

武汉理工大学

课 程 设 计

课程名称 计算机组成原理

设计题目 模型机设计与实现

班 级 软件工程 1604

学 号 0121610870910

姓 名 冯 钢 果

指导教师 田 小 华

日 期 2018 年 6 月 28 日

课程设计任务书

学生姓名： 冯钢果 专业班级： 软件工程 16 级
指导教师： 田小华 工作单位： 计算机学院
题 目： 基本模型机
初始条件：

1. 完成<<计算机组成原理>>课程教学与实验
2. TD-CMA 计算机组成原理教学实验系统

要求完成的主要任务：（包括课程设计工作量及其技术要求，以及说明书撰写等具体要求）

1. 掌握复杂指令系统计算机的微控制器功能与结构特点
2. 熟悉 TD-CMA 教学实验系统的微指令格式
3. 设计五条机器指令，并编写对应的微程序
4. 在 TDN-CMA 教学实验系统中调试机器指令程序，确认运行结果
5. 建立复杂指令系统计算机的整机概念模型

时间安排：

1. 第 16 周周五(6 月 7 日)晚上 16:00~17:40:
软件 15 级全体同学集中讲解课程设计原理与方法
2. 第 17 周周一~五(6 月 12~16 日):
分班实验, 调试机器指令程序, 撰写课程设计报告

指导教师签名： 田小华 2018 年 6 月 28 日

系主任（或责任教师）签名： 2018 年 6 月 28 日

基本模型机

1 课程设计目的

- (1) 掌握一个简单 CPU 的组成原理；
- (2) 在掌握部件单元电路的基础上，进一步将其构造一台基本模型计算机；
- (3) 为其定义五条机器指令，编写相应的微程序，并上机调试掌握整机概念。

2 课程设计设备

- (1) PC 机一台；
- (2) TDN-CM 计算机组成原理实验系统；
- (3) 排线若干。

3 课程设计要求

3.1 实验原理

本次课程设计实现一个简单的 CPU，由此构建一个简单模型计算机。CPU 由 ALU、微控制器（MC）、通用寄存器（R0），指令寄存器（IR）、程序计数器（PC）和地址寄存器（AR）组成，如图 1 所示。

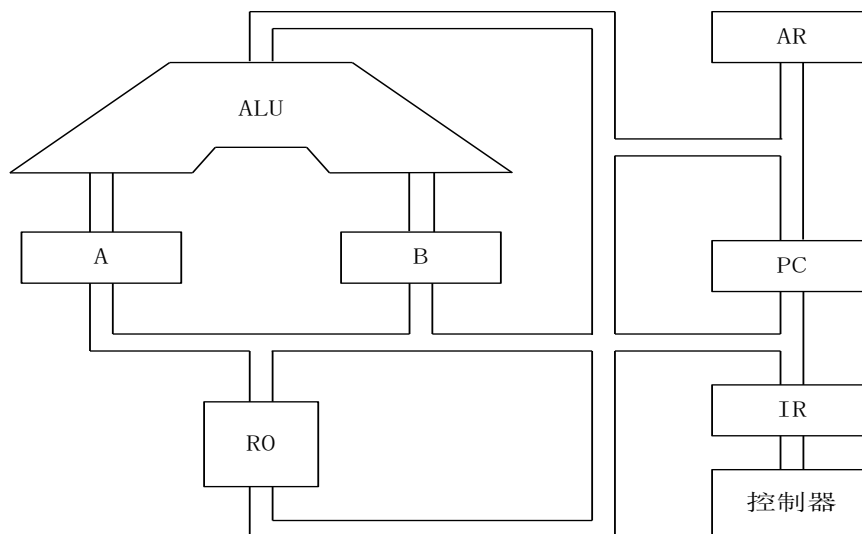


图 1 基本 CP 构成原理图

模型机 CPU 在写入相应的微指令后，就具备执行相应机器指令的功能。在 CPU 基础上，增加配套的主存和基本的输入输出部件，构成一个简单的模型计算机。

3.1.1 机器指令

课程设计采用五条机器指令：IN, ADD, OUT, JMP, HLT, 对应的源程序内容如下：

地址	内 容	助记符	说明
0000 0000	0010 0000	;START: IN R0	从 IN 单元读入数据送 R0
0000 0001	0000 0000	;ADD R0,R0	R0 和自身相加，结果送 R0
0000 0010	0011 0000	;OUT R0	R0 的值送 OUT 单元显示
0000 0011	1110 0000	;JMP START	跳转至 00H 地址
0000 0100	0000 0000	;	
0000 0101	0101 0000	;HLT	停机

指令码中高 4 位为操作码，JMP 为双字节指令，其余为单字节指令。微控制器实验的指令是手动/联机给出，模型机实验要求 CPU 自动从存储器读取机器指令并执行。

3.1.2 微指令格式

模型机实验在微控制器实验的基础上，增加了 PC、AR 和主存，在微指令中应增加相应的控制位，其微指令格式见图 2。

23	22	21	20	19	18-15	14-12	11-9	8-6	5-0
M23	M22	WR	RD	IOM	S3-S0	A 字段	B 字段	C 字段	MA5-MA0

A 字段				B 字段				C 字段			
14	13	12	选择	11	10	9	选择	8	7	6	选择
0	0	0	NOP	0	0	0	NOP	0	0	0	NOP
0	0	1	LDA	0	0	1	ALU_B	0	0	1	P<1>
0	1	0	LDB	0	1	0	RO_B	0	1	0	保留
0	1	1	LDRO	0	1	1	保留	0	1	1	保留
1	0	0	保留	1	0	0	保留	1	0	0	保留
1	0	1	LOAD	1	0	1	保留	1	0	1	LDPC
1	1	0	LDAR	1	1	0	PC_B	1	1	0	保留
1	1	1	LDIR	1	1	1	保留	1	1	1	保留

图 2 微指令格式

3.1.3 数据通路图

根据以上设计要求，相关的数据通路图见图 3。对应的控制信号控制相关功能部件的操作。从图中可以看出机器指令程序运行的大致过程。

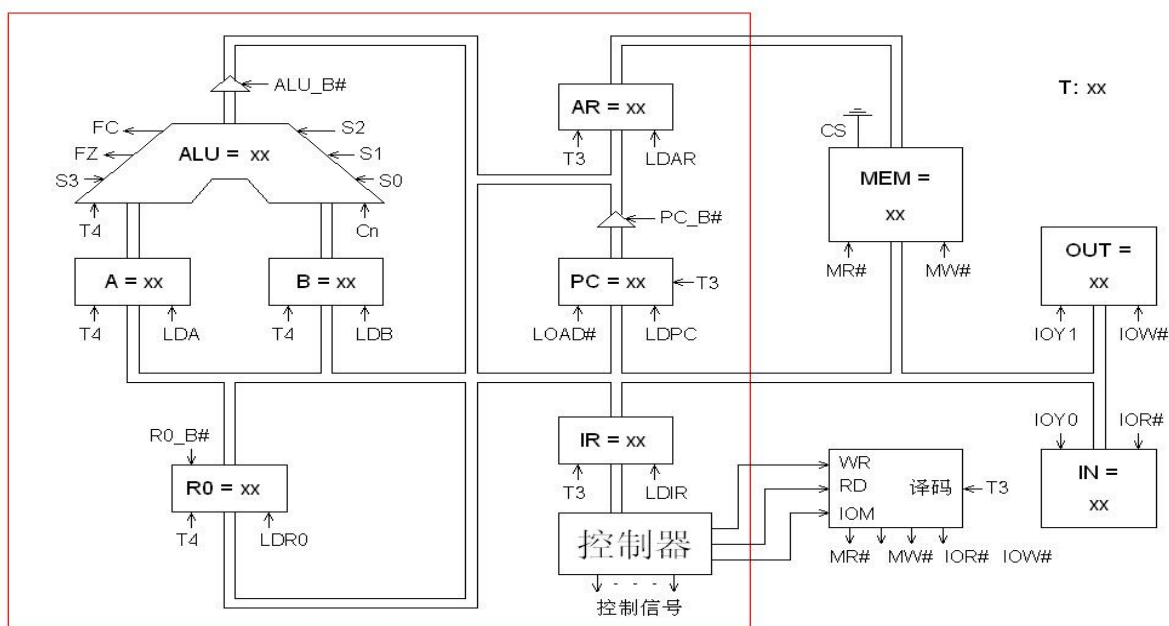


图 3 模型机数据通路

3.1.4 微程序流程图

机器指令译码原理见图 5-4 所示，相关部件结构原理图分别见图 4~图 8 所示。模型机系统涉及到的微程序流程图，详见图 9。当拟定“取指”微指令时，该微指令的判别测试字段为 P<1>测试。

由于“取指”微指令是所有微程序都使用的公用微指令，因此 P<1>的测试结果出现多路分支。本机使用指令寄存器的高 6 位（IR7-IR2）作为测试条件，出现 5 路分支，占用 5 个固定微地址，微程序流程图上的单元地址均为 16 进制编码的地址。

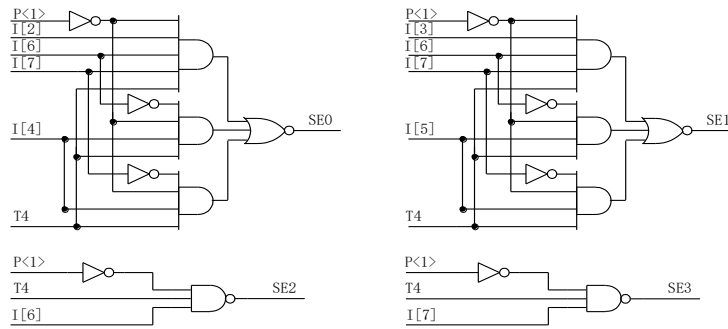


图 4 指令译码原理图

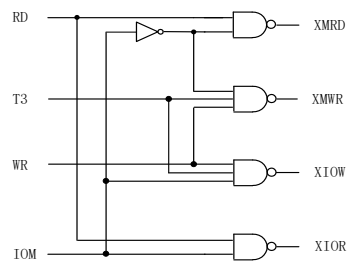


图 5 读写控制逻辑

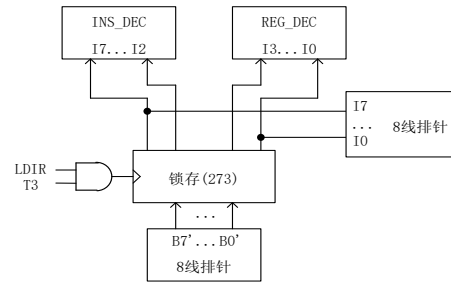


图 6 IR 单元原理图

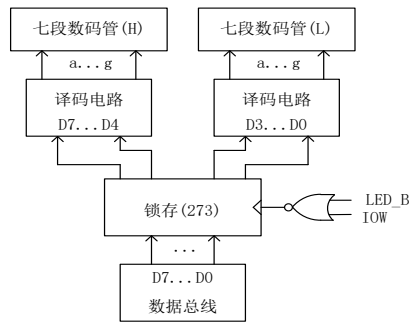


图 7 OUT 单元原理图

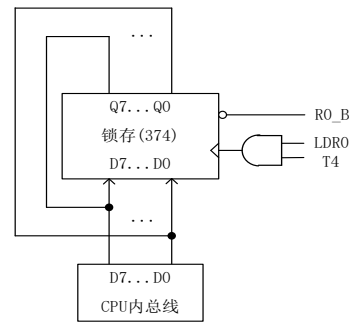


图 8 RO 原理图

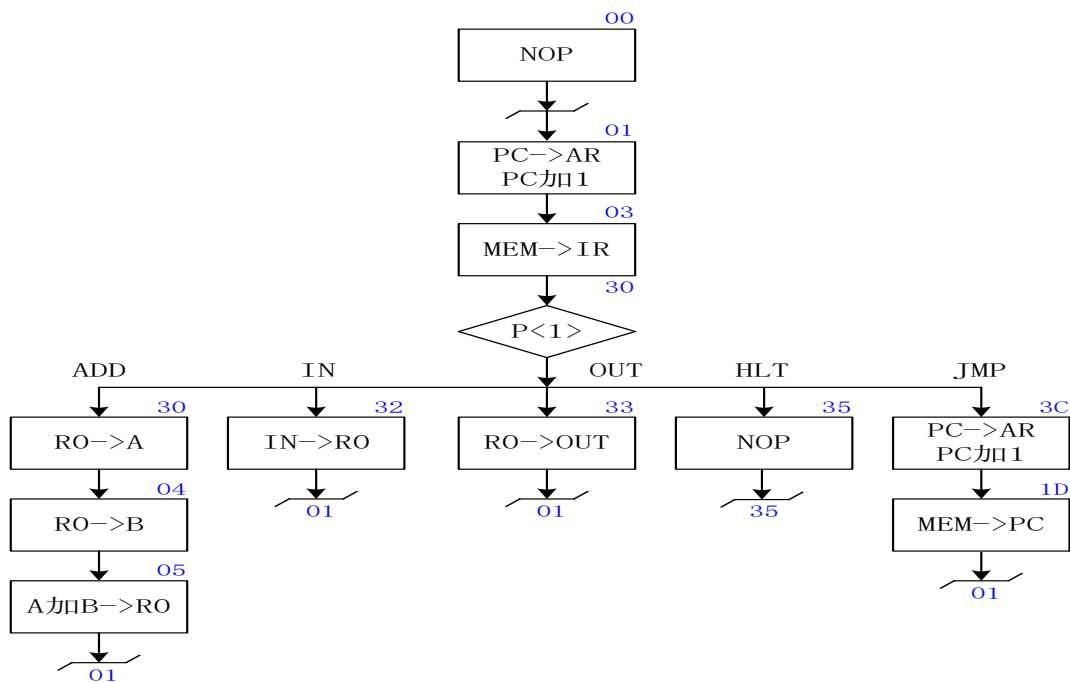


图 9 微程序流程图

每条机器指令与一段特殊的微程序相对应，该微程序执行完成意味着与之对应的机

器指令功能就实现了。

3.1.5 微指令二进制微代码表

全部微程序设计完毕，应将每条微指令代码化，按序排列，见表 1。

地址	十六进制内容	高五位	$S_3S_2S_1S_0$	A 字段	B 字段	C 字段	$MA_5\sim MA_0$
00H	00 00 01	00000	0000	000	000	000	000001
01H	00 6D 43	00000	0000	110	110	101	000011
03H	10 70 70	00010	0000	111	000	001	110000
04H	00 24 05	00000	0000	010	010	000	000101
05H	04 B2 01	00000	1001	011	001	000	000001
1DH	10 51 41	00010	0000	101	000	101	000001
30H	00 14 04	00000	0000	001	010	000	000100
32H	18 30 01	00011	0000	011	000	000	000001
33H	28 04 01	00101	0000	000	010	000	000001
35H	00 00 35	00000	0000	000	000	000	110101
3CH	00 6D 5D	00000	0000	110	110	101	011101

表 1 二进制微代码表

3.1.6 机器指令程序

设计一段机器指令程序：从 IN 单元读入自己的学号，存放于 R0, 将 R0 和 R0 相加，结果存于 R0, 再将 R0 的值送到 OUT 单元显示。机器指令程序如下，地址和内容均为二进制：

地址	内 容	助记符	说明
0000 0000	0010 0000	;START: IN R0	序号 → R0
0000 0001	0000 0000	;ADD R0,R0	R0 + R0 → R0
0000 0010	0011 0000	;OUT R0	R0 → LED
0000 0011	1110 0000	;JMP START	跳转至 00H 地址
0000 0100	0000 0000	;	
0000 0101	0101 0000	;HLT	停机

3.2 实验步骤

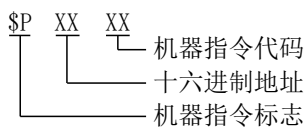
1. 按图 10 接线

接线前，首先检测多芯排线的状况，确保排线接触良好。

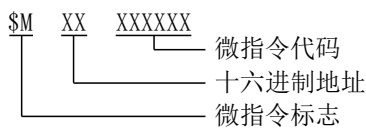
2. 写入实验程序，并进行校验，可用联机写入。

联机软件提供了微程序和机器程序下载功能，以代替手动读写微程序和机器程序，但是微程序和机器程序得以指定的格式写入到以 TXT 为后缀的文件中，微程序和机器程序的格式如下：

机器指令格式说明：



微指令格式说明：



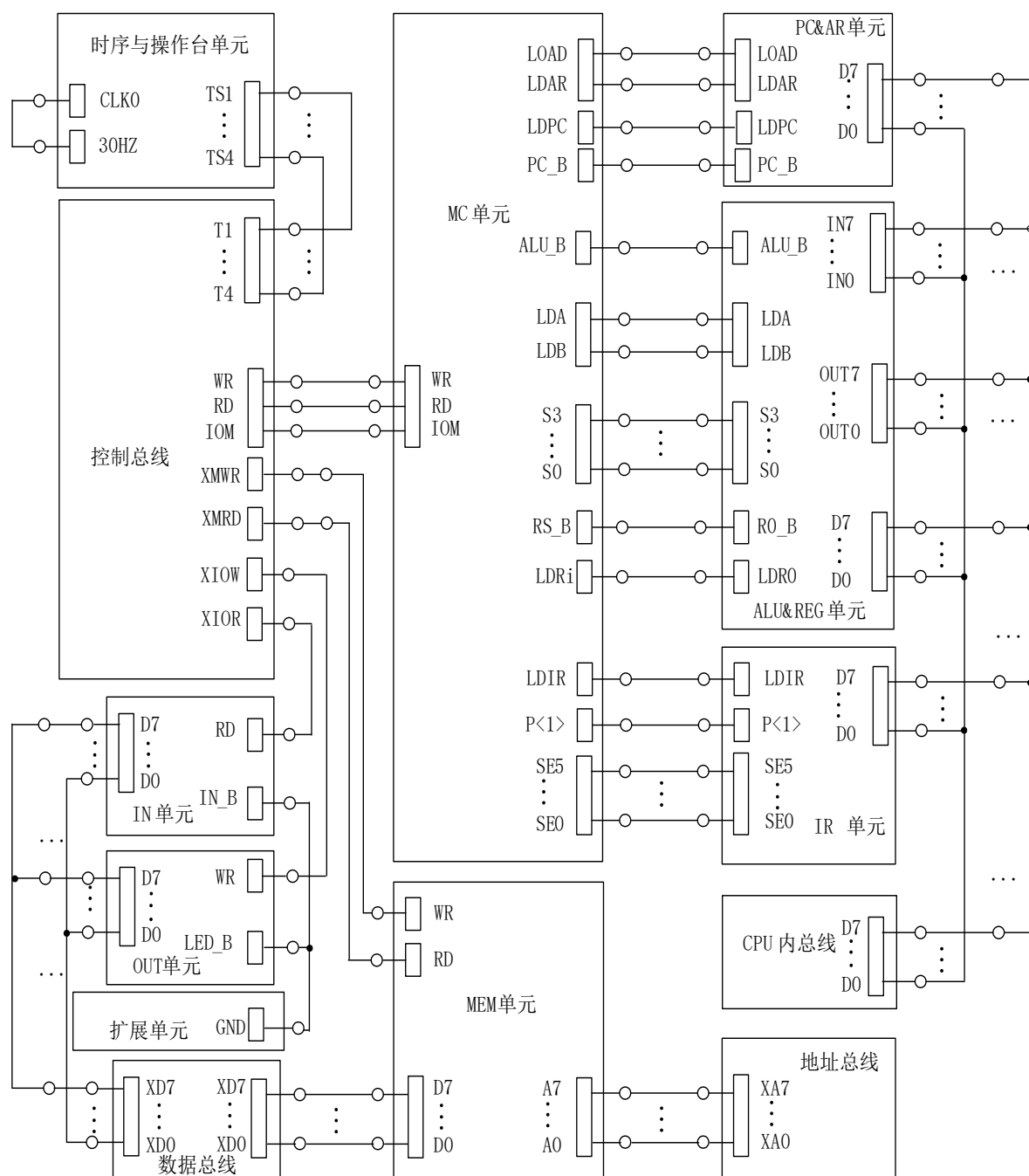


图 10 实验接线图

调试过程:

选择联机软件的“[转储]—[装载]”功能，在打开文件对话框中选择相关文件，软件自动将机器指令程序和微程序写入指定单元。

选择联机软件的“[转储]—[刷新指令区]”功能，可以读出下位机所有的机器指令和微指令，并在指令区显示。对照文件检查微程序和机器指令程序是否正确。

附实验程序如下，程序中分号‘;’为注释符，分号后面的内容在下载时将被忽略掉：（说明：这里是把自己序号 + 75H 得到输入值，IN 值为 75H + 08H = 7D，因此在联机运行时，先把 IN 单元的值调为：0111 1101，即可得到输入联机运行的结果）

```

; //*****//
; //
; //      CPU 与简单模型机实验指令文件      //
; //
; //      By TangDu CO.,LTD      //
; //
; //*****//

; //***** Start Of Main Memory Data *****//
$P 00 20      ; START: IN  R0      从 IN 单元读入数据送 R0
$P 01 00      ; ADD R0,R0          R0 和自身相加, 结果送 R0
$P 02 30      ; OUT R0            R0 的值送 OUT 单元显示
$P 03 E0      ; JMP START         跳转至 00H 地址
$P 04 00      ;
$P 05 50      ; HLT                停机
; //***** End Of Main Memory Data *****//

; //**** Start Of MicroController Data ****//
$M 00 000001   ; NOP
$M 01 006D43   ; PC->AR, PC 加 1
$M 03 107070   ; MEM->IR, P<1>
$M 04 002405   ; R0->B
$M 05 04B201   ; A 加 B->R0
$M 1D 105141   ; MEM->PC
$M 30 001404   ; R0->A
$M 32 183001   ; IN->R0
$M 33 280401   ; R0->OUT
$M 35 000035   ; NOP
$M 3C 006D5D   ; PC->AR, PC 加 1
; //**** End Of MicroController Data ****//

```

3. 联机运行程序

将 KK1、KK3 置为“运行”档，进入软件界面，选择菜单命令“[实验]—[简单模型机]”，打开简单模型机数据通路图。并在 IN 单元输入 0111 1101。

按动 CON 单元的总清开关 CLR，然后通过软件运行程序，选择相应的功能命令，即可联机运行、监控、调试程序，当模型机执行完 JMP 指令后，检查 OUT 单元显示的数据是否为 OUT 单元值的 2 倍。在数据通路图和微程序流中观测指令的执行过程，并观察软件中地址总线、数据总线以及微指令显示和下位机是否一致。

4 实验过程现象、结果分析

4.1 实验过程记录

- (1) 当联机开始运行时，先将 IN 中数据送入与数据总线连接的 IN 中：（如图 4-1）这里输入的是 0111 1101（即 7D H），因此，IN 单元显示为 7DH。

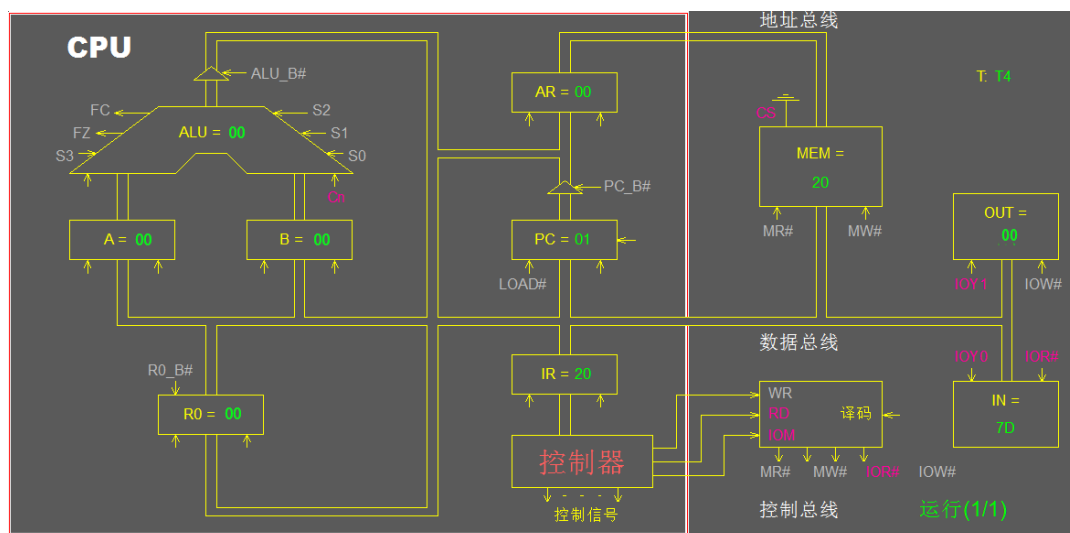


图 4-1

- (2) 然后 MEM 的指令在送到 IR 中，并传到控制器，然后执行指令 IN，将 IN 单元数据读入到 R0 寄存器中。（如图 4-2）

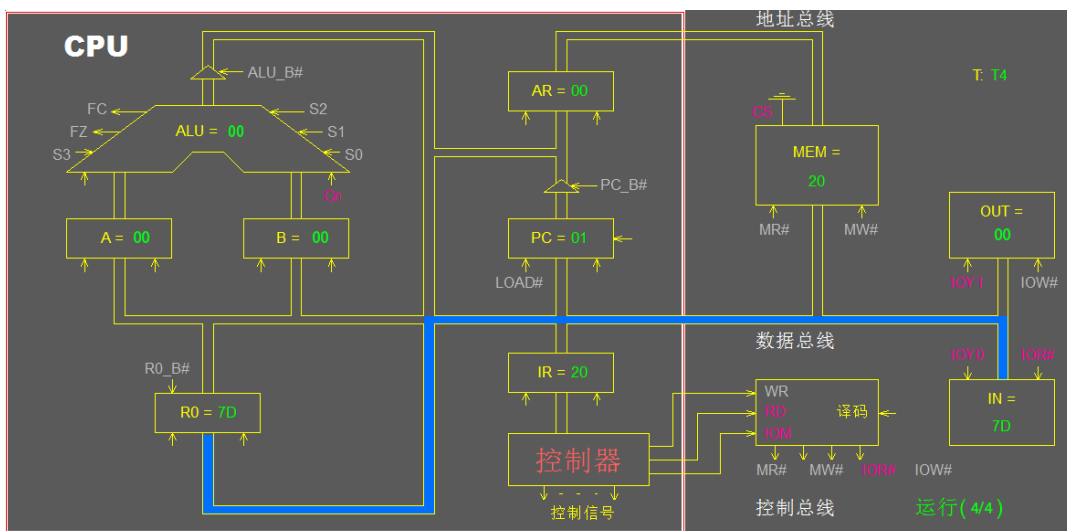
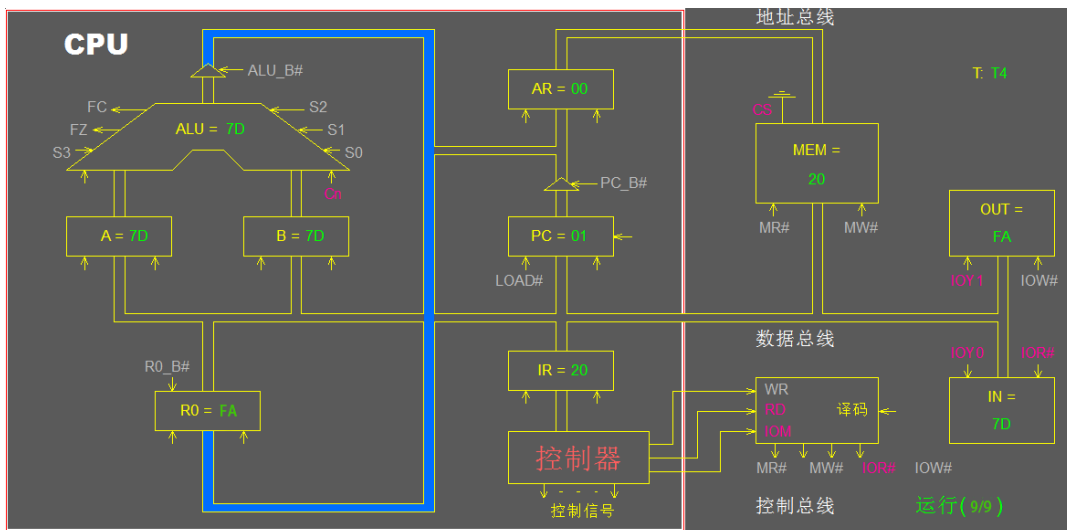


图 4-2

- (3) 然后在读取其他指令，即将 MEM 的机器指令送到 IR 中，再到控制器进行执行，直到执行到将数据分别读入到 A 和 B 中暂存，然后再自身相加。（如图 4-3）



- (4) 然后执行 OUT 指令将结果从 R0 送至 OUT 单元，此时 LED 显示也为 FA，然后再执行停机操作即可。（如图 4-4）

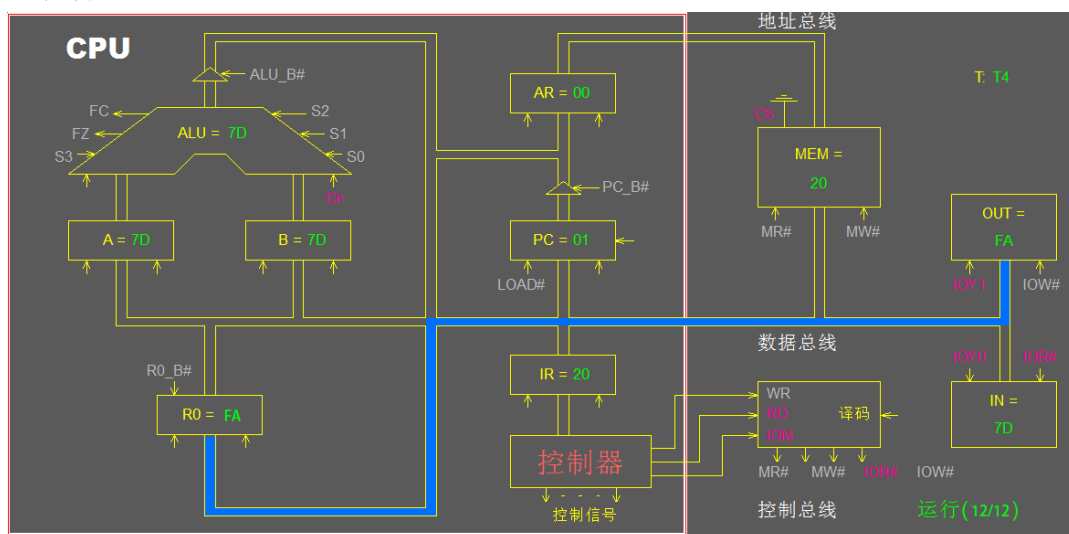


图 4-4

4.2 结果分析

根据上述实验现象，可以得到以下结论：

- (1) 实验结果为 FA，而输入的为 7D，得到结果应为 $7D + 7D = FA$ ，因此得到期望结果，实验成功。
- (2) 指令执行过程中，先将 $PC+1$ ，获取要执行指令，再送到控制器，由控制器做出相应的指令。
- (3) 数据获取一般保存在寄存器中，而进行数据相加，一般是先传入一个参数，紧接着传入另一个参数，然后有 ALU 进行算术或逻辑运算，送到寄存器中。
- (4) 最终得到结论：当前模拟的计算机是主要由数据总线、控制总线保证计算机指令正常执行，并且数据和指令存放在一起，典型的冯·诺依曼计算机模型，并且计算机处理的是二进制数据，在这里为了便于操作者观察，联机运行一般显示的是 16 进制，这里仅仅实现加法，加法与此类似。

5 课程设计总结

5.1 每条机器指令的微程序：IN, ADD, OUT, JMP, HLT

（注：执行地址顺序由上到下）

指令	执行地址顺序	十六进制
IN	01H	00 6D 43H
	03H	10 70 70H
	32H	18 30 01H
ADD	01H	00 6D 43H
	03H	10 70 70H
	30H	00 14 04H
	04H	00 24 05H
	05H	04 B2 01H
JMP	01H	00 6D 43H
	03H	10 70 70H
	3CH	00 6D 5DH
	1DH	10 51 41H

续表：

指令	执行地址顺序	十六进制
OUT	01H	00 6D 43H
	03H	10 70 70H
	33H	28 04 01H
HLT	01H	00 6D 43H
	03H	10 70 70H
	35H	00 00 35H

5.2 课程设计心得、经验教训及注意事项

(1) 课程设计心得

- ① 本次实验主题是“基本模型机”，通过动手实验，我掌握了一个简单的 CPU 组成原理，并在这些简单连线的基础上构造出了一个简单的基本模型计算机。
- ② 实验过程中主要用到了五条机器指令，编写微程序，五条指令的核心指令为 ADD 指令，结合学过的汇编语言的知识，在实验前可以猜测到 ADD 运算是分别把暂存器中的数据相加，并写入到寄存器中，而实验中可以理解为一个三操作数的 ADD 指令，或者理解为一个简单的二操作数与 MOV 指令结合，如下列转换式子：
 MOV AX, A
 MOV BX, B
 ADD AX, BX
 MOV R0, AX
 上式为 IBM-PC 8086 汇编语言格式，通过上式可以简单的将 A 与 B 相加，但我认为上式更有助于理解计算过程：先将要操作的数值保存到寄存器中，然后进行寄存器操作运算，最后再将值传回 R0 即可，当然，ADD 运算便于实现。
- ③ 对于加法运算 SUB、乘法运算 MUL 和除法 DIV 运算，除了 SUB 与 ADD 相似，MUL、DIV 与 ADD 有较大差别，这两个指令再 8086 指令系统中是单操作数，直接对寄存器 AX 和 DX 进行操作，我认为在进行实验拓展时可以根据情况加入 MUL 指令，或者设计 ADD 加法产生进位的情况。
- ④ 本次模拟的简单模型机可以看作简单的冯·诺依曼计算机，模型机主要由数据总线、控制总线保证计算机指令正常执行，并且数据和指令存放在一起，并且计算机处理的是二进制数据，因此符合冯·诺依曼计算机特点。

(2) 注意事项和经验教训

- ① 注意 1：连线一定要十分小心谨慎，不能有任何差错，并且不能看似破损的线；
- ② 注意 2：连线完成后进行联机实验不要忘记再实验箱的 IN 单元输入数据，再联机过程中最好使用单步或单周期运行以便于观察现象；
- ③ 注意 3：在进行联机运行是最好进行联机测试，确保 PC 机与实验箱连接畅通，并进行测试连线，再次确保连线没有大的错误；
- ④ 注意 4：联机运行开始前不要忘记按 con 单元的 CLR 清空当前实验箱寄存器中数据，否则会导致结果错误。
- ⑤ 教训 1：在实验过程连线不认真，把 8 排线的插口错位导致实验错误；
- ⑥ 教训 2：在第一次联机运行中，忘记 IN 单元置数，导致实验结果一直为 0；
- ⑦ 教训 3：在多次实验中，有时忘记按 CLR 导致实验结果错误；
- ⑧ 教训 4：在第一次联机运行，没有进行接口测试，导致计算机一直提示 COM1 不能使用，从而不能控制实验箱，经过选择 COM 接口解决问题。

本科生课程设计成绩评定表

班级：软件 1604 班

姓名：冯钢果

学号：0121610870910

序号	评分项目	满分	实得分
1	学习态度认真、遵守纪律	10	
2	设计分析合理性	10	
3	设计方案正确性、可行性、创造性	20	
4	设计结果正确性	40	
5	设计报告的规范性	10	
6	设计验收	10	
		总得分/等级	
评语：			

注：最终成绩以五级分制记。优（90-100 分）、良（80-89 分）、中（70-79 分）、及格（60-69 分）、60 分以下为不及格

指导教师签名：田小华

2018 年 6 月 28 日