# 实验七 简单模型机实验

## 一、实验目的

1. 通过总线将微程序控制器与运算器、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条机器指令组成的简单程序。
3. 掌握微指令与机器指令的关系，建立整机概念。

## 预习要求

1. 读懂实验电路图；
2. 预习实验电路图，熟悉实验元器件的功能特性和使用方法。

## 三、实验要求

1. 做好实验预习，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. 按照实验内容与步骤的要求进行实验，遇到问题冷静、独立思考，认真仔细地完成实验；
3. 完成实验报告。

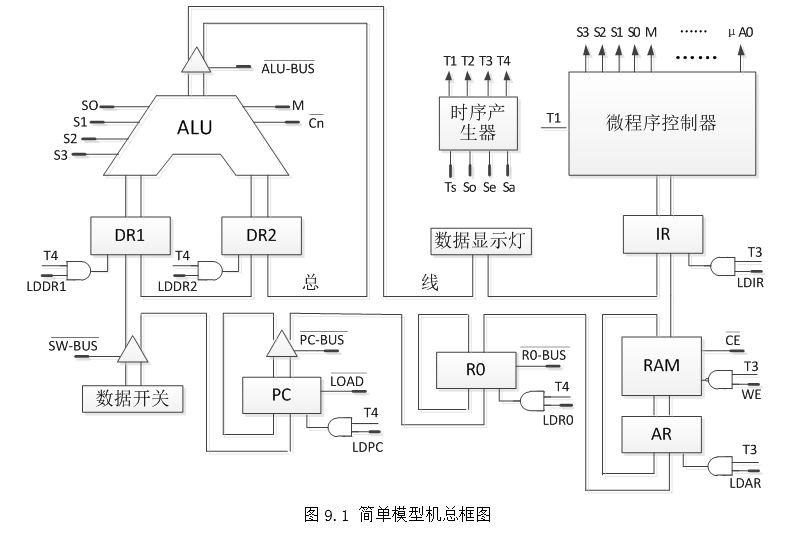
## 四、实验内容

### 实验电路图

本实验综合了实验1到实验6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9.1所示。

实验电路中的主要控制信号如下：

1. M：选择ALU的运算模式。
2. S3，S2，S1，S0：选择ALU的运算类型。
3. /Cn：向ALU最低位输入的进位信号。
4. LDDR1：DR1的数据加载信号，为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。
6. /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. /SW-BUS：开关输出三态门使能信号。
8. /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. /LOAD：PC的置数信号。
10. LDPC：PC加载信号，为1时，在T4的上升沿执行清零、置数或计数。
11. /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的上升沿将数据存入R0。
13. LDIR：IR的加载信号，为1时，在T3的上升沿将指令锁存到IR。
14. /CE：6116片选信号。
15. WE：存储器写信号。

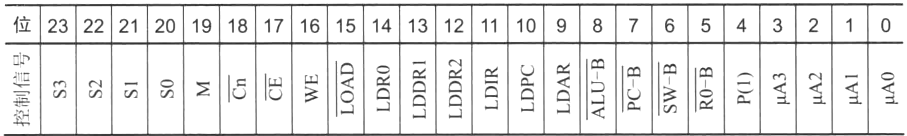
图9.1 简单模型机总框图

### 实验原理

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9-1所示。

表9-1 微指令格式



本实验使用的微程序流程设计如图9.2所示。对应的微程序代码存放中控制存储器中，如表9-2所示。

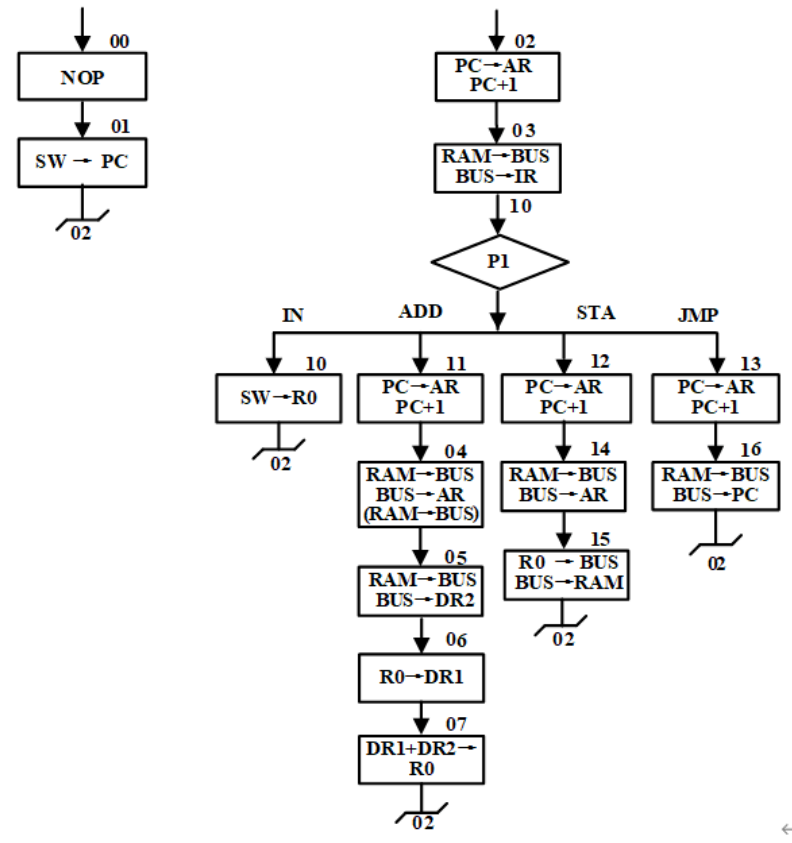
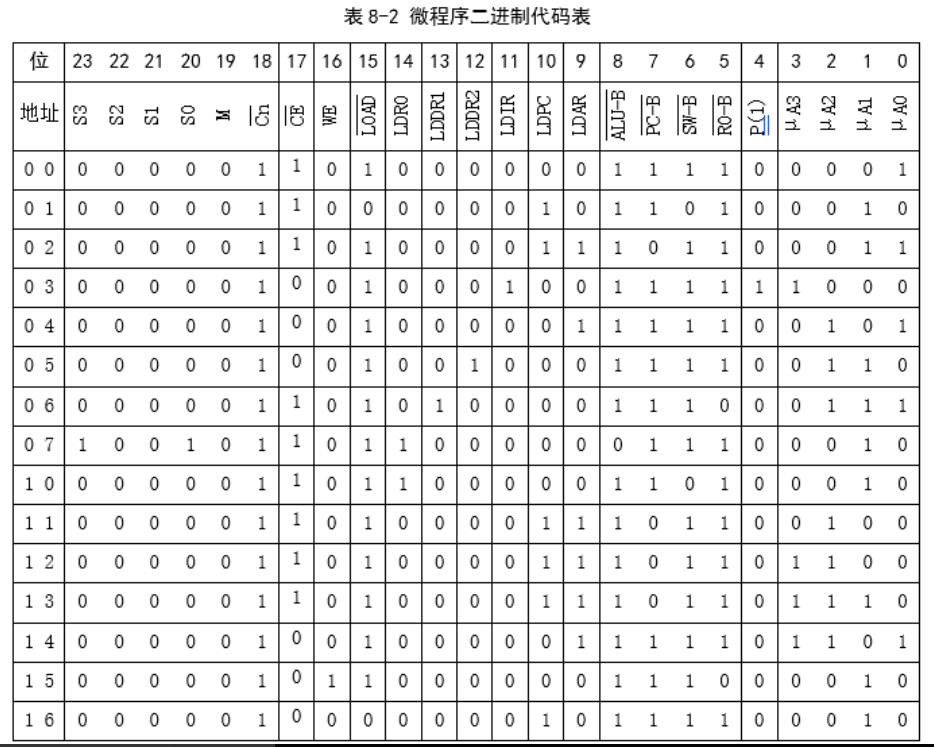


图9.2 微程序流程图

表9-2 微程序二进制代码表



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。

请根据表9-2，随机挑选两条微指令，写出实现其功能的微命令组合

|  |  |  |
| --- | --- | --- |
| 地址 | 微指令编码 | 说明 |
| 02 | 000001101000011101100011 | PC的LOAD,LDPC为1，AR的LDAR为1，PC的值通过总线将数据传送到AR中，ALU-BUS为1,SW-BUS,RO-BUS为1，PC的值加1后存入PC |
| 03 | 000001001000100111111000 | LOAD为1，LAIR为1,ALU-BUS,PC-BUS,SW-BUS,RO-BUS为1，RAM中的值通过总线传递到IR地址寄存器中 |

模型机共包含4条指令，指令格式如表9-3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中装入的程序和数据如表9-4所示。

表9-3 机器指令格式

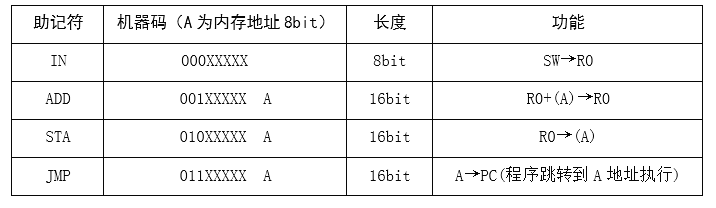


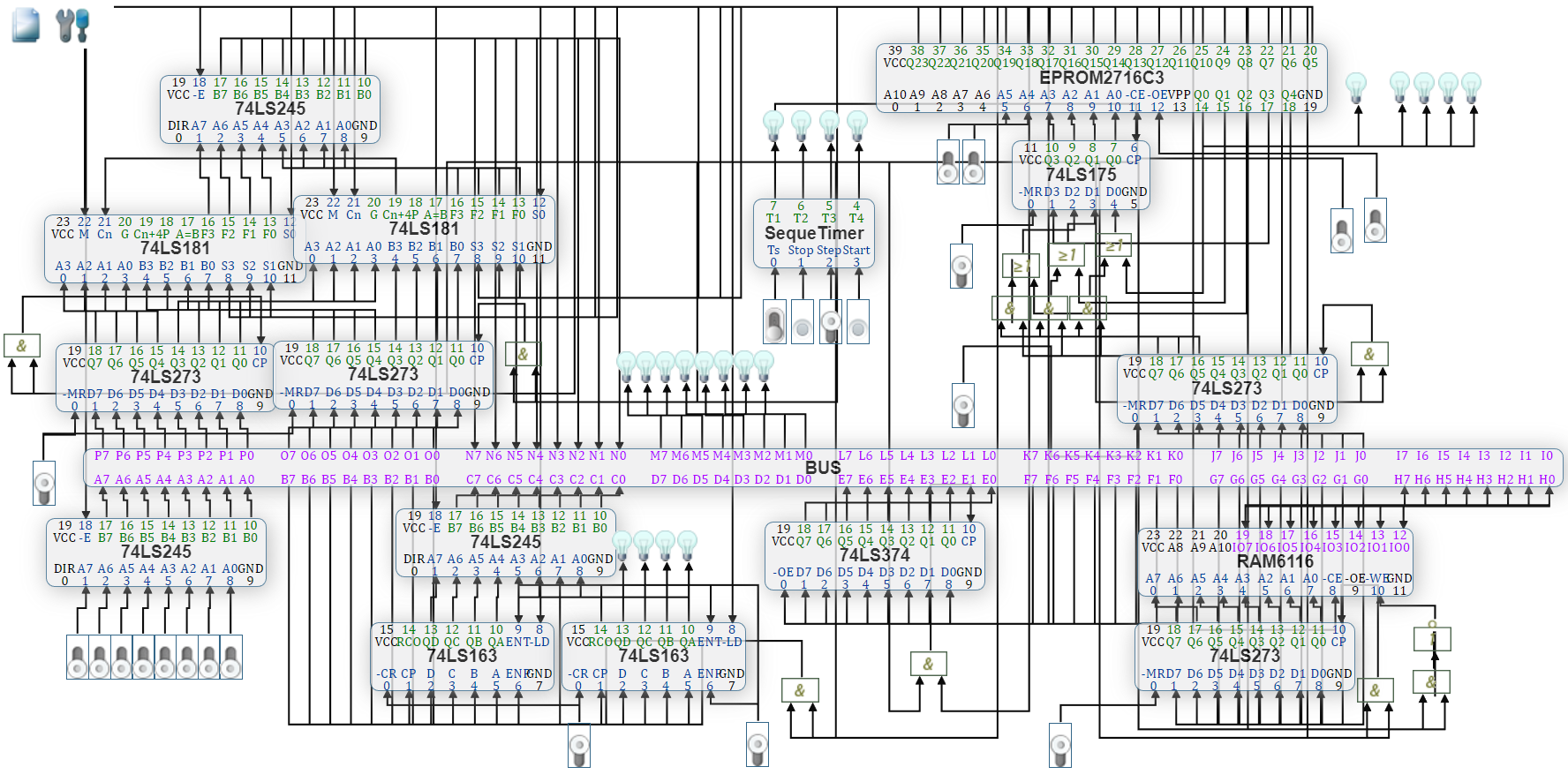
表9-4 RAM中的程序和数据



### 实验内容与步骤

1. 运行虚拟实验系统，按照图9.1绘制实验电路图：

（请在此处粘贴电路截图）



1. 电路预设值：将DR1、DR2和AR的/MR置1，将/CR、ENT、ENP置1，Step置1，将74LS175和IR的/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00H开始运行。
2. 打开电源。
3. 在数据开关SW7~SW0上设置好程序的起始地址（00000000）。
4. 点击1次时序发生器的Start，思考并回答：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？

答：执行的是第01条微指令。作用：将数据开关上的程序初始地址送入PC中。

1. 再单步执行两条微指令，思考并回答：这两条微指令的作用是什么？

答：先执行第02条微指令，将PC中程序的初始地址输入到地址寄存器AR中，并且PC加1；再执行第03条指令，将程序的第一条机器指令从RAM通过总线送入指令寄存器中。

1. 通过数据开关SW7~SW0设置操作数的值为0000 0111(学号20201210207，最后一位为7)。思考并回答：此设置是否可以提前？如果可以最早应该在以上第几步之后？

答：可以提前，最早可以提前到 步骤(4)设置程序的起始地址 之后。

1. 点击Start，执行微指令SW→R0，将操作数1保存到累加器R0中。
2. 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9-5补充完整。

表9-5 总线数据表

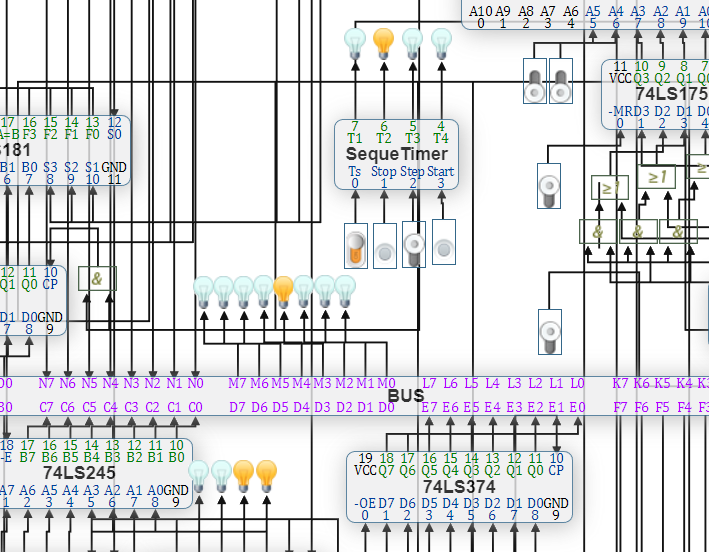
学号为20201210207，最低位为7通过开关输入，0通过修改存储器的内容读入存储器。

|  |  |  |  |
| --- | --- | --- | --- |
| 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 | 0000 0001 | 02 | (PC)→AR, PC:01 |
| 2 | 0000 0010 | 02 | PC+1, PC:02 |
| 3 | 0010 0000 | 03 | RAM→IR, BUS→IR,ADD指令 |
| 4 | 0000 0010 | 11 | PC→AR, PC:02 |
| 5 | 0000 0011 | 11 | PC+1, PC:03 |
| 6 | 0000 1000 | 04 | RAM→BUS,先显示地址，0000 1000八进制的10 |
| 7 | 0000 0000 | 04 | BUS→AR,显示0000 1000地址的数据0000 0000 |
| 8 | 0000 0000 | 05 | RAM→BUS, BUS→DR2, DR2:0000 000(0)，把数据存到DR2 |
| 9 | 0000 0111 | 06 | R0→DR1, DR1:0000 0111(7) |
| 10 | 0000 1001 | 07 | DR1+DR2→R0, R0: 0000 01111(7) |
| 11 | 0000 0011 | 02 | PC→AR, PC:03 |
| 12 | 0000 0100 | 02 | PC+1, PC:04 |
| 13 | 0100 0000 | 03 | RAM→BUS, BUS→IR, STA指令 |
| 14 | 0000 0100 | 12 | PC→AR, PC:04 |
| 15 | 0000 0101 | 12 | PC+1, PC:05 |
| 16 | 0000 1001 | 14 | RAM→BUS,BUS→AR, 取得地址八进制11地址 |
| 17 | 0000 0111 | 15 | R0→BUS, BUS→RAM, 数据0000 0111写入地址为1001的RAM |
| 18 | 0000 0101 | 02 | (PC)→AR, PC:05 |
| 19 | 0000 0110 | 02 | PC+1, PC:06 |
| 20 | 0110 0000 | 03 | RAM→BUS, BUS→IR,JMP指令 |
| 21 | 0000 0110 | 13 | (PC)→AR, PC:06 |
| 22 | 0000 0111 | 13 | PC+1, PC:07 |
| 23 | 0000 00000 | 16 | RAM→BUS,BUS→PC, PC:00 |

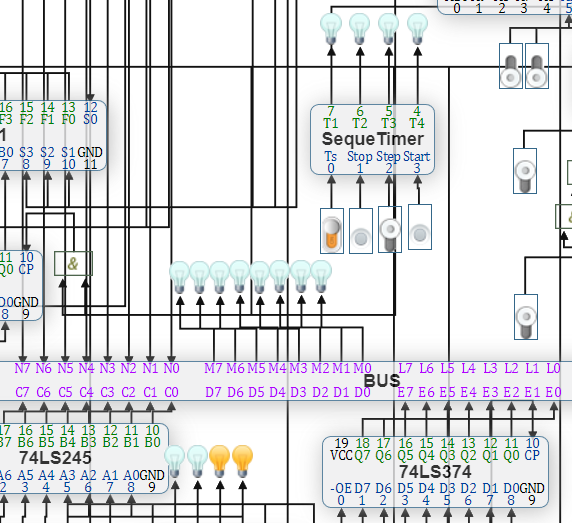
在做实验时，表格中的序号6,7， 04操作时，先显示的地址灯，最后显示的是地址对应的数据灯。

-------------

04操作时，步骤灯还没显示完，在过程中，先显示的地址灯即 0000 1000。RAM->BUS

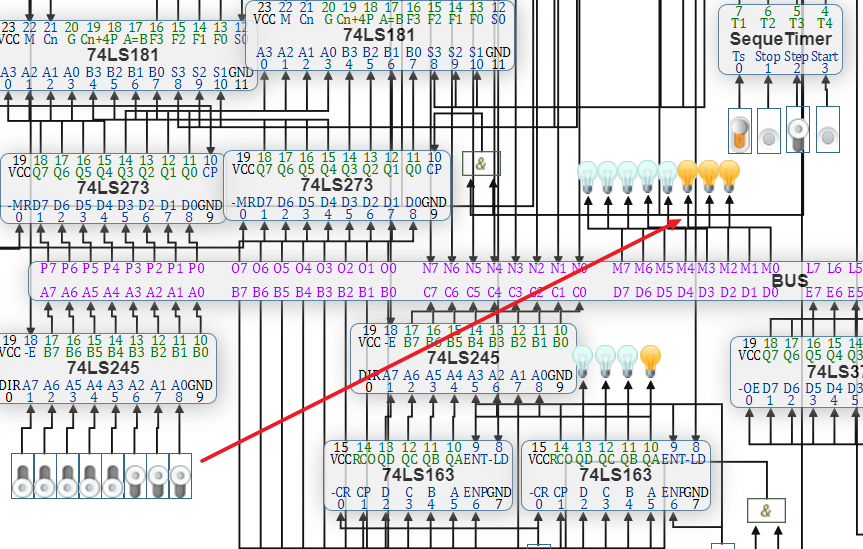


04地址的操作最后，BUS->AR，显示了0000 1000地址对应的数据(0000 0000)灯

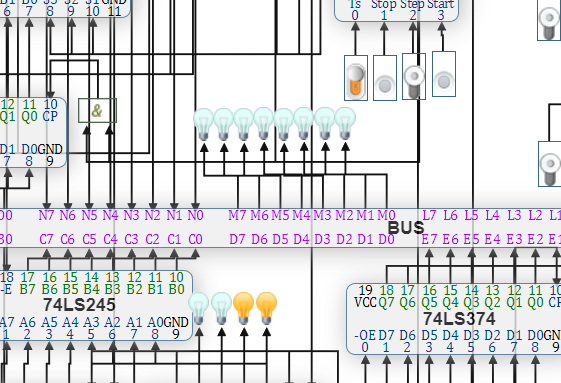


----------------

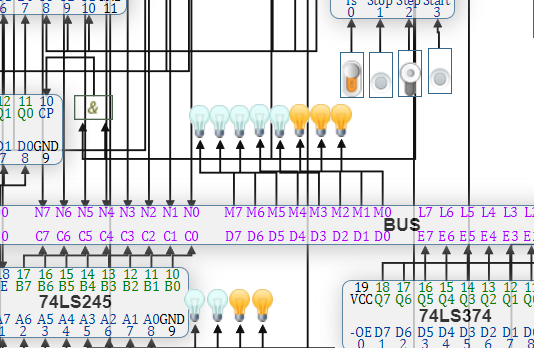
通过开关输入7, 0000 0111



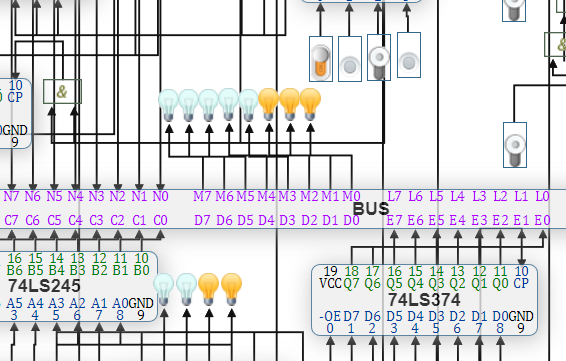
写入DR2的数据 0000 0000



写入DR1的数据0000 0111，



DR1+DR2运算后的结果，7 + 0 = 7， 0000 0111





1. 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。



### 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？

指令是最基本的计算机指令，指定了计算机要执行的操作，微指令是指令的一个子集，制定了如何执行指令。

程序是一组指令的集合，用来完成特定任务的计算机程序，微程序是程序的一部分负责控制程序的执行。

微指令和微程序都是指令和程序的一个子集，负责控制计算机执行的具体细节。指令和程序是由人类设计和编程的高级控制方式。

1. 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。

程序的起始指令地址由CPU装入PC，然后PC可以根据起始指令地址然后迭代增加得到下一条执行指令。

微指令的顺序控制字段由设计设制定好放入存储器中或者由设计者制定的判断字段控制产生后继微指令地址。