
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： 模型机的设计-2

专业班级：

姓 名：

学 号：

指导教师： 谢 鹏 寿

成 绩：

前 言

计算机组成原理课程是计算机系很重要的一门专业基础课，从这门课的内容特点看，它属于工程性、技术性和实践性都很强的一门课，因此，在进行课堂教学的同时，必须对实验教学环节给予足够的重视，要有良好的实验环境，能进行反应主要教学内容的、水平确实比较高的实验项目，在深化计算机各功能部件实验的同时，加强对计算机整机硬件系统组成与运行原理有关内容的实验；在教学实验的整个过程中，坚持以硬件知识为主的同时，加深对计算机整机系统中软硬件的联系与配合的认识。目前，有些单位和院校都研制出一些用于计算机组成原理课程教学实验的系统或装置，也各具特色但基本上都是相对孤立的功能部件的实验，整机硬件方面的实验很难胜任，更不能对计算机系统中硬软件的联系和配合的学习提供足够的帮助。而 DVCC 系列实验计算机系统就是专为计算机组成原理课的授课和教学实验而研制的。

DVCC 系列计算机组成原理实验系统作为较高层次、专用于计算机原理课程教学实验的实验计算机系统具有良好的实验性能和系统的完整性以及可扩展性。

良好的实验性体现在 DVCC 系列机能很好地完成计算机硬件系统各功能部件的教学实验，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；计算机的 CPU 自行设计与实现，配有小的监控程序，有自己的汇编语言的支持。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。

系统的完整性体现在 DVCC 系列机与学生常见到的简单计算机大体相同，其主要组成与运行方式和 PC 机差不多，该系列机是一台硬软件相对完整、配置巧妙合理的完整的计算机系统，通过它能体现出重要教学内容、能完成主要教学实验项目。

目 录

摘 要.....	iii
第 1 章 模型机设计概述.....	1
1.1 设计目的	1
1.2 设计任务	1
1.3 设计原理	1
1.4 实验设备与器材	3
第 2 章 模型机总体设计.....	4
2.1 模型机的逻辑结构	4
2.1.1 运算器	4
2.1.2 存储系统	4
2.1.3 指令系统	4
2.1.4 微程序控制器	6
2.1.5 输入输出模块	6
2.2 模型机的数据通路	6
第 3 章 详细设计	8
3.1 存储器系统的组成与说明	8
3.2 存储器原理图	9
3.3 存储器输入输出时序	9
第 4 章 微程序的设计与实现.....	10
4.1 微程序设计流程	10
4.2 微指令格式设计	10
4.3 二进制微代码表设计	11
第 5 章 系统调试及运行报告.....	13
5.1 微程序的设计	13
5.1.1 微程序	13
5.1.2 监控程序	14
5.2 调试结果	14
5.3 调试时的问题及解决	17
设计总结	18
参考文献	19
致 谢	20

摘 要

计算机的指令系统是由一系列的机器指令组成的。指令就是计算机执行某种操作的命令。从计算机组成的层次结构来说，计算机的指令有微指令、机器指令和宏指令。微指令是微程序级的命令，属于硬件；宏指令是由若干条机器指令组成的软件指令，属于软件；机器指令是介于微指令与宏指令之间，通常称为指令，每一条机器指令可完成一个独立的算术运算或逻辑运算操作。

本系统采用 TDN-CM+计算机组成原理教学实验系统搭建电路图，从而在实验箱上实现一指令系统。该系统在基本模型机的基础上改进并实现 IN,STA,OUT,加 ADD，跳转 JMP，SUB 与等 6 条指令的功能。

关键词：计算机组成原理；微指令；机器指令；基本模型机

第1章 模型机设计概述

1.1 设计目的

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

- 1、根据任务要求设计整机系统的方案。
- 2、存储系统：使用模型机的存储模块，说明存储器的输入输出时序，模块连接方式等。
- 3、运算器：使用模型机的器件，组成带有片间串行进位 16 位算数逻辑运算功能的运算器。
- 4、微程序控制器模块：使用教学机的系统，设计微程序控制器。
- 5、设计模型机指令系统：（含设计微指令格式、微程序流程图，每条指令所对应的微程序等）。指令系统包括下列指令：IN、OUT、STA、LDA、JMP、BZC、CLR、MOV、ADD、SUB、ADC、ADT、INC、DEC、SBT、SBC
- 6、了解并说明教学模型机的输入输出模块。
- 7、在自己设计的指令系统基础上，编制一个汇编语言小程序并进行调试通过。其中运
- 8、整机设计分模块进行，说明模块中数据和控制信号的来源、去向、功能、时序，以及模块间数据和控制信号的来源、去向、功能、时序等。

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，如运算器实验中对 74LS—181 芯片的控制，存储器中对存储器芯片的控制信号，以及几个实验中对输入设备的控制。这里，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一段微程序。

本系统使用两种外部设备，一种是二进制代码开关(DATA UNIT)，它作为输入设备；另一种是发光二极管(BUS UNIT 上的一组发光二极管)，它作为输出设备。例如：输入时，

二进制开关数据直接经过三态门送到总线上，只要开关状态不变，输入的信息也不变。输出时，将输出数据送到数据总线 BUS 上，驱动发光二极管显示。

本次设计采用六条机器指令；IN（输入）、ADD（二进制加法）、STA（存数）、OUT（输出）、JMP（无条件转移），AND（与运算），得指令格式表（前 4 位为操作码），如表 1.1 所示。

表 1.1 指令格式

助记符	机器指令码	说明
IN	0000 0000	“DATA UNIT”中的开关状态 →R0
ADD addr	0001 0000 xxxx xxxx	R0+[addr]→R0
STA addr	0010 0000 xxxx xxxx	R0→[addr]
OUT addr	0011 0000 xxxx xxxx	[addr]→BUS
JMP addr	0100 0000 xxxx xxxx	Addr→PC
SUB [addr]	0101 0000 xxxx xxxx	R0 SUB [addr]→R0

其中 IN 为单字长（8 位），其余为双字长指令，xxxxxxx 为 addr 对应的二进制地址码。

微控器读取一条机器指令后，将通过如下的逻辑电路，对 SE1~SE5 中的某一位或者几位激活，从而实现机器指令与微程序的对应。当然，该逻辑电路还能接收外部控制输入 SWA、SWB，内部状态输出 FC、FZ 等信号，并对这些信号给出相应的输出。

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“0 0”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“0 1”时，按 START 微动开关，可对 RAM 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 为“1 1”时，按 START 微动开关，即可转入到第 01 号“取指”微指令，启动程序运行。

上述三条控制台指令用两个开关 SWB、SWA 的状态来设置，得 SWB、SWA 定义表，如表 1.2 所示：

表 1.2 SWB、SWA 定义表

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）

1.4 实验设备与器材

DVCC 试验箱；74LS181 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 74LS161，74LS138 译码器，E2PROM 2816 芯片，6264 芯片；PC 机一台；排线若干

第2章 模型机总体设计

2.1 模型机的逻辑结构

2.1.1 运算器

设计中所用 8 位运算器电路。其中运算器由四片 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~D6 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, 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, 并经过一个三态门 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。

2.1.2 存储系统

主存储器单元电路主要用于存放实验中的机器指令

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。

存储器芯片种类繁多、容量不一样。当一片 RAM 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。

2.1.3 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 8 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

(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)，1 条转移指令，即无条件转移 (JMP)，指令格式为：

7	6	5	4	3	2	1	0
0		M		OP-COD		RD	
0				E			
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				ADDR		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。

2.1.4 微程序控制器

微控制器主要由控制器，微指令寄存器和地址转移逻辑三大部分组成。其中，微指令寄存器分为微地址寄存器和微命令寄存器两部分。

控制存储器用来存放实现全部指令系统的微程序，它是一种只读型存储器。一旦微程序固化，机器运行时则只读不写。在串行方式的微程序控制器中，微指令周期就是只读存储器的工作周期。控制存储器的字长就是微指令字的长度，其存储容量视机器指令系统而定，即取决于微程序的数量。对控制存储器的要求是速度快，读出周期要短。

微指令寄存器用来存放由控制存储器读出的一条微指令信息。其中微地址寄存器决定将要访问的下一条微指令地址，而微命令寄存器则保存一条微指令的操作控制字段和判别测试字段的信息。

地址转移逻辑在一般情况下，微指令由控制存储器读出后直接给出下一条微指令的地址，通常我们简称为微地址，这个微地址信息就存放在微地址寄存器中。如果微程序不出现分支，那么下一条微指令的地址就直接由微地址寄存器给出，当微程序出现分支时，意味着微程序出现条件转移。在这种情况下，通过判别测试字段 P 和执行部件的“状态条件”反馈信息，去修改微地址寄存器的内容，并按改好的内容去读下一条微指令。地址转移逻辑就承担自动完成修改微地址的任务。

2.1.5 输入输出模块

输入设备单元

系统中用8个拨动开关作为输入设备，通过总线驱动器 74LS245 (U51) 输出到系统的扩展数据总线 EXD0~EXD7 上，输入的数据显示在 LD0~LD7 八个 LED 上，高电平亮，低电平灭。

输出设备单元

此单元设置两个七段数码管，用于显示需要输出的数据。七段数码管的译码电路由两片 GAL16V8 (U53、U54) 组成。

2.2 模型机的数据通路

本次课设所能实现的功能 将由微程序控制自动产生各部件单元控制信号，实现特定指令的功能。这里，实验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一个微程序。

总体设计数据通路框图如图 2.1 所示：

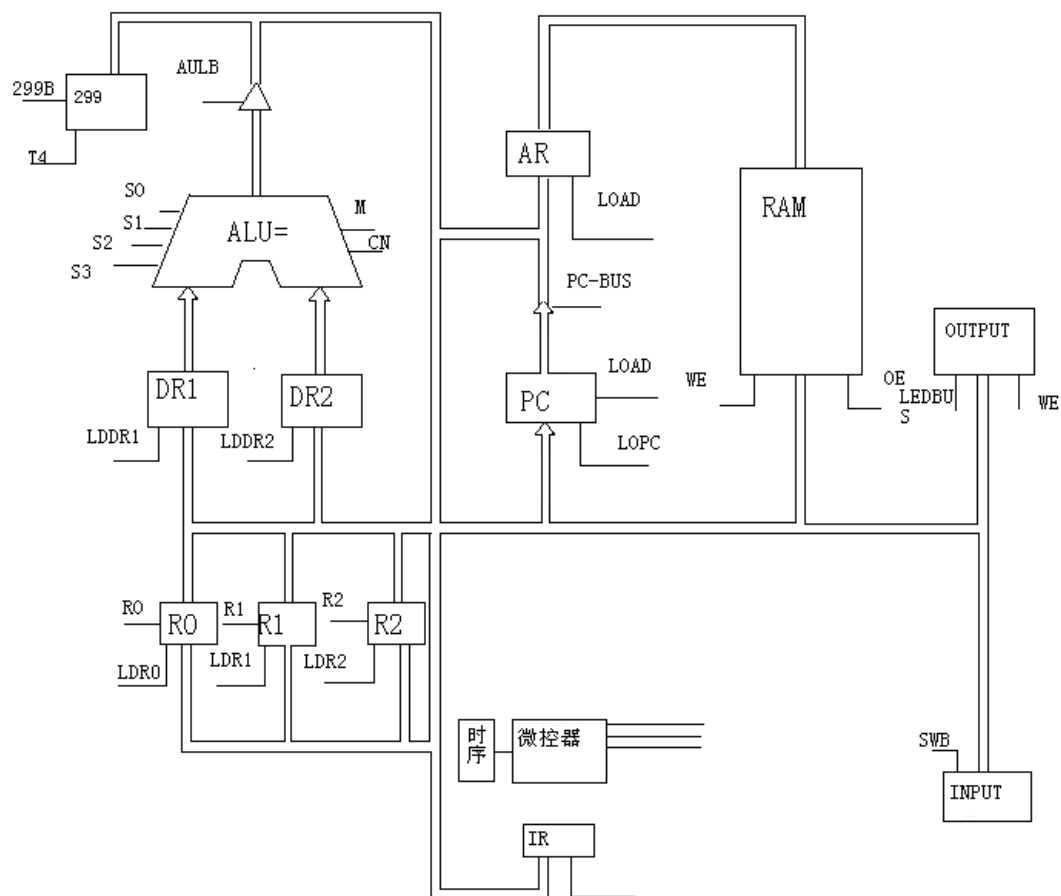


图2.1 数据通路框图

第3章 详细设计

3.1 存储器系统的组成与说明

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，功能表如下：

表2.1 存储器功能表

工作方式	I/O		输入		
	DI	DO	/OE	/WE	/CS
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	L	L

主存储器单元电路主要用于存放实验中的机器指令，它的数据总线挂在扩展数据总线 EXD0~EXD7 上；它的地址总线由地址寄存器单元电路中的 地址寄存器 74LS273 (U37) 给出，地址值由 8 个 LED 灯 LAD0~LAD7 显示，高电平亮，低电平灭；它的读信号直接接地；它的写信号和片选信号由写入方式确定。该存储器中机器指令的读写分手动和自动两种方式。手动方式下，写信号由 W/R 提供，片选信号由 CE' 提供；自动方式下，写信号由控制 CPU 的 P1.2 提供，片选信号由控制 CPU 的 P1.1 提供。

组成(主存储器)：见图 3.1 存储器原理图。

存储器芯片种类繁多、容量不一样。当一片 RAM 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。

3.2 存储器原理图

存储器原理图如图 3.1 所示:

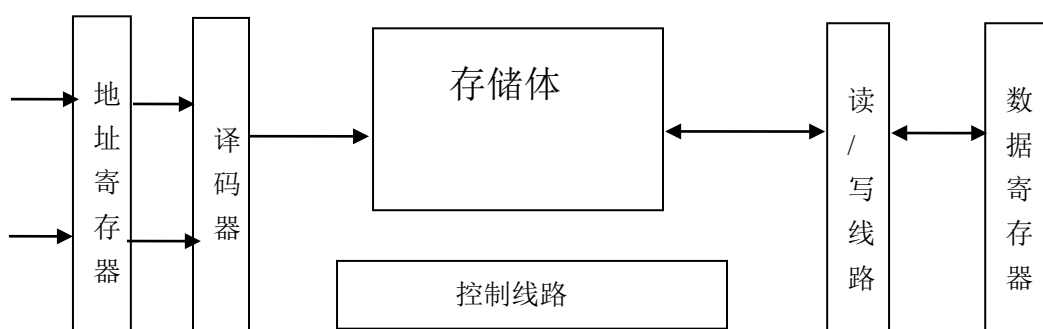


图 3.1 存储器原理图

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。

3.3 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。

第4章 微程序的设计与实现

4.1 微程序设计流程

根据机器指令的要求和数据通路图，设计各指令对应的微程序流程图

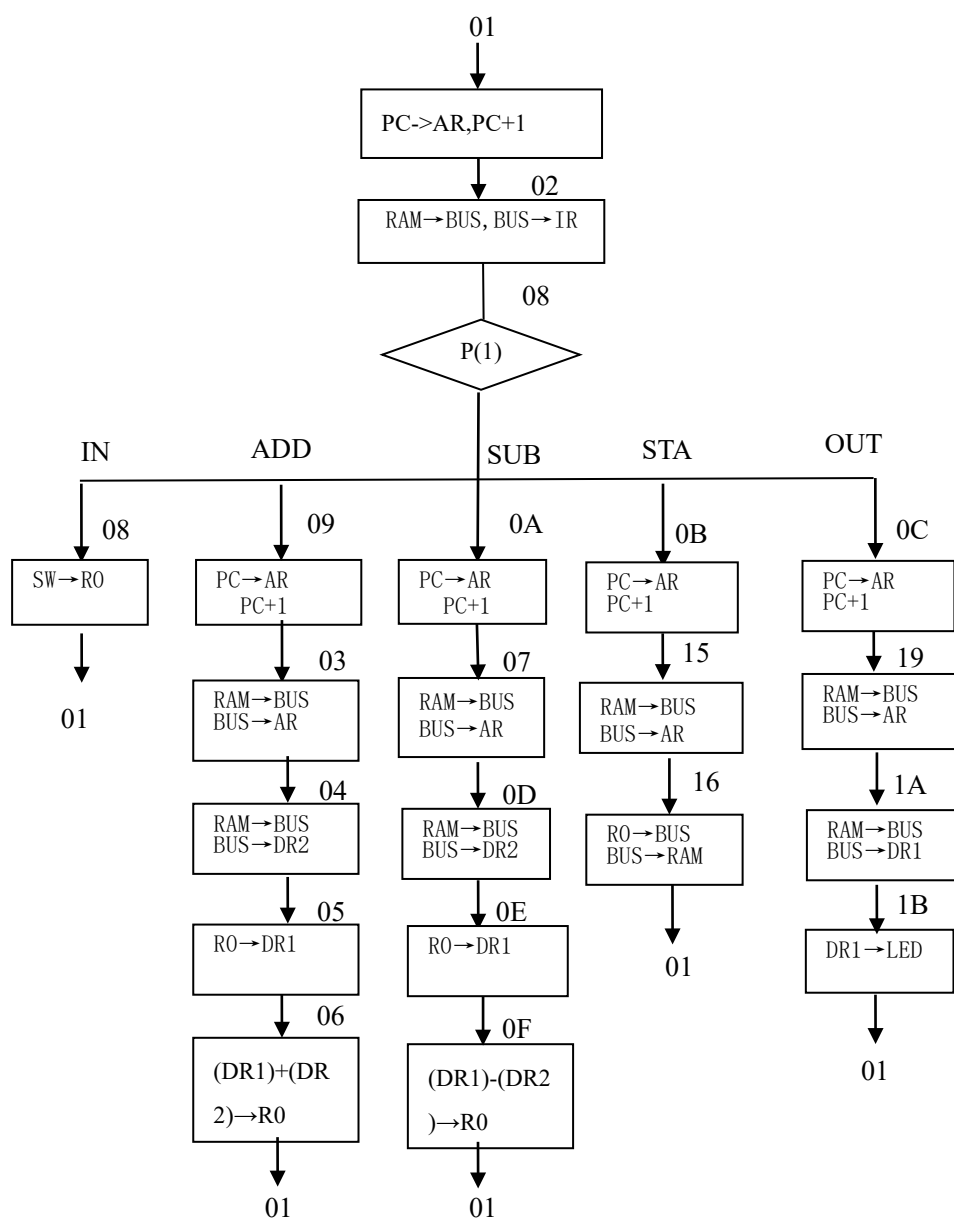


图4.1微程序流程图

4.2 微指令格式设计

微指令格式：

表 2.2 微指令格式图

24	23	22	21	20	19	18	17	16	15-13	12-10	9-7	6	5	4	3	2	1
S3	S2	S1	S0	M	C	W	C	LDP	A	B	C	uA5	uA4	uA3	uA2	uA1	uA0
					n	E	E	C									

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

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

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	SW-B
1	1	1	PC-B

其中 UA5~UA0 为下一条微指令微地址，A、B、C 为三个译码字段，分别由三个控制位译码出多种不同控制信号。

A 字段中的 LDRi 为打入工作寄存器信号的译码器使能控制位。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。C 字段中的 P(1)~P(4) 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微地址入口，从而实现微程序的顺序、分支、循环运行。AR 为算术运算是否影响进位及零标志位控制位。

指令格式的设计见 2.1.3 指令系统。

4.3 二进制微代码表设计

根据流程图，设计各机器指令的微程序表，如表 3.1 所示

表 3.1 二进制微代码表

微地址	S3 S2 S1 S0 M C N W E B1	A	B	C	UA5...UA0
	B0				
00	0 0 0 0 0 0 0 1 1	000	000	100	010000
01	0 0 0 0 0 0 0 1 1	110	110	110	000010
02	0 0 0 0 0 0 0 0 1	100	000	001	001000
03	0 0 0 0 0 0 0 0 1	110	000	000	000100
04	0 0 0 0 0 0 0 0 1	011	000	000	000101
05	0 0 0 0 0 0 1 1	010	001	000	000110

06	1 0 0 1 0 0 0 1 1	001	101	000	000001
07	0 0 0 0 0 0 0 0 1	110	000	000	001101
08	0 0 0 0 0 0 0 0 0	001	000	000	000001
09	0 0 0 0 0 0 1 1 1	110	110	110	000011
0A	0 0 0 0 0 0 1 1 1	110	110	110	000111
0B	0 0 0 0 0 0 1 1 1	110	110	110	001110
0C	0 0 0 0 0 0 1 1 1	110	110	110	010110
0D	0 0 0 0 0 1 0 1 1	000	001	000	000001
0E	0 0 0 0 0 0 0 1 1	110	000	000	001111
0F	0 0 0 0 0 0 0 1 1	010	000	000	010101
10	0 0 0 0 0 0 1 1 1	110	110	110	010010
11	0 0 0 0 0 0 1 1 1	110	110	110	010100
12	0 0 0 0 0 0 0 1 1	010	000	000	010111
13	0 0 0 0 0 0 1 1 1	000	000	000	000001
14	0 0 0 0 0 0 0 0 0	010	000	000	011000
15	0 0 0 0 0 1 1 1 0	000	101	000	000001
16	0 0 0 0 0 0 0 1 1	101	000	110	000001
17	0 0 0 0 0 1 1 1 0	000	101	000	010000
18	0 0 0 0 0 1 0 1 1	000	101	000	010001
19	0 0 1 1 1 0 0 1 1	001	101	000	000001
1A	0 0 0 0 1 0 1 1 1	001	001	000	000001
1B	0 0 0 0 0 0 1 1 1	110	110	110	011110
1C	0 0 0 0 0 0 1 1 1	110	110	110	100010
1D	0 0 0 0 0 0 1 1 1	110	110	110	100100
1E	0 0 0 0 0 0 0 1 1	110	000	000	011111
1F	0 0 0 0 0 0 0 1 1	011	000	000	100000
20	0 0 0 0 0 0 1 1 1	010	001	000	100001
21	0 1 1 0 0 0 1 1 1	010	001	000	000001
22	0 0 0 0 0 0 0 1 1	110	000	000	100011
23	0 0 0 0 0 0 0 1 1	010	000	000	000001
24	0 0 0 0 0 0 0 1 1	110	000	000	100101
25	0 0 0 0 0 0 0 1 1	010	000	000	100110
26	0 0 0 0 1 0 1 1 1	101	101	110	000001

第5章 系统调试及运行报告

5.1 微程序的设计

5.1.1 微程序

微程序：

\$M00 108101

\$M01 82ED01

\$M02 48C000

\$M03 04E000

\$M04 05B000

\$M05 06A201

\$M06 019A91

\$M07 0DE000

\$M08 011000

\$M09 83ED01

\$M0A 87ED01

\$M0B 8EED01

\$M0C 96ED01

\$M0D 018202

\$M0E 0FE000

\$M0F 15A000

\$M10 92ED01

\$M11 94ED01

\$M12 17A000

\$M13 018001

\$M14 182000

\$M15 010A03

\$M16 81D100

\$M17 100A03

\$M18 118A02
\$M19 019A39
\$M1A 019205
\$M1B 9EED01
\$M1C A2ED01
\$M1D A4ED01
\$M1E 1FE000
\$M1F 20B000
\$M20 21A201
\$M21 01A261
\$M22 23E000
\$M23 01A000
\$M24 25E000
\$M25 26A000
\$M26 81DB05

5.1.2 监控程序

监控程序

\$P00 00
\$P01 10
\$P02 0A
\$P03 20
\$P04 0B
\$P05 30
\$P06 0B
\$P07 40
\$P08 00
\$P0A 01

5.2 调试结果

A. 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。

B. 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。

C. 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。

调试结果：

PC=00

PC=01->AR=00->RAM(00)

RAM(00)->IR=00->微控器

INPUT(07)->R0=07

PC=02->AR=01->RAM(10)

RAM(10)->IR=10->微控器

PC=03->AR=02->RAM(0B)

RAM(0B)->AR=0B->RAM(02)

RAM(02)->DR2=02

R0=07->DR1=07

ALU=05->R0=05

PC=04->AR=03->RAM(50)

RAM(50)->IR=50->微控器

R0=05->299(05)

299(0A)->R0=0A

PC=05->AR=04->RAM(20)

RAM(20)->IR=20->微控器

PC=06->AR=05->RAM(0C)

RAM(0C)->AR=0C->RAM(0A)

R0=0A->RAM(0A)

PC=07->AR=06->RAM(30)

RAM(30)->IR=30->微控器

PC=08->AR=07->RAM(0C)

RAM(0C)->AR=0C->RAM(0A)

RAM(0A)->DR1=0A->ALU=0A

ALU=0A->OUTPUT(0A)

PC=09->AR=08->RAM(40)

RAM(40)->IR=40->微控器

PC=0A->AR=09->RAM(00)

RAM(00)->PC=00

5.3 调试时的问题及解决

调试时的问题及解决：

在调试程序时，由于理论和实践没有很好的结合在一起，我们遇到了很多问题，总结起来有以下几点：

1.接线错误。

例如：没有检查排线是否正常或者由于粗心，排线的插孔没有对齐，排线接错，导致程序运行错误

2.在写程序时的错误

例如：在推导 RLC 的微地址时，由于只是从理论出发，我们推导通过 P1 测试后 RLC 操作转向的微代码的地址为 1E，经过程的调试，这个地址是错误的，应该为 0D。

我们将 SBB 操作的微指令 DR1-DR0—>R0 的微代码推理错误，经调试，其 S3~S0、M、CN 位应为 011000。

设计总结

这次我做的题目是基本模型机,这个模型机需要包括输入 INT、输出 OUT、存数 STA、加 ADD、跳转 JMP、与 SUB 这六个运算,学院在实验课阶段已经安排了关于简单模型机的学习,所以我对设计充满了好奇。

老师给我们讲授了基本的设计课时分配,我觉得原理很清晰,我们需要将相应的机器指令码设计出来,通过计算机程序将其输入到连接好的实验箱上去,关于从 SN~UA0 的 24 位二进制代码,我学习过基本模型机指令结构图,了解到微地址是以十六进制来设计的,S3~CN 是逻辑算术运算,WE=1 时代表写,WE=0 时代表读,UA5~UA0 代表下址这些知识虽然很简单,但确实是我规划主程序流程图的最重要依据,关于 A 字段,B 字段,P 字段,从指导书上的对应意义可以轻松的得出。关于机器指令程序阶段的部分,对我来说是最难的部分,按照我先设计的流程图将相应的地址的内容准确的告诉给实验箱,一切衔接起来让我觉得这个课设很有趣,中间走了很多弯路,当我们请教老师问题时,老师很细心的给我们讲解,很清晰,很明了,我们很庆幸有他做我们的指导老师。其实,这个课程设计并不难,老师是想给我们找点计算机组成原理的自己的感觉,所有的都是由那些琐碎的知识结合在一起构成的。期间,最耽误时间、考验耐心和细心的就是指令的存储和机器指令的输入。

在这次设计中,我了解了实验的原理,明确了实验的步骤,大概了解了实验箱的使用。在设计时锻炼了我们的实际动手能力和将理论和实践结合起来的能力。看来,理论和实践是有一定的差别的,它俩在结合的过程中,不是我们想象的那么简单的。在设计的过程中,让我们明白了团队意识的重要性,小组成员必须团结起来,在互相有矛盾的情况下,必须先解决矛盾,再将小组成员的意见统一起来。那样才可以使设计完满结束。

在设计的过程中,非常感谢我们的谢老师。

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. 科学出版社, 2006. 8
- [3]王爱英. 计算机组成与结构, 清华大学出版社, 1999
- [4]王诚. 计算机组成与结构, 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

致 谢

感谢任学谢老师对我们细心的指导，使得我们顺利完成。老师渊博的专业知识，严谨的治学态度，精益求精的工作作风，诲人不倦的高尚师德，严以律己、宽以待人的崇高风范，朴实无华、平易近人的人格魅力对我影响深远。

本程序从选题到完成，每一步都是在老师的指导下完成的，倾注了老师大量的心血，并为我指点迷津，帮助我开拓研究思路，精心点拨、热忱鼓励。对老师的感激之情是无法用言语表达的。在此，谨向老师表示崇高的敬意和衷心的感谢！你们永远健康快乐是我最大的心愿！

无论是在专业知识还是在学习生活中的问题，我们问起来时谢老师总是有问必答，而且把问题分析的通透透透，用最简明扼要的语言分析开了最难懂的问题。谢老师真是一个处处为学生着想，认认真真工作的好老师。谢老师认真负责的工作态度，严谨的治学精神和深厚的理论水平使我们受益匪浅，成为我们以后学习和工作的榜样。我觉得任老师决不止是在专业知识对我们的教育，他看来更像一个迷雾中的导航灯，让我们学到知识的同时认识到我们当前的位置和市场的联系，给我们指明地方，有很多思想教育都是让我深思的。在此向谢老师表示深深的感谢和由衷的敬意。

在此，我再次感谢我们的谢老师，谢谢您的教导！

