动总清开关 CLR 后，控制台开关 SWB、SWA 置为“11”时，按 START 微动开关，即可转入到第 01 号“取址”微指令，启动程序运行。

第二章 总体设计

2.1模型机数据通路图

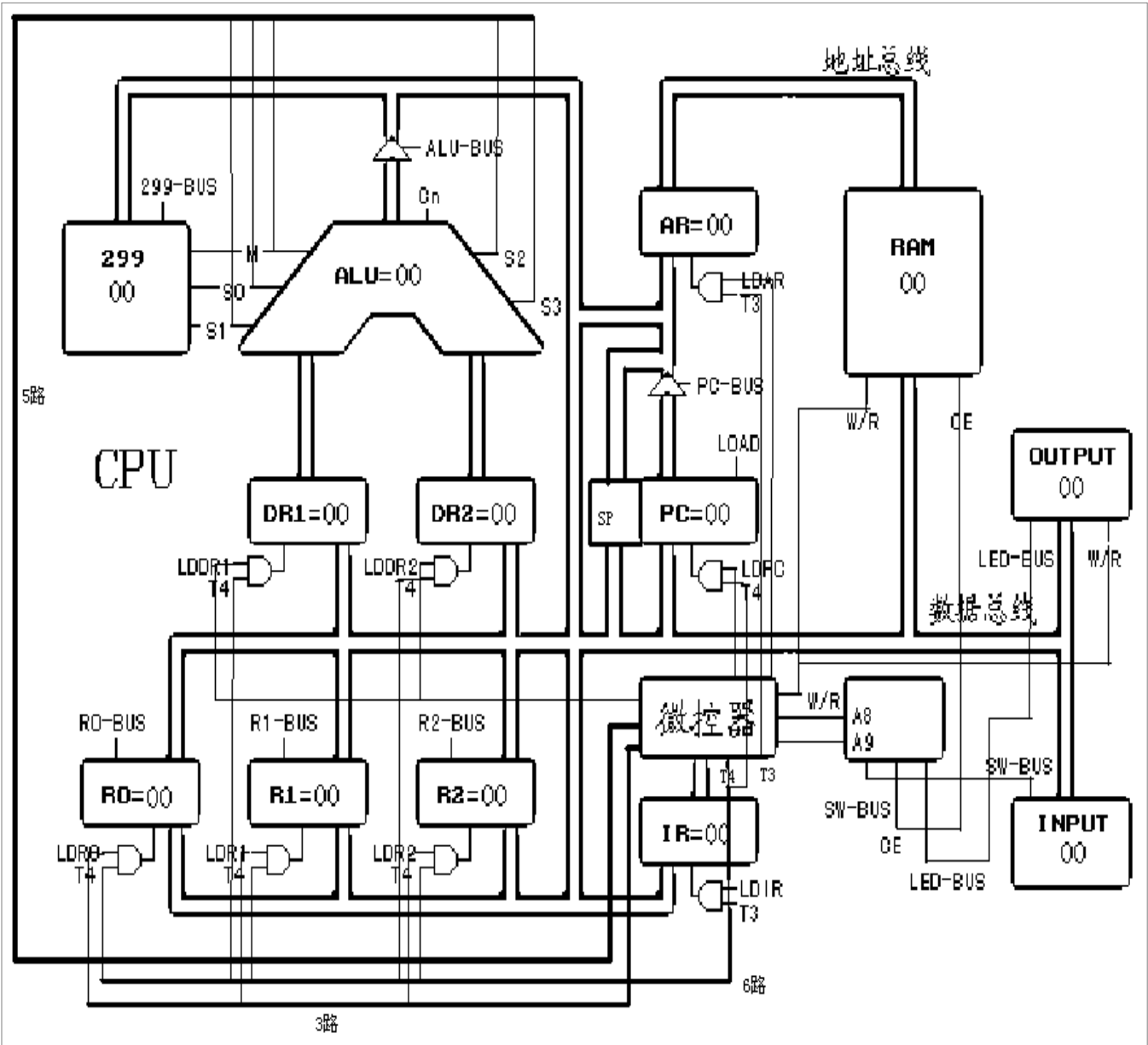


图 2-1-1 数据数据通路图如图

该模型机数据通路图如图 2-1-1 所示。其运行的主要主要分为两个阶段：取指令和执行指令。所有指令的取指令阶段均相同，而执行阶段不同的指令则有所差别。取指令过程如下：首先程序计数器PC中装入第一条指令的地址；然后PC的内容装入地址寄存器AR；然后程序计数器的内容自动加 1，为执行下一条指令做准备；接着地址寄存器的内容放到地址总线上；从而使存储单元的内容（即指令）传送到数据缓冲寄存器DR；然后DR的内容传送到指令寄存器IR，并对指令进行译码。执行阶段以加法指令ADD为例：操作控制器OC送出控制信号到通用寄存器，选择R1 作为源寄存器，R0 作为目标寄存器；将R0 的内容送入DR1，将R1 的内容送入DR2；OC送出控制信号，打开ALU，指定ALU做DR1 (R1) 和DR2 (R2) 的加法操作；OC送出控制命令，打开ALU输出三态，运算结果放到DBUS上；OC送出控制信号，将DBUS上的数据打入DR，并将ALU产生的进位信号保存到PSW中。而其它指令的执行与之类似，不再赘述。

2.2 硬件总体框图

该模型机主要由运算器，控制器，存储器，输入设备，输出设备五大部分组成。其硬件框图如图 2-2-1 所示。

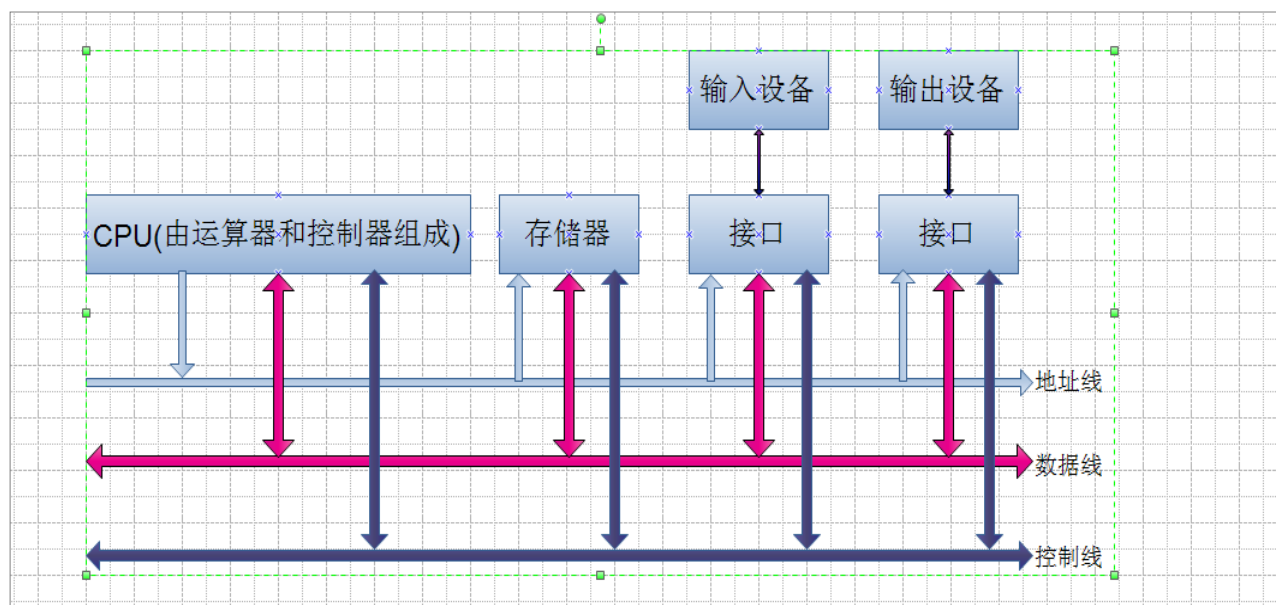


图 2-2 硬件总体框图

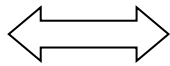
累加寄存器、数据缓冲寄存器和状态条件寄存器组成，它是数据加工处理部件，并受控制器的控制。其主要完成各种算术运算和逻辑运算。

控制器：根据指令操作码和时序信号，产生各种操作控制信号，以便正确地建立数据通路，从而完成取指令和执行指令的控制。

存储器：作为计算机的记忆部件，用于存放程序和数据。

输入设备：本模型机以八个拨动开关作为输入设备，主要完成信息的输入。

输出设备：本系统以数码管作为输出设备，主要用于显示运算的结果。



2.3模型机微程序流程图

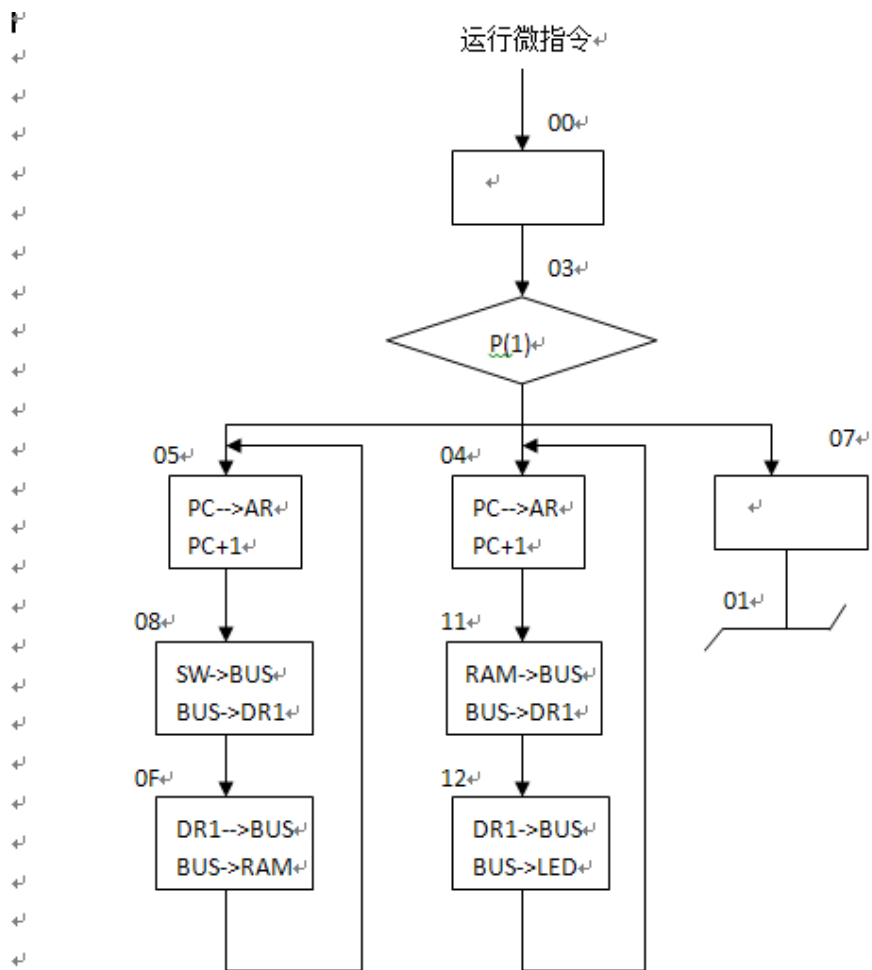
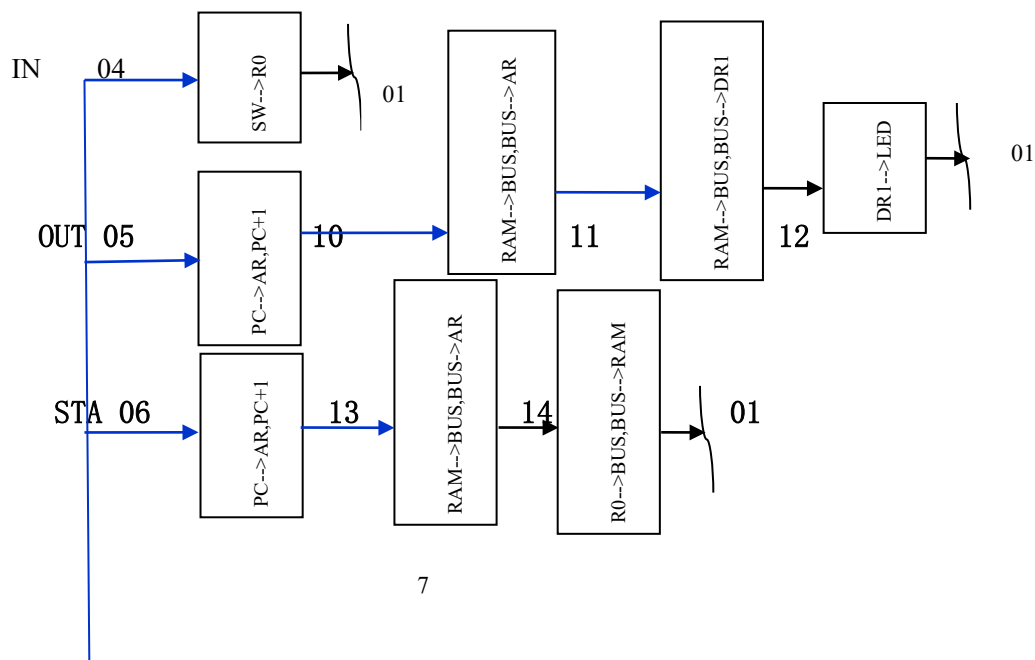
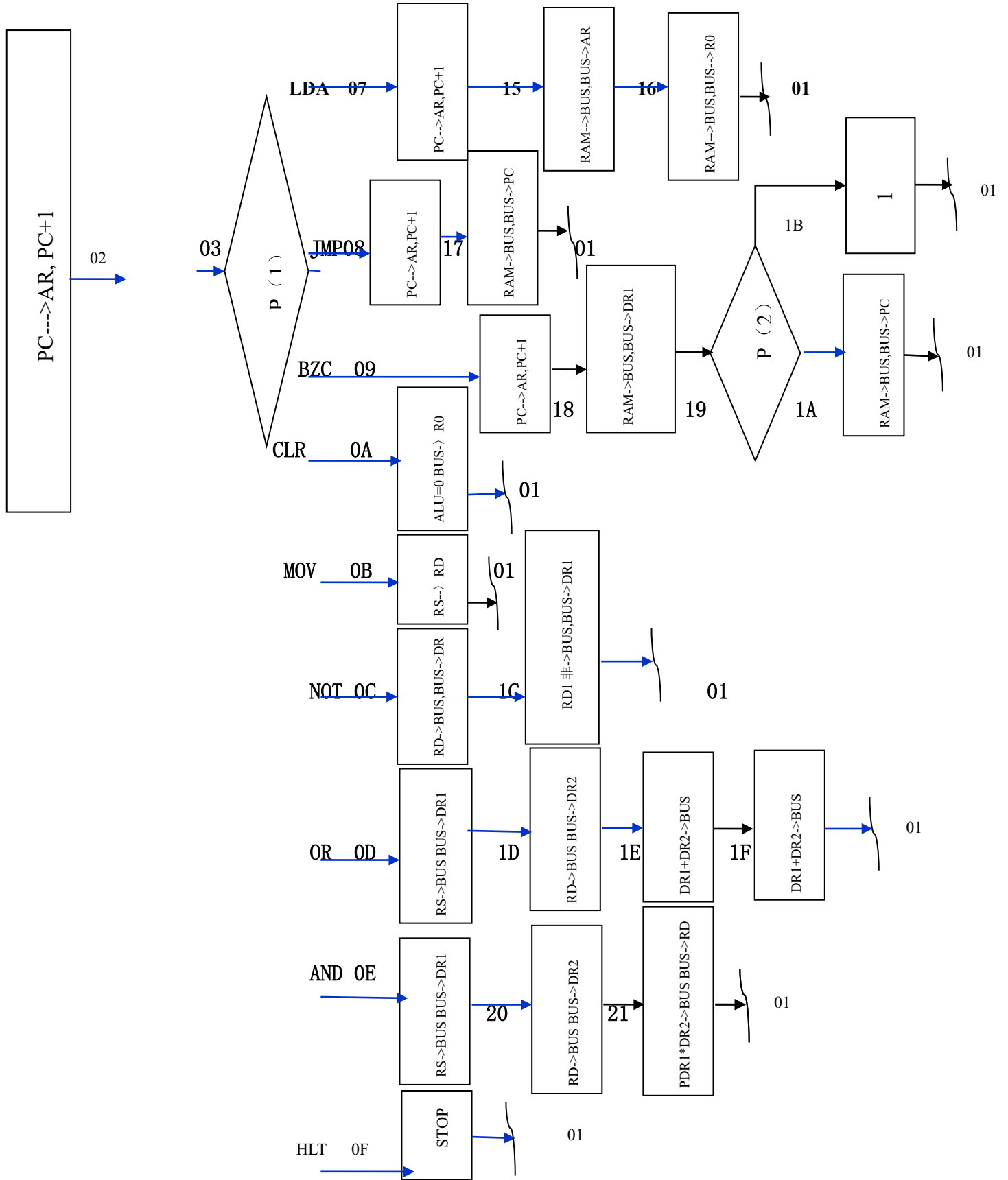


图 2-3-1 微程序流程图控制部分





第三章 详细设计

3.1 运算器的物理结构

3.1.1 运算器功能：加工信息包括算术运算和逻辑运算。

3.1.2 设计带有片间串行进位 16 位算数逻辑运算功能的运算器

此次设计的 16 位运算器由四片 74LS181 以并/串形成 16 位字长的 ALU 构成。低 8 位运算器的输出经过一个三态门 74LS245 (U33) 到 ALUO1 插座，实验时用 8 芯排线和内部数据总线 BUSD0~D7 插座 BUS1~6 中的任一个相连，低 8 位数据总线通过 LZD0~LZD7 显示灯显示；高 8 位运算器的输出经过一个三态门 74LS245 (U33') 到 ALUO1' 插座，实验时用 8 芯排线和高 8 位数据总线 BUSD8~D15 插座 KBUS1 或 KBUS2 相连，高 8 位数据总线通过 LZD8~LZD15 显示灯显示；参与运算的四个数据输入端分别由四个锁存器 74LS273 (U29、U30、U29'、U30') 锁存，实验时四个锁存器的输入并联后用 8 芯排线连至外部数据总线 EXD0~D7 插座 EXJ1~EXJ3 中的任一个；参与运算的数据源来自于 8 位数据开并 KD0~KD7，并经过一三态门 74LS245 (U51) 直接连至外部数据总线 EXD0~EXD7，输入的数据通过 LD0~LD7 显示。

3.1.3 ALU 控制电路

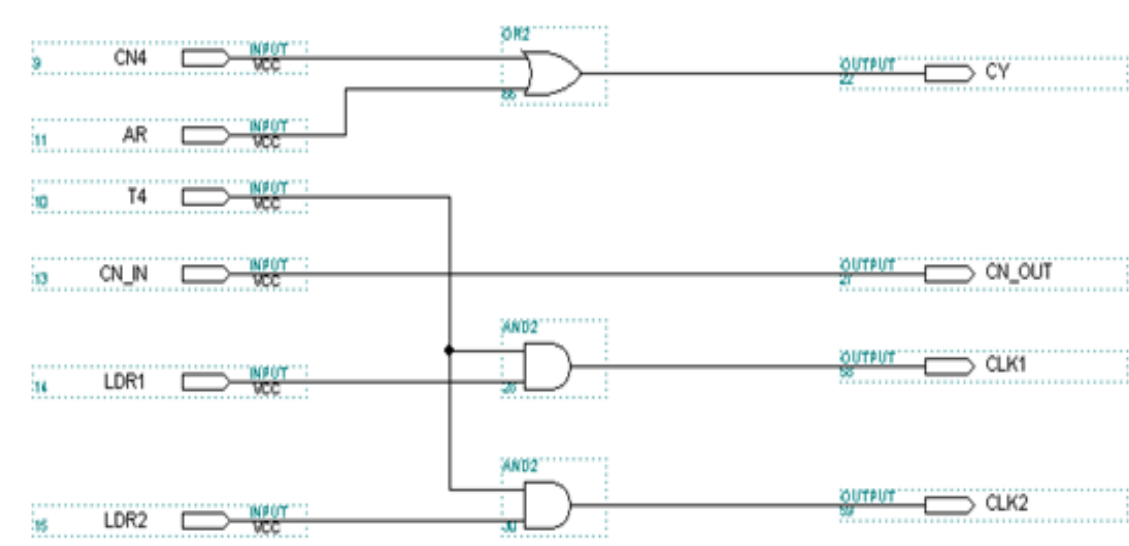


图 3-1-1 16 位逻辑运算器数据通路图

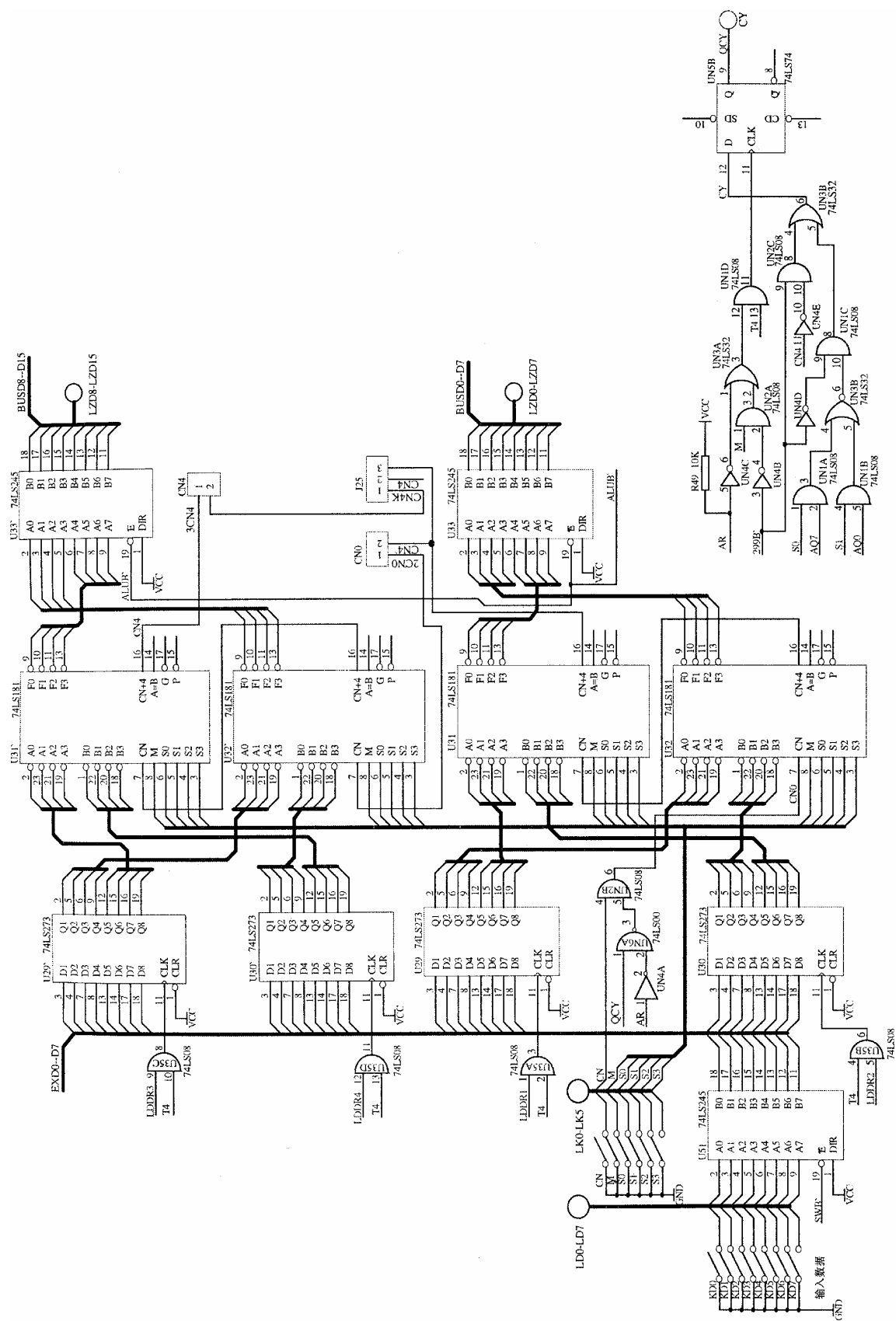


图 3-1-2 运算器内部电路图

3.1.4 移位运算器实验原理图

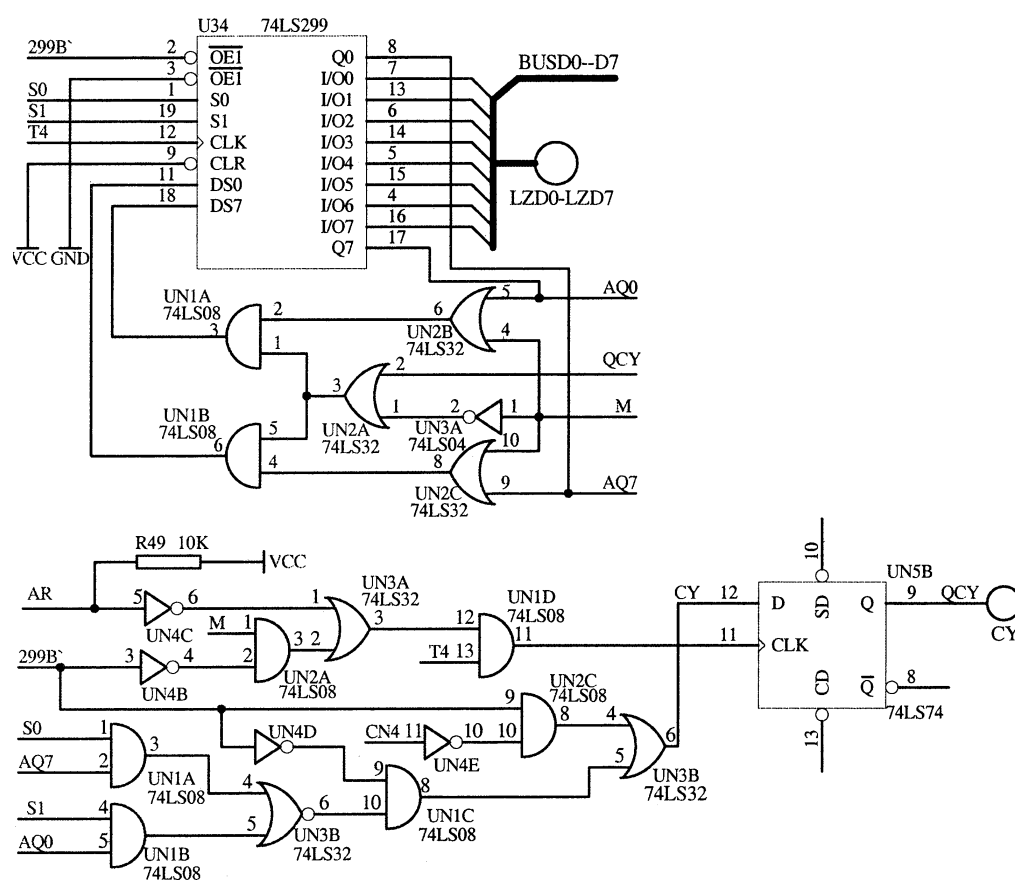


图 3-1-3 移位运算器原理图

3.2 存储器系统的组成与说明

3.2.1 存储器相关知识

- 功能：存储信息。
- 组成(主存储器)：见图 3-2 存储器单元电路图
- 存储体：存放信息的实体。
- 寻址系统：对地址码译码，选择存储单元。
- 读/写线路和数据寄存器：完成读/写操作，暂存读/写数据。
- 控制线路：产生读/写时序，控制读/写操作。
- 本模型机存储器由一片 6264 组成，实际使用容量为 256 字节，6264 有四个控制线:CS1 第一片选线，CS2 第二片选线，OE 读线，WE 写线，功能表见表 3-2-1

工作方式	I/O		输入		
	DI	DO	/OE	/WE	/CS1
非选择	X	HIGH-Z	X	X	H
读出	HIGH-Z	DO	L	H	L
写入	DI	HIGH-Z	H	L	L
写入	DI	HIGH-Z	L	L	L
选择	X	HIGH-Z	H	H	L

表 3-2-1 6264 功能表

3.2.2 存储器单元电路图

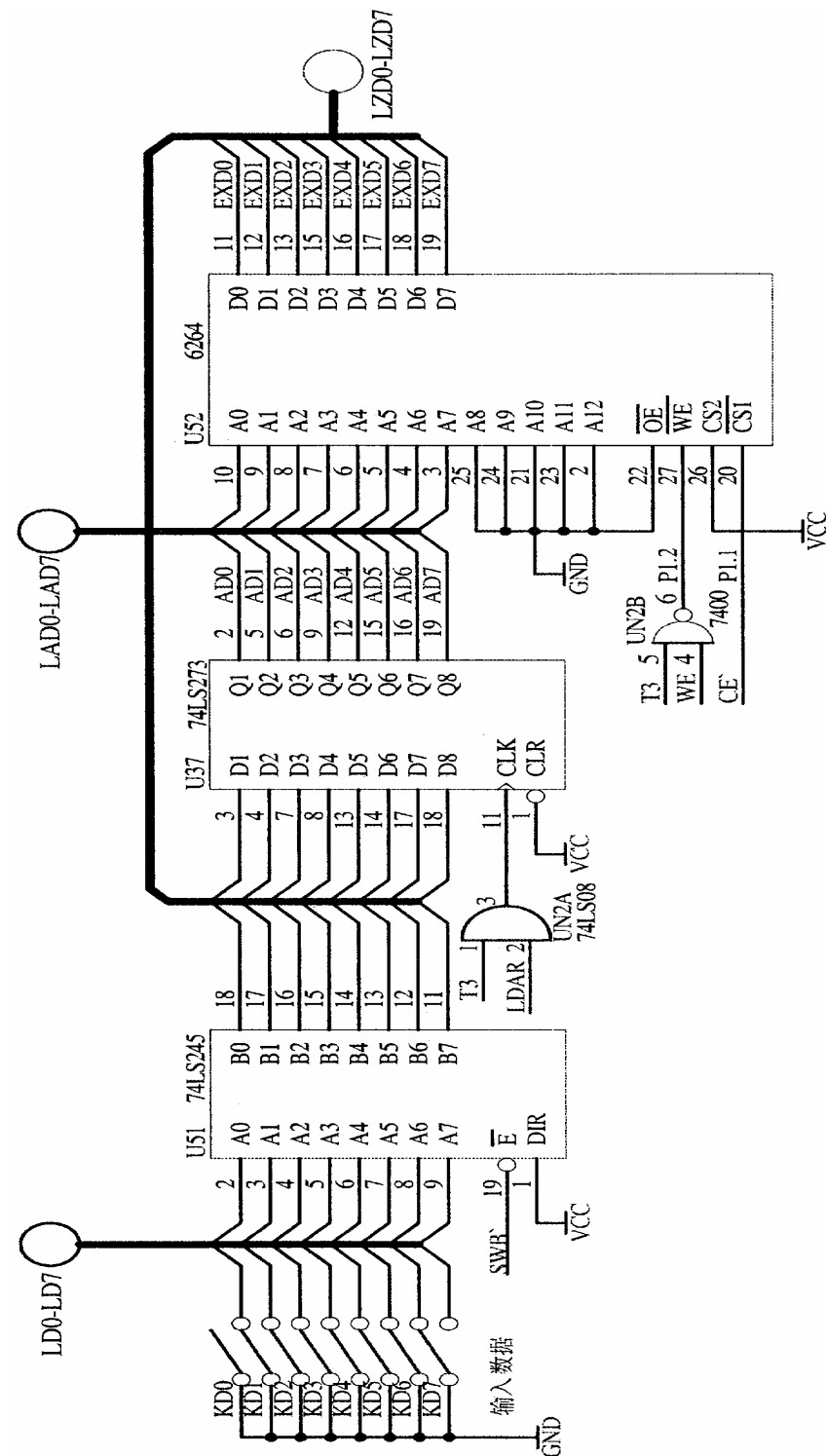


图 3-2-1 存储器单元电路图

3.2.3 存储器输入输出时序

形成时钟脉冲信号 T3 的方法如下：在时序电路模块中有两个二进制开关“运行控制”和“运行方式”。将“运行控制”开关置为“运行”状态、“运行方式”开关置为“连续”状态时，按动“运行启动”开关，则 T3 有连续的方波信号输出，此时调节电位器 W1，用示波器观察，使 T3 输出实验要求的脉冲信号；本实验中“运行方式”开关置为“单步”状态，每按动一次“启动运行”开关，则 T3 输出一个正单脉冲，其脉冲宽度与连续方式相同。

3.3 指令系统的设计与指令格式分析

1. 数据格式

复杂模型机采用 7 位定点补码表示法表示数据，字长 8 位，第 7 位为符号位。

2. 指令格式

模型机设计四大类指令共 12 条，其中包括算术逻辑指令、I/O 指令、存数指令、取数指令、转移指令和停机指令等。存取数指令与转移指令设计成二字节指令，其他指令都设计成一字节指令。

分类如下：

- 算术逻辑指令：ADD、SUB、NOT、OR、AND
- I/O 指令：IN、OUT
- 存取数指令：LDA、STA
- 转移指令：JMP、BZC
- 停机指令：HLT
- 数据传送指令：MOV

(1) 访存指令及转移指令

设计 4 条访存指令及转移指令采用二字节表示，在指定机器指令的操作码中的 M=00 时，为直接寻址，M=01 时做间接寻址，M=10 时，变址寻址（在此系统中，已指定 R2 作为固定的变址寄存器），当 M=11 时为相对寻址。指令格式为：

7 654	3 2	1 0
操作码	M	RD
D		

表 3-3-1 访问指令及转移指令

其中，RD 为目的寄存器地址（LDA、STA 指令使用）。D 为位移量（正负均可），M 为寻址模式，本模型机规定变址 RI 指定为寄存器 R2。

(2) 算术逻辑指令

设计 5 条算术逻辑指令采用单字节表示，寻址方式采用寄存器直接寻址，其格式如下：

7 6 5 4	3 2	1 0
操作码	源寄存器	目的寄存器

表 3-3-2 算术逻辑指令格式

(3) I/O 指令

I/O 指令采用单字节指令，其格式如下：

7654	32	10
操作码	addr	RD

表 3-3-3 I/O 指令

其中，addr=01 时，选中输入数据开关组 KD0~KD7 作为输入设备，addr=10 时，选中 2 位数码管作为输出设备。

汇编符号	指令格式	功能
LDA M, D, rd	0000 M rd	E→rs
STA M, D, rd	0001 M rd	rd→E
JMP M, D	0010 M rd	E→PC
BZC M, D	0011 M rd	当 cy=1 时 E→PC
ADD rs, rd	0100 rs rd	Rs+rd→rd
SUB rs, rd	0101 rs rd	Rs-rd→rd
AND rs, rd	0110 rs rd	rs∧rd→rd
OR rs, rd	0111 rs rd	rs∨rd→rd
NOT rd	1000 rd	/rd → rd
IN addr, rd	1001 01 rd	addr→rd
OUT addr, rd	1010 10 rd	rd→addr
CLR rd	1011 00 rd	0→rd
MOV rs, rd	1100 rs rd	rs→rd
HLT	1101 00 00	停机

表 3-3-4 复杂指令系统

3.4 微程序控制器的逻辑结构及功能

3.4.1 微控制器逻辑结构及微程序控制器组成图

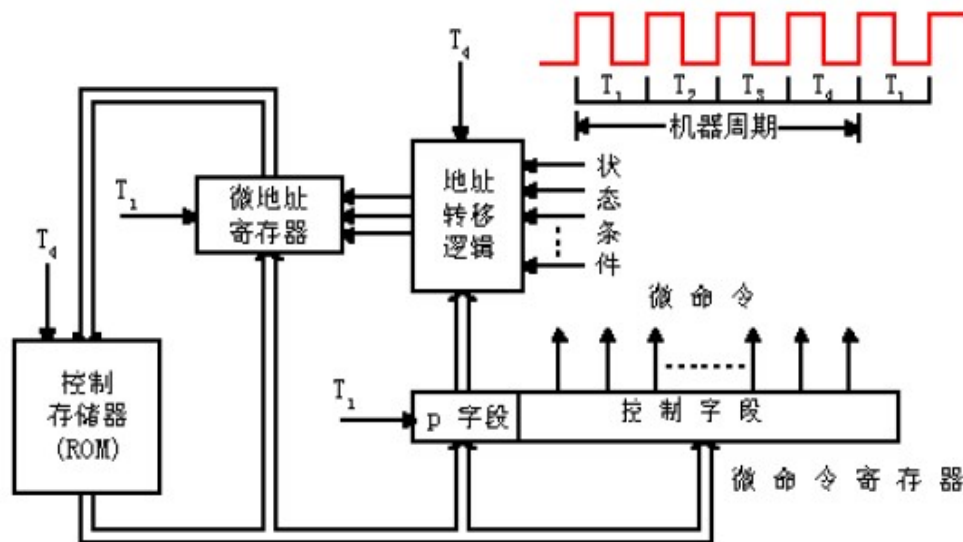


图 3-4-1 微控制器逻辑结构图

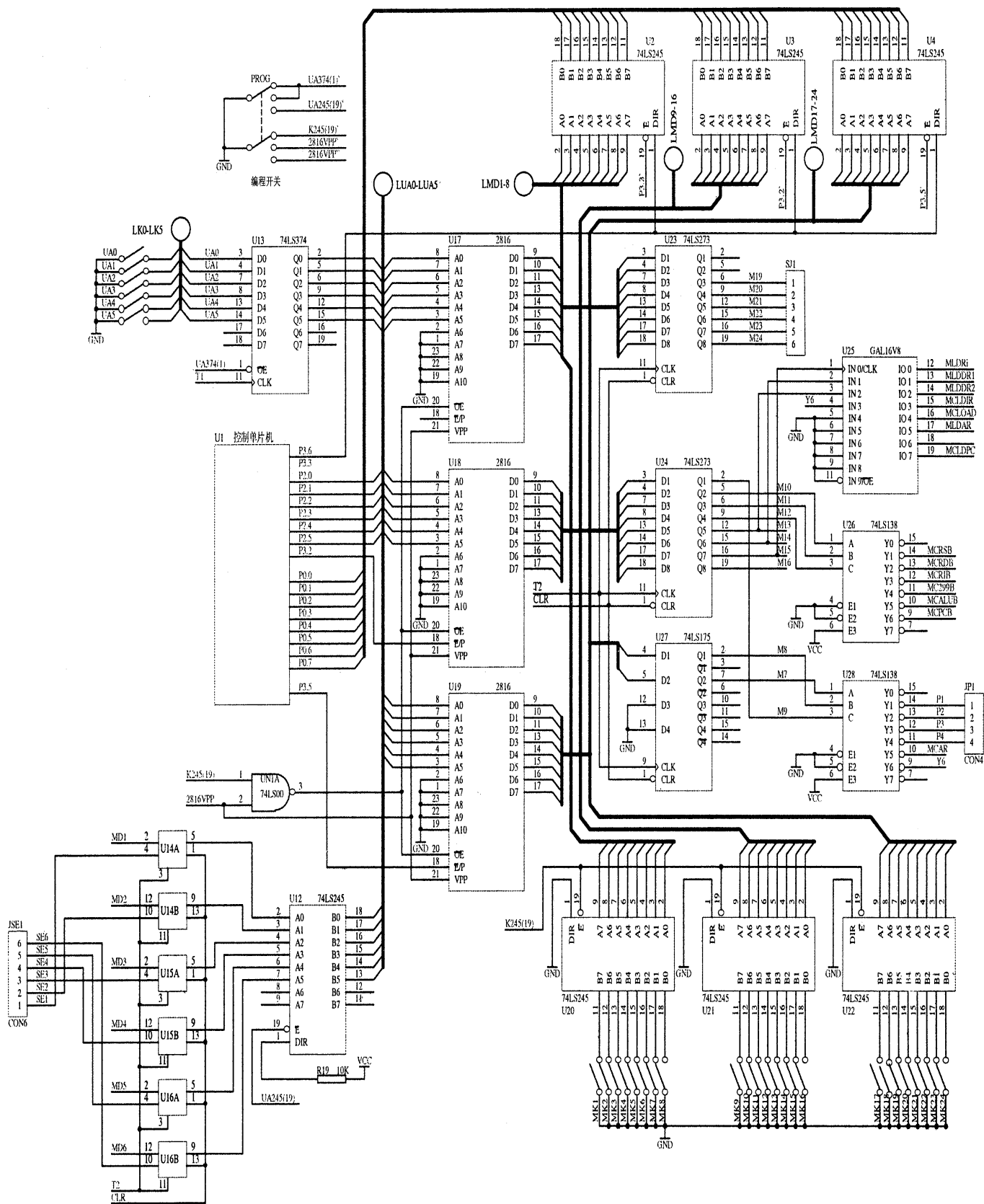


图 3-4-2 微程序控制器组成图

微程序控制器的组成如图 3-4-2 所示，其中控制存储器采用 3 片 E2PROM 2816 芯片，具有掉电保护功能，微命令寄存器 18 位，用两片 8D 触发器 74LS273（U23、U24）和一片 4D 触发器 74LS175（U27）组成。微地址寄存器 6 位，用三片正沿触发的双 D 触发器 74LS74（U14~U16）组成，它们带有清“0”端和预置端。在不进行判别测试的情况下，T2 时刻打入微地址寄存器的内容即为下一条微指令地址。当 T4 时刻进行测试判别时，转移逻辑满足条件后输出的负脉冲通过强置端将某一触发器置为“1”状态，完成地址修改。

在本实验电路中设有一个编程开关，它具有三种状态：写入、读出、运行。当处于“写”状态时，学生可根据微地址和微指令格式将微指令二进制代码写入到控制存储器 2816 中。当处于“读”时，可以对写入控制存储器中的二进制代码进行验证，从而可判断写入的二进制代码是否正确。当处于“运行”状态时，只要给出微程序的入口微地址，则可根据微程序流程图自动执行微程序。图中微地址寄存器输出端增加了一组三态门（U12），目的是隔离触发器的输出，增加抗干扰能力，并用来驱动微地址显示灯。

3.4.2 微控制器功能：产生控制命令(微命令)，控制全机操作。

- 微命令产生方式(指令执行控制方式):
- 组合逻辑控制方式：由组合逻辑电路产生微命令。
- 微程序控制方式：由微指令产生微命令。

3.4.3 时序信号

用双踪示波器（或用 PC 示波器功能）观察方波信号源的输出。方法如下：将“运行控制”开关置为“运行”、“运行方式”开关置为“连续”。按动“启动运行”开关，从示波器上可观察到 TS1（J20）、TS2（J21）、TS3（J22）、TS4（J23）各点的波形，画出其波形如图 3-4-3 所示，并标注测量所得的脉冲宽度。

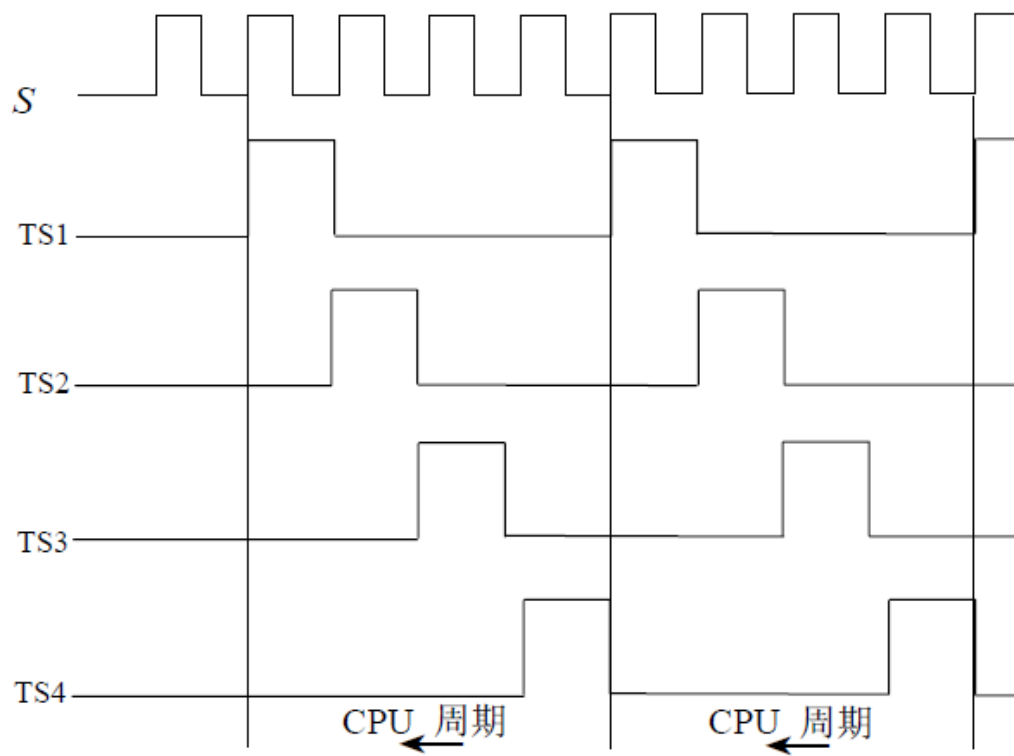


图 3-4-3 时序信号图

可调的方波信号。学生可根据实验要求自行选择方波信号的频率及脉宽。为了便于控制程序的运行，时序电路发生器设计了一个启停控制触发器 UN1B，使 TS1~TS4 信号输出可控。图中“运行方式”、“运行控制”、“启动运行”三个信号分别是来自实验机上三个开关。当“运行控制”开关置为“运行”，“运行方式”开关置为“连续”时，一旦按下“启动运行”开关，运行触发器 UN1B 的输出 QT 一直处于“1”状态，因此时序信号 TS1~TS4 将周而复始地发送出去；当“运行控制”开关置为“运行”，“运行方式”开关置为“单步”时，一旦按下“启动运行”开关，机器便处于单步运行状态，即此时只发送一个 CPU 周期的时序信号就停机。利用单步方式，每次只运行一条微指令，停机后可以观察微指令的代码和当前微指令的执行结果。另外，当实验机连续运行时，如果“运行方式”开关置“单步”位置，也会使实验机停机

3.5 微程序的设计与实现

3.5.1 设计复杂模型机的监控软件

监控程序详细如下：

```

SP00 44      IN 01 R0
SP00 46      IN 01 R1
SP02 98      ADD R0,R1
SP03 81      MOV R0,R1
SP04 F5      CLR R1
SP05 0C      BZC 00,00

```

SP06 00

微指令格式：

位→	24	23	22	21	20	19	18	17	16
功能→	S3	S2	S1	S0	M	CN	WE	B1	B0
位→	15	14	13	12	11	10	9	8	7
功能→	A			B			C		
位→	6	5	4	3	2	1			
功能→	uA5	uA4	uA3	uA2	uA1	uA0			

表 3-5-1

字段 24~19 是运算器 74LS181 芯片的功能控制端，详见表 3-5-2 74LS181 的功能表。

4 位 ALU	S3S2 S1			M = 0（算数运算）		M = 1 （逻辑运算）
	S0			C n=1 无进位	C n=0 有进位	
	0	0	0 0	F=A	F=A 加 1	$F=\overline{A}$
	0	0	0 1	F=A+B	F=（A+B）加 1	$F=\overline{\left(A+B\right)}$
	0	0	1 0	$F=A+\overline{B}$	F=（A+ \overline{B} ）加 1	$F=\overline{A} * B$
	0	0	1 1	F=减 1	F=0	F=0
	0	1	0 0	F=A 加（A* \overline{B} ）	F=A 加（A* \overline{B} ）加 1	$F=\overline{\left(A * B\right)}$
	0	1	0 1	F=（A+B）加（A* \overline{B} ）	F=(A+B)加(A* \overline{B})加 1	$F=\overline{B}$
	0	1	1 0	F=A 减 B 减 1	F=A 减 B	$F=\left(A \oplus B\right)$
	0	1	1 1	$F=\left(A * \overline{B}\right)$ 减 1	$F=\left(A * \overline{B}\right)$	$F=\left(A * \overline{B}\right)$
	1	0	0 0	F=A 加 A*B	F=A 加 A*B 加 1	$F=\overline{A}+B$
	1	0	0 1	F=A 加 B	F=A 加 B 加 1	$F=\overline{\left(A \oplus B\right)}$
	1	0	1 0	$F=\left(A+\overline{B}\right)$ 加 A*B	$F=\left(A+\overline{B}\right)$ 加 A*B 加 1	F=B
	1	0	1 1	F=A*B 减 1	F=A*B	F=A*B
	1	1	0 0	F=A 加 A	F=A 加 A 加 1	F=1
	1	1	0 1	F=（A+B）加 A	F=(A+B)加 A 加 1	$F=A+\overline{B}$
	1	1	1 0	$F=\left(A+\overline{B}\right)$ 加 A	$F=\left(A+\overline{B}\right)$ 加 A 加 1	F=A+B
	1	1 1	1	F=A 减 1	F=A	F=A

表 3-5-2 74LS181 的功能表

299B	S1	S0	M	功 能
0	0	0	任 意	保持
0	1	0	0	循环右移
0	1	0	1	带进位循环右移
0	0	1	0	循环左移
0	0	1	1	带进位循环左移
任意	1	1	任 意	装数

表 3-5-3 74LS299 功能表

字段 17 和 16 为对外部设备（RAM, OUTPUT, INPUT）地址进行译码，具体如下表：

B1	B0	选中的外设
0	0	选中 INPUT
0	1	选中 RAM
1	0	选中 OUTPUT
1	1	外部设备都不选中

表 3-5-4 外部设备选择表

A、B、C 字段功能分别说明如下：

1 5	14	13	选择	说明
0	0	0	NULL	空操作
0	0	1	LDRi	寄存器输入选中，具体选择由 IR 的最低 2 位决定 R0, R1, R2
0	1	0	LDDR1	暂存器 DR1 被选中
0	1	1	LDDR2	暂存器 DR2 被选中
1	0	0	LDIR	指令寄存器 IR 被选中
1	0	1	LOAD	总线数据直接装载到 PC 计数器
1	1	0	LDAR	地址寄存器 AR 被选中

表 3-5-5 A 字段

12	11	10	选择	说明
0	0	0	NULL	空操作
0	0	1	RS-B	源寄存器输出选中；R0、R1、R2 由 IR 的 3, 4 位决定
0	1	0	RD-B	目的寄存器输出选中；R0、R1、R2 由 IR 的最低 2 位决定
0	1	1	RI-B	变址寄存器选中，本机固定为 R2
1	0	0	299-B	移位寄存器输出选中
1	0	1	ALU-B	逻辑运算单元结果输出
1	1	0	PC-B	PC 计数器输出

表 3-5-6 B 字段

9	8	7	选择	说明
0	0	0	NULL	空操作
0	0	1	P(1)	分支判断 1 和 IR 的高四位作为测试条件；可分 16 个分支
0	1	0	P(2)	分支判断 2 和 IR 的 3, 4 位作为测试条件；可分 4 个分支
0	1	1	P(3)	分支判断 3 和 CY 或 ZI 作为测试条件，有两个分支
1	0	0	P(4)	分支判断 4 和开关 SWB, SBA 作为测试条件，4 个分支；用于控制台控制区(读程序，写程序和运行程序)
1	0	1	AR	进行算术运算时是否影响进位和判零标志的控制位。选中时进行带进位运算。
1	1	0	LDPC	为 PC 计数信号选中

表 3-5-7 C 字段

字段 6~1 为下一步微地址，其决定执行哪条微程序。

指令的后续地址的产生方法是:在没有跳转的指令中后六位就是下一条微指令的入口地址。在有跳转的指令根据跳转的条件微控制器根据相应的条件和地址将下地址直接送到为控制器的地址强制端得到下一条指令的地址。

微程序是按顺序在在为控存中存放在系统初始化的是时候指令是从 00H 地址开始的 00H 地址中存放的是一条跳转指令直接可以跳转到 01H 的中存放的就是真正在控制程序功能的指令。机器就根据指令一条的执行。在微控制器的控制下让机器根据指令的来进行有条不紊的工作。

微指令的入口地址的形成是根据机器指令的高四位进行判断后得出的。每一条微指令都对应相应的一个地址。地址的编制和每一微指令是一一对应。不存在冲突。二进制代码表如表 3-5-8 所示。

微地址	S3 S2 S1 S0 M CN WE B1 B0	A	B	C	UA5...UA0	十六进制
00	0 0 0 0 0 1 0 1 1	0 0 0	0 0 0	1 0 0	0 1 0 0 0 0	058110
01	0 0 0 0 0 1 0 1 1	1 1 0	1 1 0	1 1 0	0 0 0 0 1 0	05ED82
02	0 0 0 0 0 1 0 0 1	1 0 0	0 0 0	0 0 0	0 0 0 0 1 1	04C003
03	0 0 0 0 0 1 0 0 0	0 0 1	0 0 0	0 0 1	0 0 0 1 0 0	041044
04	0 0 0 0 0 1 0 1 1	1 1 0	1 1 0	1 1 0	0 0 0 0 0 1	05ED81
05	0 0 0 0 0 1 0 1 1	1 1 0	1 1 0	1 1 0	0 1 0 0 0 0	05ED90
06	0 0 0 0 0 1 0 1 1	1 1 0	1 1 0	1 1 0	0 1 0 0 1 1	05ED93
07	0 0 0 0 0 0 0 1 1	0 0 1	0 1 0	1 1 0	0 1 0 1 0 1	019595
08	0 0 0 0 0 1 0 1 1	1 1 0	1 1 0	1 1 0	0 1 0 1 1 1	05ED97
09	0 0 0 0 0 1 0 1 1	1 1 0	1 1 0	1 1 0	0 1 1 0 0 0	05ED98
0A	0 0 0 0 0 1 0 0 1	0 1 0	0 0 0	0 0 0	0 0 0 0 0 1	04A001
0B	0 0 0 0 0 1 0 1 1	0 0 0	0 0 0	0 0 0	0 0 0 0 0 1	058001
0C	0 0 0 0 0 1 0 0 0	0 1 0	0 0 0	0 0 0	0 1 1 1 0 0	04201C
0D	0 0 0 0 0 1 1 0 1	0 0 0	1 0 1	0 0 0	0 1 1 1 0 1	068A1D
0E	0 0 0 0 0 1 1 1 0	0 0 0	1 0 1	0 0 0	0 1 0 0 0 0	070A10
0F	0 0 1 1 0 0 0 1 1	0 0 1	0 1 0	0 0 0	0 0 0 0 0 1	319401
10	0 0 0 0 0 1 0 1 1	1 1 0	1 1 0	1 0 1	0 1 0 0 0 1	05ED51
11	0 0 0 0 0 1 0 1 1	1 1 0	1 1 0	1 1 0	0 1 0 0 1 0	05ED92
12	0 0 0 0 0 1 0 1 1	1 1 0	1 1 0	1 1 0	0 0 0 0 0 1	05ED81
13	0 0 0 0 0 1 0 1 1	1 1 0	1 1 0	1 0 1	0 1 0 1 0 0	05ED54
14	0 0 0 0 0 1 0 1 1	1 1 0	1 1 0	1 1 0	0 0 0 0 0 1	05ED81
15	0 0 0 0 0 1 0 1 1	1 1 0	1 1 0	1 0 1	0 1 0 1 1 0	05ED56

16	0 0 0 0 0 1 0 1 1	1 1 0	1 1 0	1 1 0	0 0 0 0 0 1	05ED81
17	0 0 0 0 0 1 0 0 1	1 1 0	0 0 0	0 0 0	0 0 0 0 0 1	04E001
18	0 0 0 0 0 1 1 1 1	0 1 0	0 0 0	0 0 0	0 1 1 0 0 1	07A019
19	0 0 0 0 0 1 1 1 0	0 0 0	1 0 1	0 0 0	1 0 1 0 1 0	070A2A
1A	0 0 0 0 0 1 0 0 1	1 1 0	0 0 0	0 0 0	0 0 0 0 0 1	04E001
1B	0 0 0 0 0 1 0 0 1	0 1 1	0 0 0	0 0 0	0 0 0 0 0 1	04B001
1C	0 0 0 0 0 1 0 1 1	0 1 0	0 0 1	0 0 0	0 0 0 0 0 1	05A201
1D	1 0 0 1 0 1 0 1 1	0 0 1	1 0 1	0 0 0	0 1 1 1 1 0	959A1E
1E	0 0 0 0 0 1 0 0 1	1 1 0	0 0 0	0 0 0	0 1 1 1 1 1	04E01F
1F	0 0 0 0 0 1 0 0 1	0 1 1	0 0 0	0 0 0	0 0 0 0 0 1	04B001
20	0 0 0 0 0 1 0 1 1	0 1 0	0 0 1	0 0 0	1 0 0 0 0 1	05A221
21	0 1 1 0 0 0 0 1 1	0 0 1	1 0 1	0 0 0	0 0 0 0 0 1	619A01

表 3-5-8 二进制代码

(3) 将二进制表转换为联机操作时的十六进制格式文件（保存文件名为 test）

程序：

\$P00 44

\$P01 46

\$P02 98

\$P03 81

\$P04 F5

\$P05 0C

\$P06 00

微程序：

\$M058110

\$M 05ED82

\$M 04C003

\$M 041044

\$M 05ED81

\$M 05ED90

\$M 05ED93

\$M 019595

\$M 05ED97

\$M 05ED98

\$M 04A001

\$M 058001

\$M 04201C

\$M 068A1D

\$M 070A10

\$M 319401

\$M 05ED51

\$M 05ED92

\$M 05ED81

\$M 05ED54

\$M 05ED81

\$M 05ED56

\$M 05ED81

\$M 04E001

\$M 07A019

\$M 070A2A

\$M 04E001

\$M 04B001

\$M 05A201

\$M 959A1E

\$M 04E01F

\$M 04B001

\$M 05A221

\$M 619A01

第四章系统调试报告

4.1 实验接线

- a、跳线器 J1~J12 全部拨在右边（自动工作方式）；
- b、跳线器 J16、J18、J23、J24 全部拨在左边；
- c、跳线器 J15、J19、J25 全部拨在右边，跳线器 J13、J14 拨在左边；
- d、跳线器 J20~J22、J26、J27 连上短路片；
- e、UJ1 连 UJ2，JSE1 连 JSE2，SJ1 连 SJ2；
- f、MBUS 连 BUS2；
- g、REGBUS 连 BUS5；
- h、PCBUS 连 EXJ2；
- i、ALUBUS 连 EXJ3；
- j、ALUO1 连 BUS1；
- k、EXJ1 连 BUS3；
- l、ALUO2 连 BUS4；
- m、IJ1 连 IJ2。

4.2 联机读/写程序

用 DVCC 联机软件的装载功能将 16 进制格式文件（文件名为 test）装入实验机即可。

4.3 运行程序

① 单步运行程序

A. “编程开关”置“运行”状态，“运行方式”开关置为“单步”状态，“运行控制”开关置为“运行”状态。

B. 拨动总清开关(0→1)，微地址清零，PC 计数器清零，程序首地址为 00H。

C. 按动“启动运行”开关，即单步运行一条微指令。对照微程序流程图，观察微地址显示灯是否和流程一致。

② 连续运行程序

A. “编程开关”置“运行”状态，“运行方式”开关置为“连续”状态，“运行控制”开关置为“运行”状态。

B. 拨动总清开关，清微地址及 PC 计数器，按动“启动运行”开关，系统连续运行程序。如果要停止程序的运行，只需将“运行控制”开关置为“停止”状态，系统就停机。

4.4 解决调试中遇到的问题

①微程序准确录入，校验。

②排线的线路出问题，更换排线后得以解决。

③准确记录程序执行过程。

4.4 通过上机实验，记录指令执行过程如下，这里我们设置输入值为 10。

INPUT(10)→BUS→R0(10)

PC(01)→AR(00)→RAM(44)

RAM(44)→IR(44)→微控制器

PC(02)→AR(01)→RAM(46)

RAM(46)→IR(46)→微控制器

INPUT(10)→BUS→R2(10)

PC(03)→AR(02)→RAM(98)

RAM(98)→IR(98)→微控制器

R2(10)→DR1(10)→ALU(10)

R0(10)→DR2(10)→ALU(20)

ALU(20)→R0(20)

PC(04)→AR(03)→RAM(81)

RAM(81)→IR(81)→微控制器

R0(20)→R1(20)

PC(05)→AR(04)→RAM(F5)

RAM(F5)→IR(F5)→微控制器

R1(20)→299(20)→R1(00)

PC(06)→AR(05)→RAM(0C)

RAM(0C)→IR(0C)→微控制器

PC(07)→AR(06)→RAM(00)

RAM(00)→DR1(00)

RAM(00)→AR(00)→RAM(00)

参考文献

- [1] 白中英. 计算机组成原理. 科学出版社, 2006.8.
- [2] 白中英. 计算机组成原理、题解、题库、实验. 科学技术出版社, 2006.8.
- [3] 王爱英. 计算机组成与结构. 北京:清华大学出版社, 1999.
- [4] 王成. 计算机组成与结构,北京:清华大学出版社, 1999.
- [5] 唐朔飞. 计算机组成原理.高等教育出版社, 1993.
- [6] 胡越明. 计算机组成与系统结构.电子工业出版社,2006.2.
- [7] William Stallings. 计算机组织与结构-性能设计（第四版）(影印版).高等教育出版社,2001.8.

设计总结

通过这次的计算机组成原理的课程设计，我对计算机内部元件的工作原理有了进一步的了解与认识。许多在课堂上没有听懂，没有学会的知识在课程设计的过程中暴露无遗，不过积极的方面是通过课程设计，我把这些知识上的明显漏洞给补上了。我对机器指令及微指令的编码方法不了解，对计算机各部件的组成和工作原理也不是很理解。为了弄清楚这些问题，我又把课本拿出来重新学习相关的章节。虽然理论知识很重要，但是像我们这种工学类的课程，更重要的还是实践。理论联系实际，才能更好的掌握所学的知识。不过在请教了同学和老师之后，这些问题也都一个个迎刃而解。

这次的课程设计题目是基本模型机的设计与实现，在掌握部件单元电路实验的基础上，进一步将其组成系统构造一台基本模型计算机。并为其定义12条机器指令，并编写相应的微程序，具体上机调试掌握整机概念。以前都只是在使用计算机，对它内部的部件组成和构造不是特别的清楚，在课程设计的最初阶段，我没有急于进行具体的设计，而是先理清思路，明确自己要做什么工作是什么，如何来实现基本模型机的相关功能。这些前期的准备工作虽然用去了我很多的时间，但是最后的结果证明，我这样的方法是正确的。课程设计就像是一个系统工程，内容多而复杂，必须先整体上宏观的把握，然后再具体的进行实施。如果开始的方向就不对，那后面的改进就会很麻烦。

温故了这个学期所学的相关知识，心里也更加有信心。后面的相关设计参照课程设计任务书的内容进行了相应的改变，并顺利的在计算机上得到了实现。将实验设备通过串口连接计算机，通过联机软件将机器指令程序和编写的微指令程序存入主存中，并运行此段程序，通过联机软件显示和观察该段程序的运行，验证编写的指令和微指令的执行情况是否符合设计要求。这样的测试方法我还是第一次使用，感觉很新鲜。计算机组成原理更倾向于计算机底层部件的应用，而不像我们平时都在使用的那些界面友好交互性很强的软件，所以有些错误不是很容易发现，只能通过自己耐心的反复调试来解决。这中间也请教了同学和老师，他们对我的帮助都很大。

通过这次课程设计，我的收获很多。课程设计不仅巩固我的理论知识，将理论和实践结合起来，锻炼了我的动手能力，也让我认识到自己的不足之处，争取以后能克服这些缺点，更进一步提高自己各个方面的能力

致谢

在此次课程设计中，感谢老师们的答疑，我对整个模型机系统的设计才得以顺利完成。感谢包老师的热心指导，帮我解决了很多问题。

感谢同学的帮助，微指令二进制表的设计，原先一塌糊涂，在同学的帮助下，我顺利完成了微指令的设计，也明白了微指令各个字段的重要意义。

感谢自己辛苦的设计，结果还是挺满意的。有付出就会有成果。