计算机组成原理实验指导书

Principles of Computer Organization Experiment Instruction Book

实验 5 简单模型机实验

燕山大学软件工程系

实验 5 简单模型机实验

5.1 实验目的

（1）通过总线将微程序控制器、运算器、存储器等联机，组成一台模型计算机。

（2）用微程序控制器控制模型机数据通路，运行由 4 条机器指组成的简单程序。

（3）掌握微指令与机器指令的关系，建立整机概念。

5.2 实验要求

（1）做好实验预习，复习微指令和机器指令的概念，读懂实验电路图，熟悉实验元器件的功能特性和使用方法。

（2）对于实验任务中的问题，在实验前预先给出答案，以便与实验结果相比较。

（3）在实验过程中单步运行微程序，注意理解微程序与程序的联系和区别。

（4）写出实验报告。

5.3 实验原理

本实验综合了前面实验的电路，将运算器模块、存储器模块和控制器模块通过总线连接在一起，组成了一个简单的模型机，其电路如图 5.1 所示。

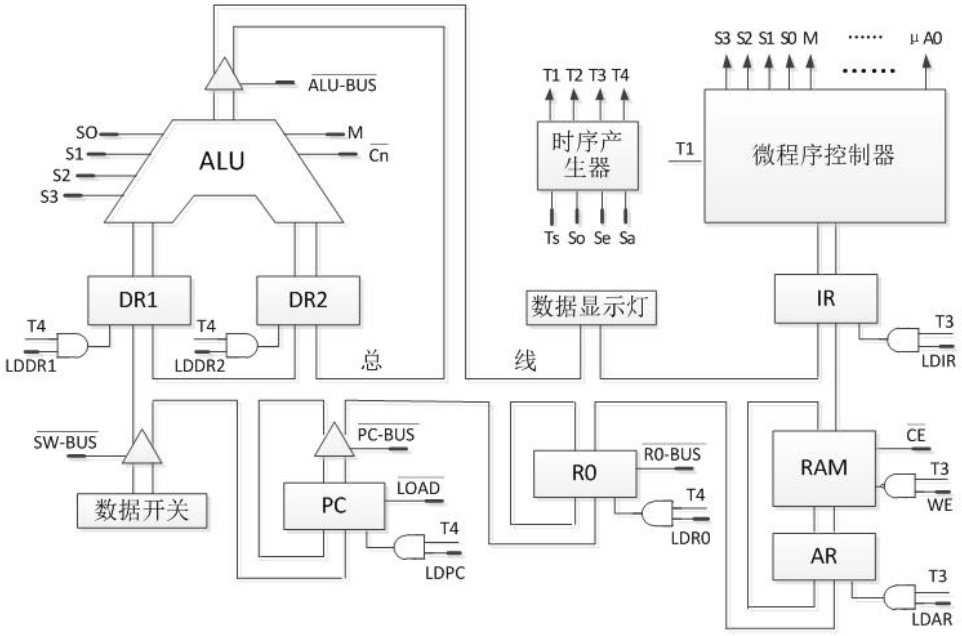

图 5.1 简单模型机总框图

实验电路中涉及的主要控制信号如下：

（1）M：选择 ALU 的运算模式(M=0，算术运算；M=1，逻辑运算)

（2）S3，S2，S1，S0：选择 ALU 的运算类型。如 M=0 时，设为 1001 表示加法运算。

（3）$\overline{Cn}$：向 ALU 最低位输入的进位信号，$\overline{Cn}$=0 时有进位输入，$\overline{Cn}$=1 时无进位输入。

（4）LDDR1：DR1 的数据加载信号，LDDR1=1 时在 T4 的上升沿将数据锁存到 DR1。

（5）LDDR2：DR2 的数据加载信号，LDDR2=1 时在 T4 的上升沿将数据锁存到 DR2.。

（6）$\overline{ALU-BUS}$：ALU 输出三态门使能信号，为 0 时将 ALU 运算结果输出到总线。

（7）$\overline{SW-BUS}$：开关输出三态门使能信号，为 0 时将 SW7~SW0 数据发送到总线。

（8）$\overline{PC-BUS}$：PC 输出三态门使能信号，为 0 时将 PC 的值输出到总线。

（9）$\overline{LOAD}$：PC 的置数信号，为 0 时 PC 工作在置数模式，可在此模式下为 PC 设置初值。

（10）LDPC：PC 的加载信号，为 1 时在 T4 的上升沿执行清零、置数或计数操作。

（11）$\overline{\text{R0-BUS}}$：R0 芯片的输出控制信号，为 0 时将 R0 中的数据输出到总线。

（12）LDR0：R0 的数据载入信号，为 1 时在 T4 上升沿将数据存入 R0。

（13）LDIR：IR 的加载信号，当 LDIR=1 时在 T3 的上升沿将指令锁存到 IR。

（14）$\overline{\text{CE}}$：6116 片选信号，为 0 时 6116 正常工作。

（15）WE：存储器写信号，在 $\overline{\text{CE}}$=0、$\overline{\text{OE}}$=0 的条件下，当 WE=1 且 T3=1 时进行写操作，否则进行读操作。

（16）LDAR：AR 的地址加载信号，当 LDAR=1 时在 T3 的上升沿将地址锁存到 AR。

（17）T1~T4：时序信号，对应一个 CPU 周期。

（18）Ts，So，Se，Sa：Ts 为时钟源输入，So 为停止信号，Sa 为开始信号，Se 为单步运行。

在控制器实验中，实现了自动按照 AR 中的指令逐条取出对应的微指令。在本实验中，程序存储在 RAM 中，微程序存储在控制存储器中，要实现自动从 RAM 里逐条取出指令放入 IR，并按照 IR 中的指令自动从控制存储器读出相应的微程序执行。

本实验用到的微指令长度为 24bit，微指令格式如表 5-1 所示。

表 5-1 微指令格式

位	23	22	21	20	19	18	17	16	15	14	13	12
控制信号	S3	S2	S1	S0	M	$\overline{\text{Cn}}$	$\overline{\text{CE}}$	WE	$\overline{\text{LOAD}}$	LDR0	LDDR1	LDDR2

位	11	10	9	8	7	6	5	4	3	2	1	0
控制信号	LDIR	LDPC	LDAR	$\overline{\text{ALU-B}}$	$\overline{\text{PC-B}}$	$\overline{\text{SW-B}}$	$\overline{\text{R0-B}}$	P(1)	μA3	μA2	μA1	μA0

本实验使用的微程序流程如图 5.2 所示。

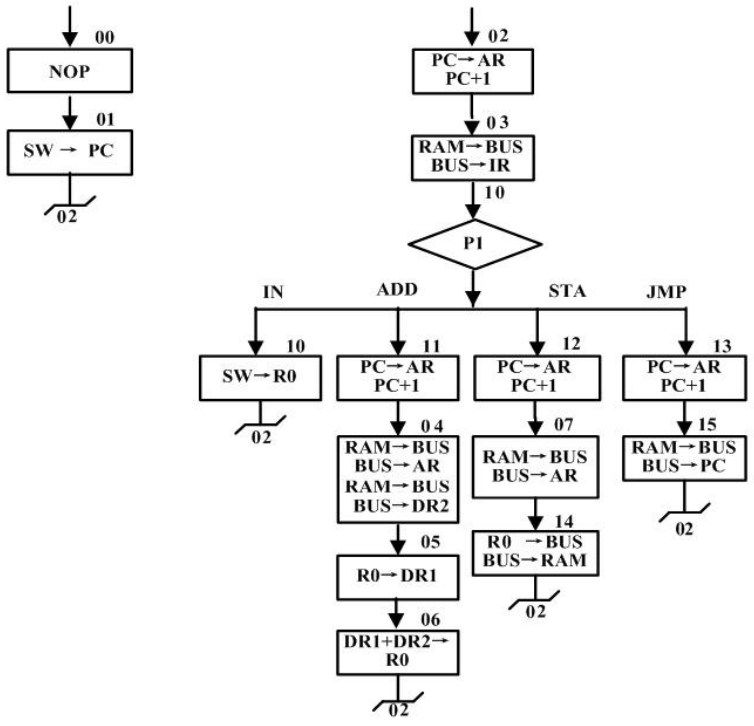

图 5.2 微程序流程图

对应的微程序代码存放在控制存储器中，如表 5-2 所示。

表 5-2 微程序二进制代码表

位	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
地址	S3	S2	S1	S0	M	Cn	CE	WE	LOAD	LDR0	LDDR1	LDDR2	LDIR	LDPC	LDAR	ALU-B	PC-B	SW-B	R0-B	P(1)	μA3	μA2	μA1	μA0
00	0	0	0	0	0	1	1	0	1	0	0	0	0	0	0	1	1	1	1	0	0	0	0	1
01	0	0	0	0	0	1	1	0	0	0	0	0	0	1	0	1	1	0	1	0	0	0	1	0
02	0	0	0	0	0	1	1	0	1	0	0	0	0	1	1	1	0	1	1	0	0	0	1	1
03	0	0	0	0	0	1	0	0	1	0	0	0	1	0	0	1	1	1	1	1	1	0	0	0
04	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	1	1	1	1	0	0	1	0	1
05	0	0	0	0	0	1	1	0	1	0	1	0	0	0	0	1	1	1	0	0	0	1	1	0
06	1	0	0	1	0	1	1	0	1	1	0	0	0	0	0	0	1	1	1	0	0	0	1	0
07	0	0	0	0	0	1	0	0	1	0	0	0	0	0	1	1	1	1	1	0	1	1	0	0
10	0	0	0	0	0	1	1	0	1	1	0	0	0	0	0	1	1	0	1	0	0	0	1	0
11	0	0	0	0	0	1	1	0	1	0	0	0	0	1	1	1	0	1	1	0	0	1	0	0
12	0	0	0	0	0	1	1	0	1	0	0	0	0	1	1	1	0	1	1	0	0	1	1	1
13	0	0	0	0	0	1	1	0	1	0	0	0	0	1	1	1	0	1	1	0	1	1	0	1
14	0	0	0	0	0	1	0	1	1	0	0	0	0	0	0	1	1	1	0	0	0	0	1	0
15	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	1	1	1	1	0	0	0	1	0

一条指令对应一个微程序，一个微程序是多条微指令的有序集合。

模型机共包含 4 条指令，指令格式如表 5-3 所示。本实验用这 4 条指令编写了一个简单程序，并已存入 RAM。RAM 中的程序和数据如表 5-4 所示。

表 5-3 机器指令格式

助记符	机器码（A 为内存地址 8bit）	长度	功能
IN	000XXXXXX	8bit	SW→R0
ADD	001XXXXXX A	16bit	R0+(A)→R0
STA	010XXXXXX A	16bit	R0→(A)
JMP	011XXXXXX A	16bit	A→PC(程序跳转到 A 地址执行)

表 5-4 RAM 中的程序和数据

地址（八进制）	内容	含义
00	00000000	IN（开关数据自定）
01	00100000	ADD
02	00001000	10（八进制）
03	01000000	STA
04	00001001	11（八进制）
05	01100000	JMP
06	00000000	00
07		
10	00001011	
11		求和结果

5.4 实验内容与步骤

1. 运行虚拟实验系统，按照图 5.1 绘制实验电路，生成如图 5.3 所示电路。

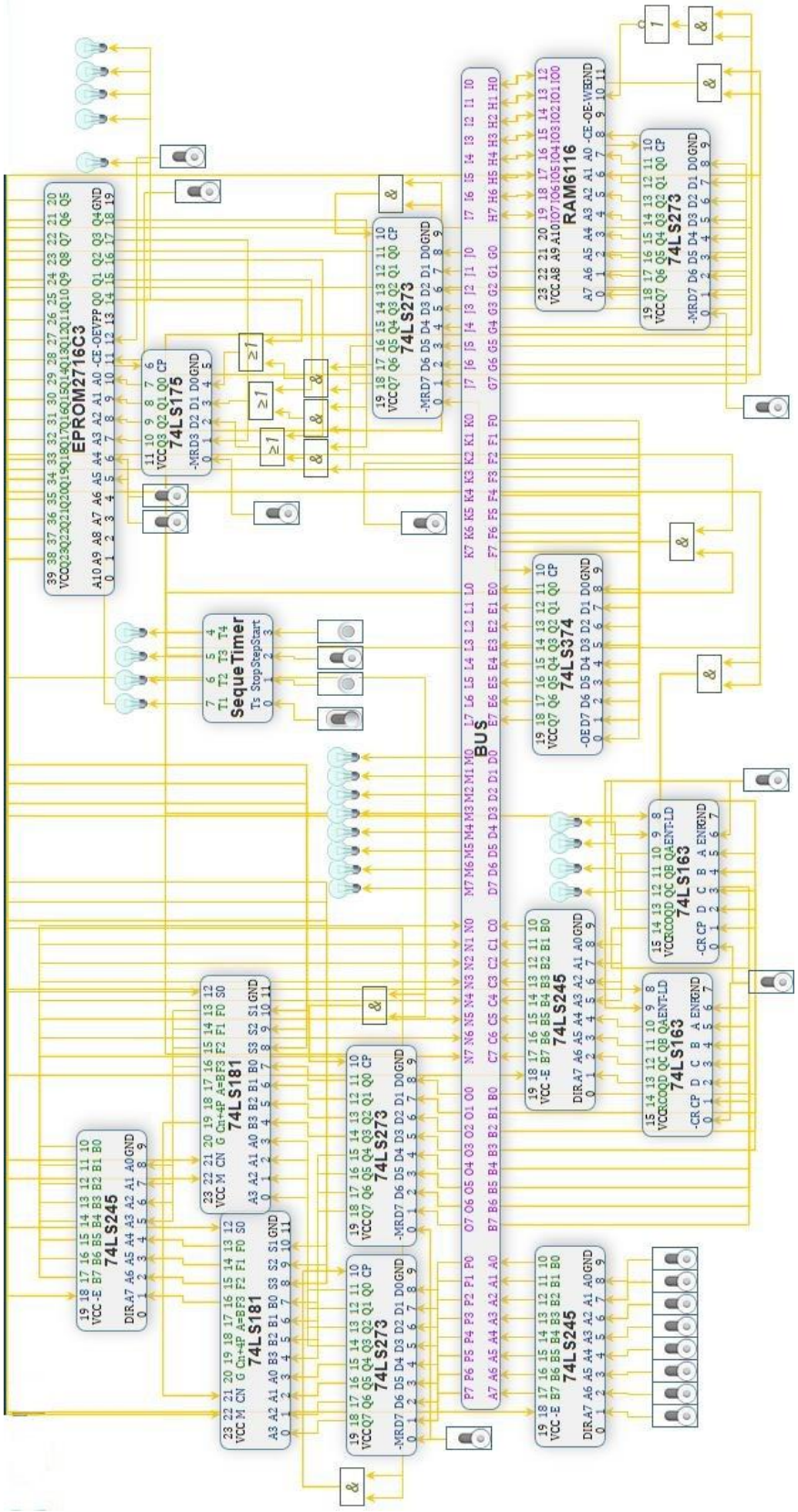

图 5.3 简单模型机虚拟实验电路

2. 打开电源开关。

3. 进行电路预设置。将 DR1、DR2 和 AR 的 $\overline{MR}$ 置 1，将计数器的 $\overline{CR}$ 、ENT、ENP 置 1，时序发生器的 Step 置 1（可在开电源之前设置）。微地址寄存器 74LS175 和指令寄存器 IR 的 $\overline{MR}$ 置 1。此时微地址寄存器和 IR 已初始化为零，模型机将从控制存储器的零地址开始运行。

4. 在数据开关（SW7~SW0）上设置好程序的起始地址（00000000）。

5. 单击 1 次时序发生器的 Start 按钮，思考并回答问题：此时执行的是微程序流程图中的第几条微指令？作用是什么？

6. 再单步执行两条微指令，思考并回答问题：这两条微指令的作用是什么？

7. 通过数据开关（SW7~SW0）设置操作数 1 的值为 00010100。

8. 单击 Start，执行微指令 SW→R0，将操作数 1 保存到累加器 R0 中。

9. 继续单步执行之后的微指令，直到第一轮循环结束。在此过程中注意观察总线上数据灯的显示，并说明每个显示出来的数字的意义，将表 5-5 补充完整。

表 5-5 总线数据表

序号	总线上数据（二进制）	微指令编号（八进制）	意义（地址用二进制表示）
1	00000001	02	当前 PC 的值，即内存地址 01
2	00000010	02	递增 1 后的 PC 值
3	00100000	03	内存地址 01 中的 ADD 指令操作码
4	00000010		
5	00000011		
6	00001000		
7	00001011		
8	00010100		
9	00011111		
10	00000011	02	当前 PC 的值，即内存地址 11
11	00000100	02	递增 1 后 PC 值
12	01000000		
13	00000100	12	当前 PC 的值，即内存地址 100
14	00000101	12	递增 1 后的 PC 值
15	00001001	07	内存地址 100 中的数据，此数据也是一个地址
16	00000000	07	内存地址 1001 中的数据
17	00011111		
18	00000101	02	当前 PC 的值，即内存地址 101
19	00000110		递增 1 后的 PC 值
20	01100000		
21	00000110	13	当前 PC 的值，即内存地址 110
22	00000111	13	递增 1 后的 PC 值
23	00000000		

10. 利用菜单“工具/存储器芯片设置”选项，查看运算结果是否已填入指定内存单元。

5.5 实验结果

本实验需要记录的结果是回答 5.4 节实验内容与步骤中，第 5、6、9、10 步提出的问题：

5. 答：

6. 答：

9. 将表 5-5 补充完整。

10. 对每条指令的执行结果进行说明，并将使用菜单“工具/存储器芯片设置”选项，查看运算结果时的界面进行截图。

指令		结果说明
IN	SW→R0	
ADD	R0+（A）→R0	
STA	R0→（A）	
JMP	A→PC	

5.6 思考与分析

1. 指令与微指令、程序与微程序之间有什么联系？

2. 无论是程序还是微程序都必须按一定的顺序执行其中的指令或微指令，请分别说明它们确定下一条要执行的指令或微指令的方法。