

\*\*\*\*\*

## 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-6

专业班级：                     

姓    名：                     

学    号：                     

指导教师：   谢鹏寿  

成    绩：

## 前 言

在当今社会，科学技术如此发达，怎样更好的把所学的知识运用到实际操作中，显得越来越重要。课程设计便是一种很好的方式，通过课程设计我们学到很多书本上没有办法学到的东西，把所学的了理论知识运用到现实中，我们收获很大。

本次设计要求我们学生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。

# 目 录

摘    要.....	4
第 1 章    模型机设计概述.....	5
1.1 设计目的.....	5
1.2 设计任务.....	5
1.3 设计原理.....	5
1.4 实验设备与器材.....	6
第 2 章    模型机总体设计.....	7
2.1    模型机的逻辑结构.....	7
2.1.1    运算器.....	7
2.1.2    存储系统.....	8
2.1.3 控制器的设计.....	9
2.1.4 指令系统.....	9
2.1.5 微程序控制器.....	12
2.1.6 输入输出模块.....	14
2.2 模型机的数据通路.....	14
第 3 章    模型机详细设计.....	15
3.1 存储器的原理.....	15
3.1.1 存储器原理图如图 3-1 所示。.....	16
3.1.2 存储器输入输出时序.....	16
3.2 存储容量的扩展.....	16
第 4 章    微程序的设计与实现.....	17
4.1    微程序设计流程.....	17
4.2    微指令格式设计.....	18
4.2.1 设计复杂模型机的监控软件.....	18
4.3    二进制微代码表设计.....	22
第 5 章    系统调试及运行报告.....	24
5.1 实验连线.....	24
5.2 联机读 / 写程序.....	24
5.3 运行程序.....	27
5.4 解决调试中遇到的问题.....	27
5.5 指令执行过程.....	27
设计总结.....	29
参考文献.....	30
致    谢.....	31

## 摘 要

本次课程设计要求了解存储器的工作原理，掌握存储器容量的扩充方法，加深理解计算机系统的基本输入/输出过程和中断处理过程。了解运算器的工作原理，了解控制器的组成和工作原理，强化计算机整机概念，学会用微程序设计方法设计新机器指令所对应的微指令码点。

关键词：存储器    中断处理    工作原理    微程序

# 第1章 模型机设计概述

## 1.1 设计目的

该设计要求学生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 设计任务

以教学实验用模型机为背景，通过调研、分析现有的模型机，建立带有带 8 位自增、自减指令的整机模型。

## 1.3 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 13 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）、AND（逻辑与）、JMP（无条件转移）、BZC（为零或有进位转移）、MOVE（数据传送）、CLR（位清零）、XOR（逻辑异或）、OR（逻辑或）。

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。

## 1.4 实验设备与器材

DVCC 试验箱，74IS181 四位算术逻辑单元/函数发生器，暂存器 74LS273，输出缓冲/显示驱动 74LS245，移位寄存器 74LS299，4 位二进制计数器 74LS161，E<sup>2</sup>PROM 2816 芯片，6264

## 第2章 模型机总体设计

### 2.1 模型机的逻辑结构

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。

#### 2.1.1 运算器

运算器模块主要由运算器U31、U32(74LS181)、暂存器U29、U30(74LS273)、输出缓冲器U33(74LS245)以及进位控制和判零标志控制电路等构成。16位运算器电路是8位电路的一倍。

下面以8位机为例说明运算器模块的工作原理:该模块中算术运算是由2片74LS181(U31、U32)构成,它是运算器的核心。它可以对两个8位的二进制数进行多种算术或逻辑运算,具体由74LS181的功能控制条件M、CN、S3、S2、S1、S0来决定。两个参加运算的数分别来自于暂存器U29和U30,运算结果直接输出到输出缓冲器U33,由输出缓冲器发送到系统的数据总线上,以便进行移位操作或参加下一次运算。

暂存器U29和U30采用8位锁存器74LS273。

输出缓冲器U33采用三态传输器件74LS245,由ALUB'信号来控制,ALUB'为“0”电平时,U33开通,由于U33的方向控制DIR接高电平,因此,U33的数据由A到B,此时其输出B0~B7等于其输入A0~A7;当ALUB'为“1”

电平时，U33不通，其输出呈高阻。

进位控制和判零标志电路如图1—5所示，图中的电路集成在大规模可编程器件中（U50）。299B<sup>'</sup>为移位寄存器U34的允许输出信号，AR为算术运算时是否影响进位及判零标志控制位，低电平有效。ZID是ALU结果为零标志信号，由ALU输出的8位数据输入到U50中，经8输入或非门产生。再看判零电路，ALU在算术运算时，M=0，且移位寄存器不在工作，则299B<sup>'</sup>=1，影响判零电路的控制位AR=0，因此UN3A输出脚3为“1”电平，当时钟脉冲T4正跳时，UN5A的时钟CLK电平产生正跳，此时，ZID状态被存入触发器74LS74（UN5A），触发器的输出QZI就是ALU结果的零标志位。QZI为“0”，表示ALU结果不为零，相应的指示灯ZI灭；QZI为“1”，表示ALU结果为零，相应的指示灯ZI点亮。

进位输入信号来自于两个方面：其一对运算器 74LS181（U31、U32）的进位输出/CN+4 进行倒相所得 CN4（UN4E 的输入 11 脚）；其二由移位寄存器 74LS299（U34）的选择参数 S0、S1、AQ0、AQ7。

### 2.1.2 存储系统

(1) 存储器相关知识：功能:存储信息。 组成(主存储器): 见图 3-2 存储器原理图，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。

(2) CPU 与存储器的连接：地址总线为地址信号，用来指明选中的存储单元地址。数据总线为数据信号，它是微处理器送往存储器的信息或存储器送往微处理器的信息。它包括指令和数据。控制总线发出存储器读写信号，以便从 ROM、RAM 中读出指令或数据，或者向 RAM 写入数据。

(3) 集成 RAM 芯片 SRAM6264 芯片介绍：常用的普通集成 RAM 芯片 SRAM6264 的封装图、电路符号及内部结构分别如图所示。图中  $\overline{CS}_1$ ， $\overline{CS}_2$  是片选信号， $\overline{WE}$  是允许与入信号， $\overline{OE}$  是允许输出(即读出信号)， $A_0 \sim A_{12}$  是地址输入代码； $I/O_0 \sim I/O_7$  是 8 位数据输出， $V_{DD}$  为电源电压，GND 接地，NC 悬空。RAM 的容量用“字数×位数”，6264 的存储容量为“8192 字×8 位”。当存储容量不够时，可以进行字位扩展。

(4) 存储容量的扩展：存储器芯片种类繁多、容量不一样。当一片 RAM(或



ROM)不能满足存储容量位数(或字数)要求时,需要多片存储芯片进行扩展,形成一个容量更大、字数位数更多的存储器。扩展方法根据需要有位扩展、字扩展和字位同时扩展 3 种: a. 位扩展 b. 字扩展 c. 字、位同时扩展。

### 2.1.3 控制器的设计

控制器由程序计数器、指令寄存器、指令译码器、时序产生器和操作控制器组成,它是发布命令的“决策机构”,即完成协调和指挥整个计算机系统的操作。

控制器的主要功能有:从内存中取出一条指令,并指出下一条指令在内存中的位置;对指令进行译码或测试,并产生相应的操作控制信号,以便启动规定的动作。比如一次内存的读/写操作,一个算术逻辑运算操作,或一个输入/输出操作;指挥并控制 cpu、内存和输入/输出,设备之间数据流动的方向。

各部件的功能分别为:

- (1) 程序计数器:程序计数器用来保存下一条指令的地址。
- (2) 指令寄存器:指令寄存器用来保存当前正在执行的一条指令。
- (3) 指令译码器:对指令进行译码或测试,并产生相应的操作控制信号。
- (4) 时序产生器:时序产生器的作用,就是对各种操作实施时间上的控制。
- (5) 操作控制器: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。

### 2.1.4 指令系统

模型机设计指令共 13 条,其中包括算术逻辑指令、I/O 指令、访问和转移指令。

#### (1) 算术逻辑指令

设计 5 条算术逻辑指令并用单字节表示,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
OP-CODE				RS		RD	

其中，OP-CODE 为操作码，RS 为源寄存器，RD 为目的寄存器，并规定：

RS 或 RD	选定的寄存器
00	R0
01	R1
10	R2

6 条算术逻辑指令的名称、功能和具体格式如表 2-1 所示。

## (2) 访问指令及转移指令

模型机设计 2 条访问指令，即存数（STA）、取数（LDA），2 条转移指令，即无条件转移（JMP）、结果为零或有进位转移指令（BZC），指令格式为：

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

其中，OP-CODE 为操作码，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，其定义如下：

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

本模型机规定变址寄存器 RI 指定为寄存器 R2。

## (3) I/O 指令

输入（IN）和输出（OUT）指令采用单字节指令，其格式如下：

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

其中，addr=01 时，选中“INPUT DEVICE”中的开关组作为输入设

备，addr=10 时，选中“OUTPUT DEVICE”中的数码块作为输出设备。

表 2-1 给出了各条指令的格式、汇编符号、指令功能。

表 2-1 指令格式

助记符号	指令格式	功能
<div>CLR rd</div> <div>MOV rs, rd</div> <div>ADD rs, rd</div> <div>SUB rs, rd</div> <div>AND rs, rd</div> <div>OR rs, rd</div> <div>XOR rs, rd</div>	<div><div><div>0111</div><div>0</div><div>rd</div></div></div> <div><div><div>1000</div><div>rs</div><div>rd</div></div></div> <div><div><div>1001</div><div>rs</div><div>rd</div></div></div> <div><div><div>1010</div><div>rs</div><div>rd</div></div></div> <div><div><div>1100</div><div>rs</div><div>rd</div></div></div> <div><div><div>1101</div><div>rs</div><div>rd</div></div></div> <div><div><div>1110</div><div>rs</div><div>rd</div></div></div>	<div><math>0 \rightarrow rd</math></div> <div><math>rs \rightarrow rd</math></div> <div><math>rs + rd \rightarrow rd</math></div> <div><math>rs - rd \rightarrow rd</math></div> <div><math>rs \wedge rd \rightarrow rd</math></div> <div><math>rs \vee rd \rightarrow rd</math></div> <div><math>rs \oplus rd \rightarrow rd</math></div>
<div>LDAM,D, rd</div> <div>STA M, D, rd</div> <div>JMP M, D</div> <div>BZC M, D</div>	<div><div><div><div>00</div><div>M</div><div>00</div><div>rd</div></div><div>D</div></div><div><div><div><div>00</div><div>M</div><div>01</div><div>rd</div></div><div>D</div></div></div><div><div><div><div>00</div><div>M</div><div>00</div><div>rd</div></div><div>D</div></div></div><div><div><div><div>00</div><div>M</div><div>00</div><div>rd</div></div><div>D</div></div></div></div>	<div><math>E \rightarrow rs</math></div> <div><math>rd \rightarrow E</math></div> <div><math>E \rightarrow PC</math></div> <div>CY=1 或 Z=1, <math>E \rightarrow PC</math></div>
<div>IN addr, rd</div> <div>OUT addr, rd</div>	<div><div><div><div>010</div><div>01</div><div>rd</div></div></div></div>	<div><math>addr \rightarrow rd</math></div> <div><math>rd \rightarrow addr</math></div>

		<b>010</b> <b>1</b>	<b>01</b>	<b>r</b> <b>d</b>		
--	--	------------------------	-----------	----------------------	--	--

/

表 2-2 微指令格式

24	23	22	21	20	19	18	17	16	15	12	9	6	5	4	3	2	1
									14 13	11 10	8 7						
S3	S2	S1	S0	M	CN	WE	A9	A8	A	B	C	uA 5	uA4	uA3	uA2	uA1	uA0

A 字段

15	14	13	选择
0	0	0	
0	0	1	LDRi
0	1	0	LDDR 1
0	1	1	LDDR 2
1	0	0	LDIR
1	0	1	LOAD
1	1	0	LDAR

B 字段

12	11	10	选择
0	0	0	
0	0	1	RS-B
0	1	0	RD-B
0	1	1	RI-B
1	0	0	299-B
1	0	1	ALU-B
1	1	0	PC-B

C 字段

9	8	7	选择
0	0	0	
0	0	1	P(1)
0	1	0	P(2)
0	1	1	P(3)
1	0	0	P(4)
1	0	1	AR
1	1	0	LDPC

### 2.1.5 微程序控制器

微控制器逻辑结构如图 2-3 所示。

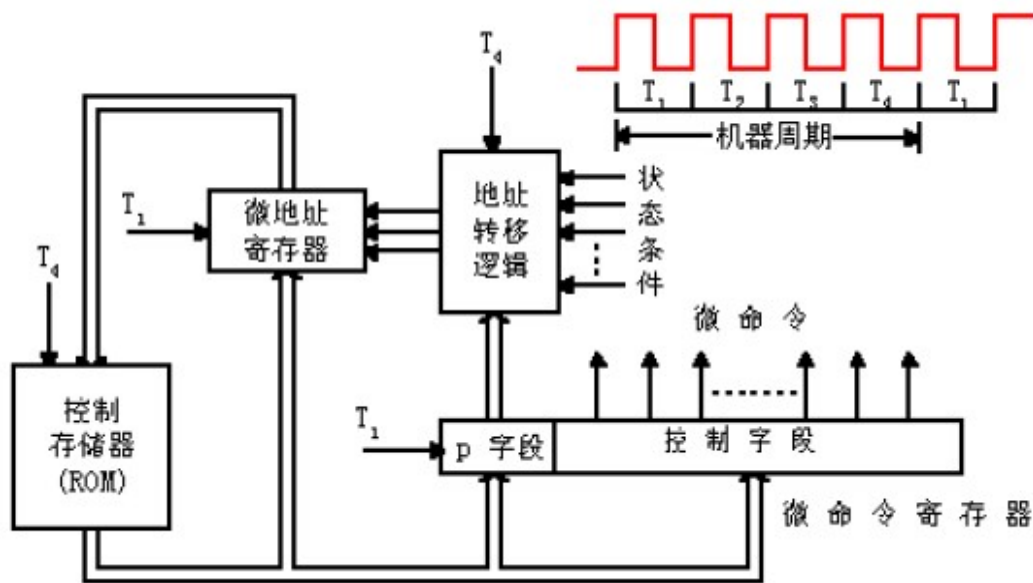


图 2-3 微控制器逻辑结构图

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。

时序信号图 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，画出其波形如下图 2—4 所示，并标注测量所得的脉冲宽度。

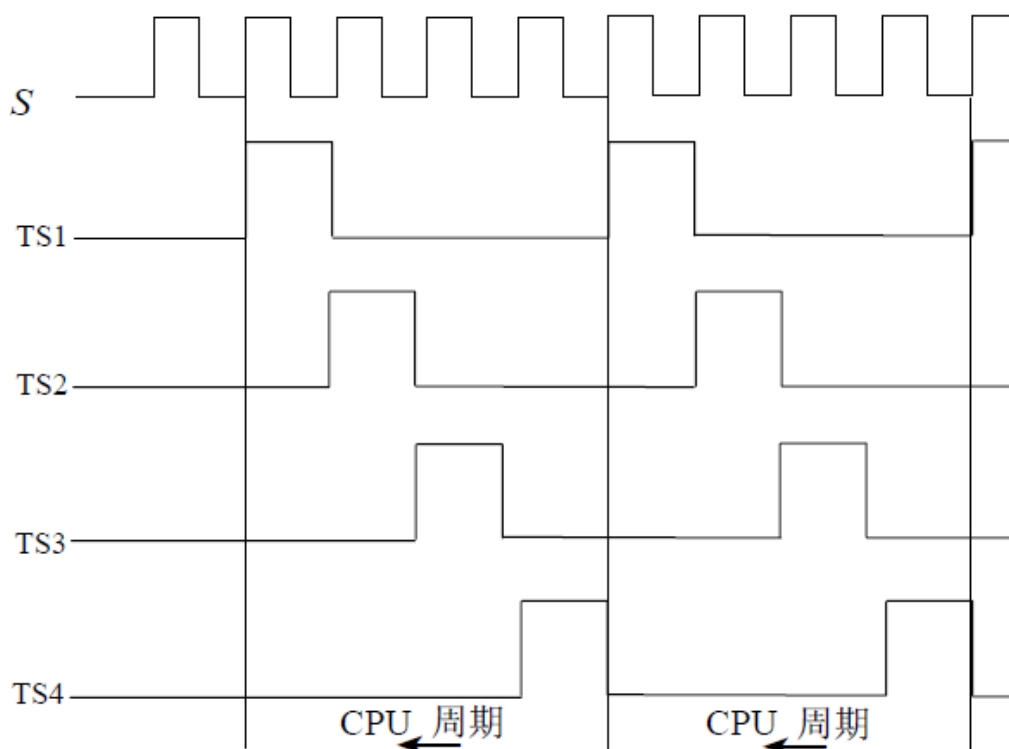


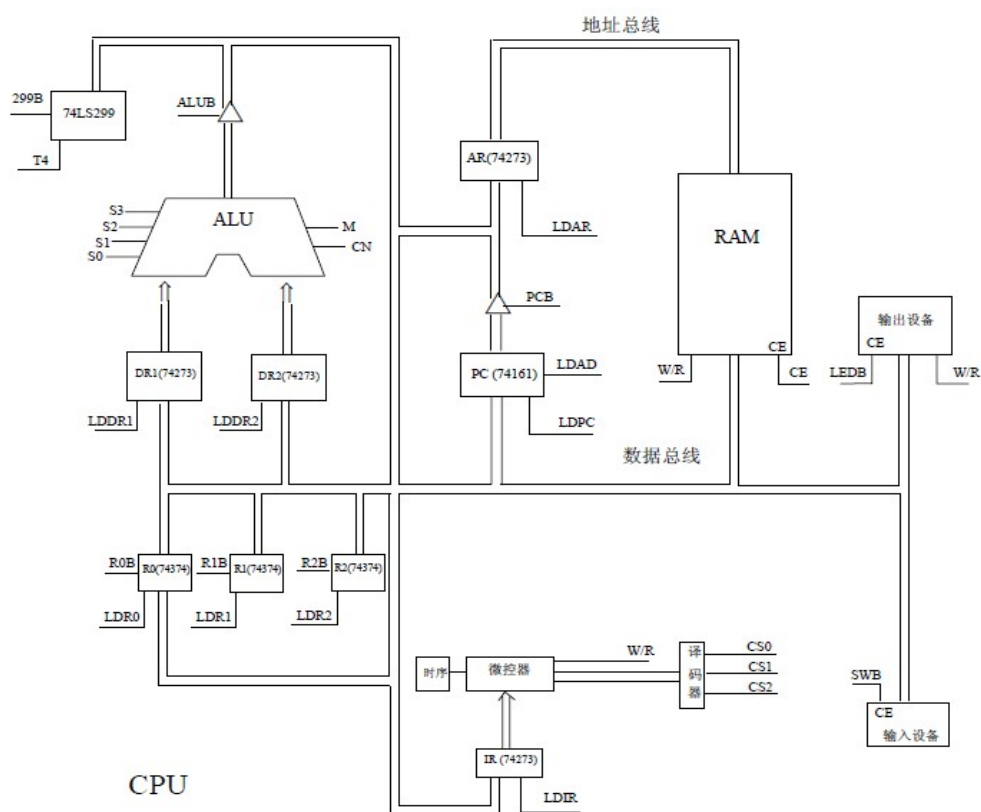
图 2-4 时序信号图

### 2.1.6 输入输出模块

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。

## 2.2 模型机的数据通路

复杂模型机的数据通路图如图 2-5 所示。



## 第3章 模型机详细设计

### 3.1 存储器的原理

## 存储器相关知识

功能:存储信息。

组成(主存储器): 见图 3-1 存储器原理图

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。

3.1.1 存储器原理图如图 3-1 所示。

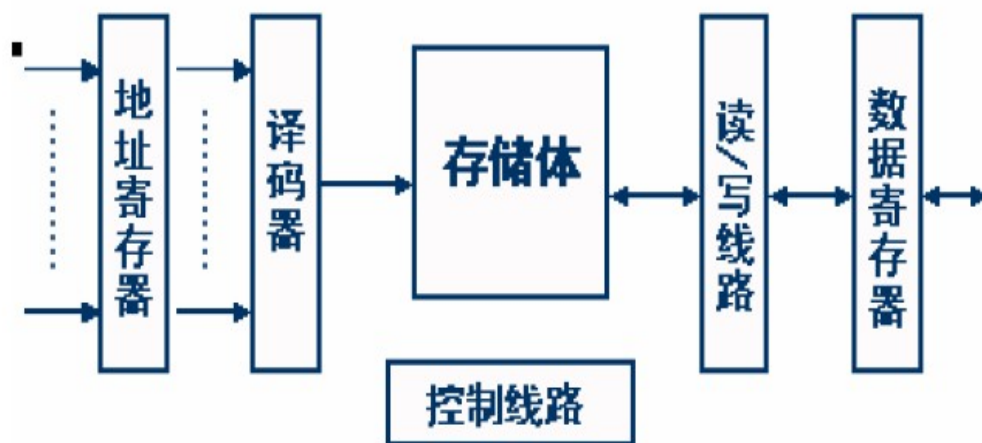


图 3-1 存储器原理图

3.1.2 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。

## 3.2 存储容量的扩展

存储器芯片种类繁多、容量不一样。当一片 RAM(或 ROM)不能满足存储容量位数(或字数)要求时，需要多片存储芯片进行扩展，形成一个容量更大、字数位数更多的存储器。扩展方法根据需要有位扩展、字扩展和字位同时扩展 3 种。



### (1) 位扩展

若一个存储器的字数用一片集成芯片已经够用,而位数不够用,则用“位扩展”方式将多片该型号集成芯片连接成满足要求的存储器。扩展的方法是将多片同型号的存储器芯片的地址线、读/写控制线( $R/\overline{W}$ )和片选信号 $\overline{CS}$ 相应连在一起,而将其数据线分别引出接到存储器的数据总线上。

### (2) 字扩展

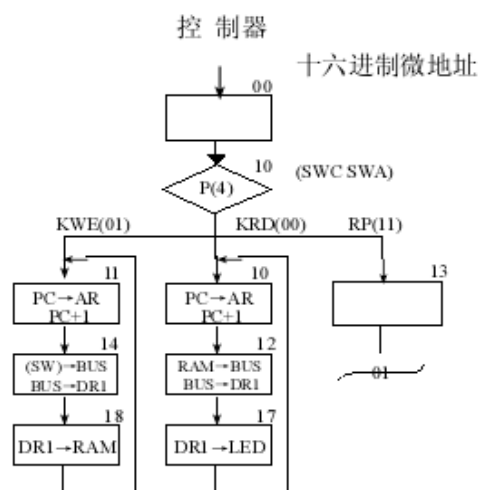
若每一片存储器的数据位数够而字线数不够时,则需要采用“字线扩展”的方式将多片该种集成芯片连接成满足要求的存储器。扩展的方法是将各个芯片的数据线、地址线和读写( $R/\overline{W}$ )控制线分别接在一起,而将片选信号线( $\overline{CS}$ )单独连接。

### (3) 字、位同时扩展

在很多情况下,要组成的存储器比现有的存储芯片的字数、位数都多,需要字位同时进行扩展。扩展时可以先计算出所需芯片的总数及片内地址线、数据线的条数,再用前面介绍的方法进行扩展,先进行位扩展,再进行字扩展。

## 微程序

### 4.1 微



## 第 4 章 的设计与实现

### 程序设计流程



\$P00 44	IN 01 R0
\$P00 46	IN 01 R1
\$P02 98	ADD R0,R1
\$P03 81	MOV R0,R1
\$P04 F5	CLR R1
\$P05 0C	BZC 00,00
\$P06 00	

#### 4. 2. 2 按照表 4-2 系统建议的微指令格式

参照微指令流程图，设计微指令，并译成二进制代码表。

表 4-2 微指令格式：

S3	S2	S1	S0	M	CN	WE	B1	B0	A	B	C	uA5	uA4	uA3	uA2	uA1	uA0

S3, S2, S1, S0, M, Cn:

为运算器74LS181芯片的控制信号。

WE : 为W/R信号对RAM和OUT进行写操作，高电平为写有效。

B1, B0 : 为对外部设备（RAM, OUTPUT, INPUT）地址进行译码，  
B1B0=00时，INPUT（即SWB）选中；B1B0=01时，RAM（即CE）选中；B1B0=10  
时，OUTPUT（即LEDB）选中，B1B0=11时，外部设备都不选中。

表 4-3 A 字段

15	14	13	选择
0	0	0	
0	0	1	LDRi
0	1	0	LDDR1

0	1	1	LDDR2
1	0	0	LDIR
1	0	1	LOAD
1	1	0	LDAR

**LDIRi:** 寄存器输入选中，具体选择同指令寄存器（IR）的最低2位（I1，I0）配合，当I1，I0=00时为输入到R0寄存器；I1，I0=01时为R1；I1，I0=10时为R2。

**LDDR1:** 暂存器DR1选中。

**LDDR2:** 暂存器DR2选中。

**LDIR:** 指令寄存器IR选中。

**LOAD:** 总线数据直接装载到PC计数器。

**LDAR:** 地址寄存器 AR 选中。

表 4-4 B 字段

12	11	10	选择
0	0	0	
0	0	1	RS-B
0	1	0	RD-B
0	1	1	RI-B
1	0	0	299-B
1	0	1	ALU
1	1	0	PC-B

**RS-B:** 为源寄存器输出选中。具体选择同指令寄存器（IR）的3，4位（I3，I2）配合，当I3，I2=00时为输入到R0寄存器；I3，I2=01时为R1；I3，I2=10时为R2。

**RD-B:** 为目的寄存器输出选中。具体选择同指令寄存器（IR）的最低2位（I1，I0）配合，当I1，I0=00时为输入到R0寄存器；I1，I0=01时为R1；I1，I0=10时为R2。

**RI-B:** 为变址寄存器选中。本机定固定为R2。

**299-B:** 移位寄存器输出选中。

**ALU-B:** 逻辑运算单元结果输出。

**PC-B :** PC计数器输出。

表 4-5 C 字段

9	8	7	选择
0	0	0	
0	0	1	P(1)
0	1	0	P(2)
0	1	1	P(3)
1	0	0	P(4)
1	0	1	AR
1	1	0	LDA R

其中 **uA5~uA0** 为 6 位的后续微地址，**A、B、C** 为三个译码字段，分别由三个控制位译码出多位。**C** 字段中的 **P1~P4** 是四个测试字位，其功能是根据机器指令及相应微代码进行译码，使微程序转入相应的为地址入口，从而实现微程序的顺序、分支、循环运行。具体来说，**P1** 测试用于“取指令”微指令，它用下址低四位 (**uA3~uA0**) 与指令寄存器高四位 (**IR7~IR4**) 相或得到各路分支；**P2** 测试用下址低 2 位 (**uA1~uA0**) 与指令寄存器的 **IR3、IR2** 相或得到各路分支；**P3** 测试用于条件转移，它用下址的 **uA4** 与(**ZI +CY**) 相或得到各路分支；**P4** 测试用于控制台操作，它用下址低 2 位 (**uA1~uA0**) 与 **SWB、SWA** 相或得到各路分支。

在上述各测试下址中未用到的位均直接保留。**AR** 为算术运算是否影响进位及判零标志控制位，其为零有效。**B** 字段中的 **RS-B、RD-B、RI-B** 分别为源寄存器选通信号，目的寄存器选通信号及变址寄存器选通信号，其功能是根据机器指令来进行三个工作寄存器 **R0、R1** 及 **R2** 的选通译码。三字段中的其他位类似与此，均是某芯片的选通信号，它们的功能都是根据机器指令来进行相应芯片的选通译码。

### 4.3 二进制微代码表设计

表 4-6 二进制代码表

微地址	S3~CN	RD	M17M16	A	B	C	uA5~uA0
00	000000	0	11	000	000	100	010000
01	000000	0	11	110	110	110	000010
02	000000	0	01	100	000	001	100000
35	000000	0	01	010	000	000	011110
36	000000	0	01	110	000	010	000100
37	000000	0	01	110	000	010	110000
60	000000	0	01	010	000	000	110001
61	000000	0	01	110	000	010	000100
54	000000	0	11	110	110	110	011101
55	000000	0	11	110	110	110	011111
56	000000	0	01	010	000	000	110010
23	000000	0	11	000	000	000	000001
24	000000	0	00	010	000	000	011000
62	000000	0	01	010	000	000	110011
63	000000	0	11	011	011	000	110100
64	100101	0	11	110	101	000	111100
20	000000	0	11	110	110	110	000011
21	000000	0	11	110	110	110	000101
22	000000	0	11	110	110	110	001101
57	000000	0	11	110	110	110	110101
51	000000	0	00	001	000	000	000001
52	000000	1	10	000	010	000	000001
46	000000	0	11	000	000	000	100110
40	001111	0	11	001	101	000	000001
41	000000	0	11	001	001	000	000001
42	000000	0	11	010	001	000	111010
43	000000	0	11	011	001	000	000011
44	000000	0	11	010	010	000	001101
45	000000	0	11	010	001	000	001110
53	000000	0	11	010	010	000	111011
47	001100	0	11	000	001	000	011010
50	001100	0	11	000	001	000	011011
05	000000	0	01	001	000	000	000001
04	000000	1	01	000	010	001	000001
07	000001	0	11	101	101	110	000001
06	000000	0	11	000	000	011	001001

11	000000	0	11	000	000	000	000001
74	100101	0	11	010	101	010	000100
65	000000	0	01	010	000	000	110111
67	000000	0	11	011	110	000	111000
70	100101	0	11	110	101	000	111001
71	100101	0	11	010	101	010	000100
72	000000	0	11	011	010	000	001010
12	100101	0	11	001	101	101	000001
03	000000	0	11	010	010	000	001100
14	011001	0	11	010	101	101	010101
25	000011	0	11	001	101	000	000001
57	000000	0	11	010	101	000	110000
60	000011	0	11	000	000	101	110001
61	100101	0	11	001	101	101	000001
15	000000	0	11	001	101	000	000001
16	000000	0	11	011	010	000	001111
31	000001	0	11	101	101	110	000001
17	101110	0	11	001	101	101	000001
73	000011	0	11	001	101	000	000001
32	001010	0	11	000	100	000	001011
13	000000	0	11	001	100	000	000001
33	000110	0	11	000	100	000	001011
72	000000	0	11	001	100	000	000001
27	000001	1	10	000	101	000	010000
30	000001	1	01	000	101	000	010001

## 第 5 章 系统调试及运行报告

### 5.1 实验连线

- 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。

### 5.2 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。

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：



**\$ M00 088105**  
**\$ M01 82ED05**  
**\$ M02 50C004**  
**\$ M03 04A004**  
**\$ M04 A0E004**  
**\$ M05 06E004**  
**\$ M06 07A004**  
**\$ M07 A0E004**  
**\$ M08 8AED05**  
**\$ M09 8CED05**  
**\$ M0A 3BA004**  
**\$ M0B 018005**  
**\$ M0C 3C2004**  
**\$ M0D 0EA004**  
**\$ M0E 0FB605**  
**\$ M0F 25EA95**  
**\$ M10 83ED05**  
**\$ M11 85ED05**  
**\$ M12 8DED05**  
**\$ M13 A6ED05**  
**\$ M14 011004**  
**\$ M15 010407**  
**\$ M16 168005**  
**\$ M17 019A3D**  
**\$ M18 019205**  
**\$ M19 2AA205**  
**\$ M1A 2CB205**  
**\$ M1B 32A205**  
**\$ M1C 33A205**  
**\$ M1D 36A205**  
**\$ M1E 378235**

**\$ M1F 398235**  
**\$ M20 019004**  
**\$ M21 018406**  
**\$ M22 81DB05**  
**\$ M23 E48005**  
**\$ M24 018005**  
**\$ M25 A0AA95**  
**\$ M26 27A004**  
**\$ M27 28BC05**  
**\$ M28 29EA95**  
**\$ M29 A0AA95**  
**\$ M2A 2BB405**  
**\$ M2B 419B95**  
**\$ M2C 2DA405**  
**\$ M2D 6EAB05**  
**\$ M2E 2FAA0D**  
**\$ M2F 30AA05**  
**\$ M30 71810D**  
**\$ M31 419B95**  
**\$ M32 019A05**  
**\$ M33 35B405**  
**\$ M34 81DB05**  
**\$ M35 419BBD**  
**\$ M36 019A0D**  
**\$ M37 38882D**  
**\$ M38 019805**  
**\$ M39 3A881D**  
**\$ M3A 019805**  
**\$ M3B 080A07**  
**\$ M3C 098A06**

### 5.3 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。

### 5.4 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。

### 5.5 指令执行过程

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)**

## 设计总结

在这次课程设计，使我加深了对计算机组成原理知识的理解、并了解了计算机部件的设计、部件间的连接、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，加深了对组成原理理论课程的理解。

通过一系列微程序的编写，使我对程序执行的流程以及指令之间的跳转有了更深入的理解，能够根据流程图和机器指令写出相应微程序，对模型机里的数据流向图也有了一定的理解，能看懂根据自己编写的微程序在模型机上的一步步流向，对 **IN**、**OUT**、**ADD**、**AND**、**BZC**、**CLR** 等指令的运用更加熟悉，对它们微程序的一步步流程也有了清晰的认识。通过对机器指令和微指令的分析和编写，理解了指令代码所对应的功能。本次课程设计里的微程序均由自己根据流程图及机器指令编写，基本完成了题目的要求。

两周的课程设计，虽然在设计过程中，因为个人的水平有限，所以设计出来的模型难免出现或多或少的问题，希望老师能给予指正，我们也会更加努力的进行改正。

## 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.

## 致 谢

首先要感谢我的父母，这么多年来不求回报的养育着我，给了我这么多与众不同，永不放弃的为梦想而执着奋斗至今。谢谢父母！

感谢指导老师，他在我的课程设计过程中提出了指导性的方案和架构，在我设计思想出现偏差的时候给予我必要的提醒和指示，让我能够从中领悟更多，并及时的修改设计方案，完善作品。非常感谢！

还要感谢教授其他课程的老师，引领我走进美妙的设计开发世界，从而使我有目的性的涉猎关于程序设计和开发方面的相关书籍，不断的发现问题，和自我解决问题。

当然要感谢 linux 实验室的老师和朋友，给予我这么好的开发环境，他们负责认真的工作态度和解决问题的思想无时无刻不影响着我，在我迷茫的时候伸出双手帮助我。这些对我是如此的重要。

最后，要感谢和我并肩开发设计的老伙计，在工作过程中，他们替我分担了很多文档方面的重任，使我们的作品能够得到更好的表达，愉快和谐的合作使我们做起事情来事半功倍。

感谢所有的人！