# 实验五、程序计数器实验

一、实验目的**：**

1．连接程序计数器、地址寄存器、存储器与指令寄存器，理解程序计数器的作用。

2．掌握使用微命令通过程序计数器从存储器中读取指令和数据的方法。

## 预习要求：

1．读懂实验电路图。

2．预习实验电路图，熟悉实验中所用芯片各引脚的功能和连接方法。

## 三、实验要求

1.做好实验预习，在实验之前填写好表7-4至表7-7，读懂实验电路图，熟悉实验元器件的

功能特性和使用方法

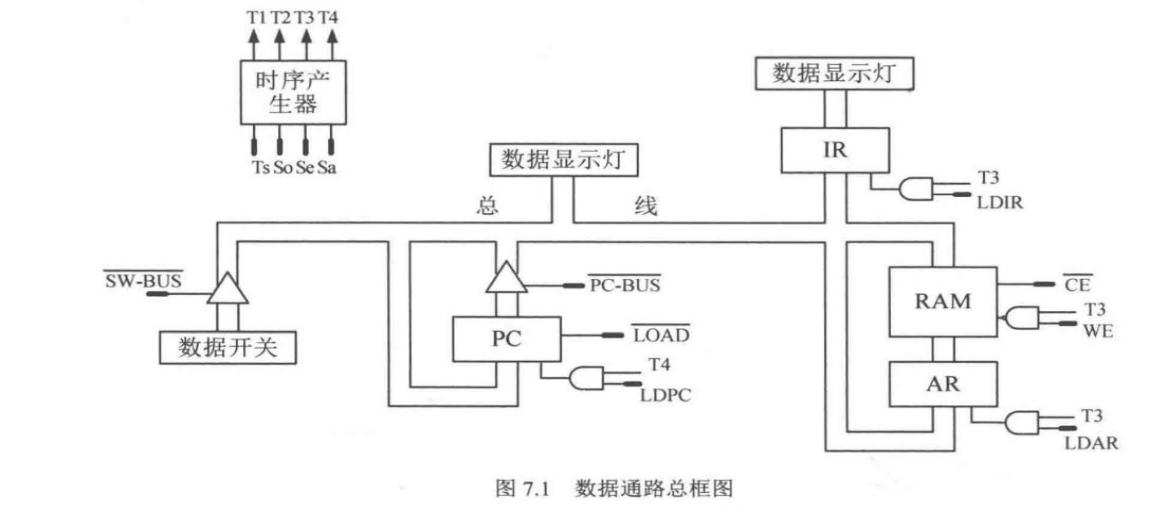
2．按照实验内容与步骤的要求进行实验，对预习时填写好的微命令进行验证与调试，遇到问

题请冷静、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



### 实验线路说明

本实验使用的主要元器件有：8 位数据锁存器 74LS273 , 4位二进制计数器74LS163 、三态输出的总线收发器74LS245，2Kx8 静态随机存储器 6116，时序发生器，与非门、与门、指示灯等．芯片详细说明请见附录。

图 7.1 为本实验数据通路总框图，其中程序计数器 PC 由 2 片 74LS163 级联构成， IR 和 AR 均为一片 74LS273 ， RAM为一片 6116 芯片，△ 表示三态门 74LS245 ， 时序发生器为虚拟实验系统提供的虚拟组件。

实验电路中涉及的主要控制信号如下：

1. LDIR：IR 的加载信号，与 T3 脉冲配合将总线上的数据打入 IR 中 。LDIR 和 T3 通过与门进行与运算之后连接到 74LS273 芯片的 CP 引脚，当 LDIR=l 时在 T3 的上升沿将指令锁存到 IR 并发送给数据显示灯。

2 .MR ：芯片 74LS273 的清零信号，低电平有效。本实验恒置为 1 。

3. CE ：6116片 选信号。为 0 时 6116 正常工作。

4. OE ：存储器读信号． CE = 0， OE = 0时为读操作，实验中将其接地，恒置为 0 。

5. WE：存储器写信号，与 T3 脉冲配合实现存储器写操作。WE 和 T3 通过与非门进行与非运算之后连接到 6116 芯片的 WE 引脚， WE 引脚低电平有效。在 CE = 0 ， 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。

6. LDAR：AR 的地址加载信号，与 T3 脉冲配合将总线上的地址打入 AR 中。LDAR 和 T3 通过与门进行与运算之后连接到 74LS273 芯片的 CP 引脚，当LDAR = 1 时在T3的上升沿将地址锁存到 AR 。

7. 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. 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. CR：PC 的清零信号，为 0 时 PC 为清零模式，本实验恒置为 1 。

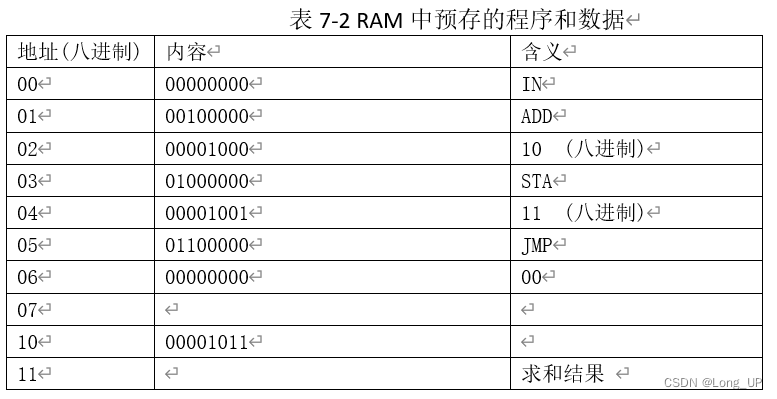
10.LOAD ：PC的置数信号，为 0 时 PC 工作在置数模式，可在此模式下为 PC 设置初始值。

11.ENT和ENP: PC的使能信号，当 LOAD＝ 1且 ENT = 1 、ENP = 1 时，PC 工作在计数模式。本实验将这两个信号恒置为 1（用于芯片级联的 ENT、ENP 引脚除外）。

12. LDPC: PC 的加载信号，与 T4 通过与门进行与运算之后连接到 74LS163 芯片的 CP 引脚，当 LDPC = 1 时，在 T4 的上升沿执行清零、置数或者计数操作。

### 实验原理

实验电路如图 7.1 所示，程序计数器、指令寄存器、地址寄存器和存储器等通过总线相连。存储器中预先存放了一小段程序和数据，程序是指令的有序集合，程序计数器用于生成下一条要执行的指令的地址。本实验用到的四条机器指令格式如表 7-1 所示，预先装入的程序和数据如表 7-2 所示．实验任务就是利用程序计数器，将RAM中的指令一一读出。

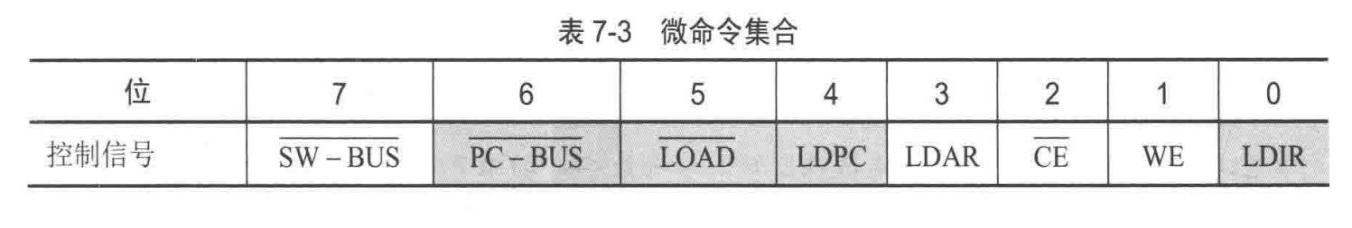


在程序开始执行前，必须将PC的值设置为程序的起始地址，即程序的一条指令所在的内存单元地址．在程序执行过程中，CPU 将自动修改 PC 的值，使其保持为下一条指令的地址。由于大多数指令都是顺序执行的，所以修改的过程通常只是简单的对 PC 加 1 。当程序转移时，转移指令实际就是将 PC 的值设置为转去的目的地址，以实现跳转。

实验电路使用两片 74LS163 芯片级联构成程序计数器，74LS163 芯片有三种工作模式，即清零模式、置数模式和计数模式，可通过相应输入引脚设置其工作模式。74LS163 的 CP 引脚接时钟脉冲信号，在脉冲信号的上升沿触发当前工作模式对应的操作。例如，若当前工作模式为清零模式，则在脉冲信号的上升沿将 PC 清零：若当前工作模式为计数模式，则在脉冲信号上升沿到来时，PC 值加 1 。

数据开关（SW7~SW0）设置的程序起始地址经三态门发送至总线。PC 从总线上接收起始地址并设置为计数初值，PC 中的值经过三态门送至总线，PC 的值递增 1 。地址寄存器 AR 从总线上获取地址并送至存储器，存储器按地址进行读操作，将读出的指令或数据发送至总线。IR 从总线上获取指令并锁存。流经总线的所有数据和地址都将在数据灯上显示。

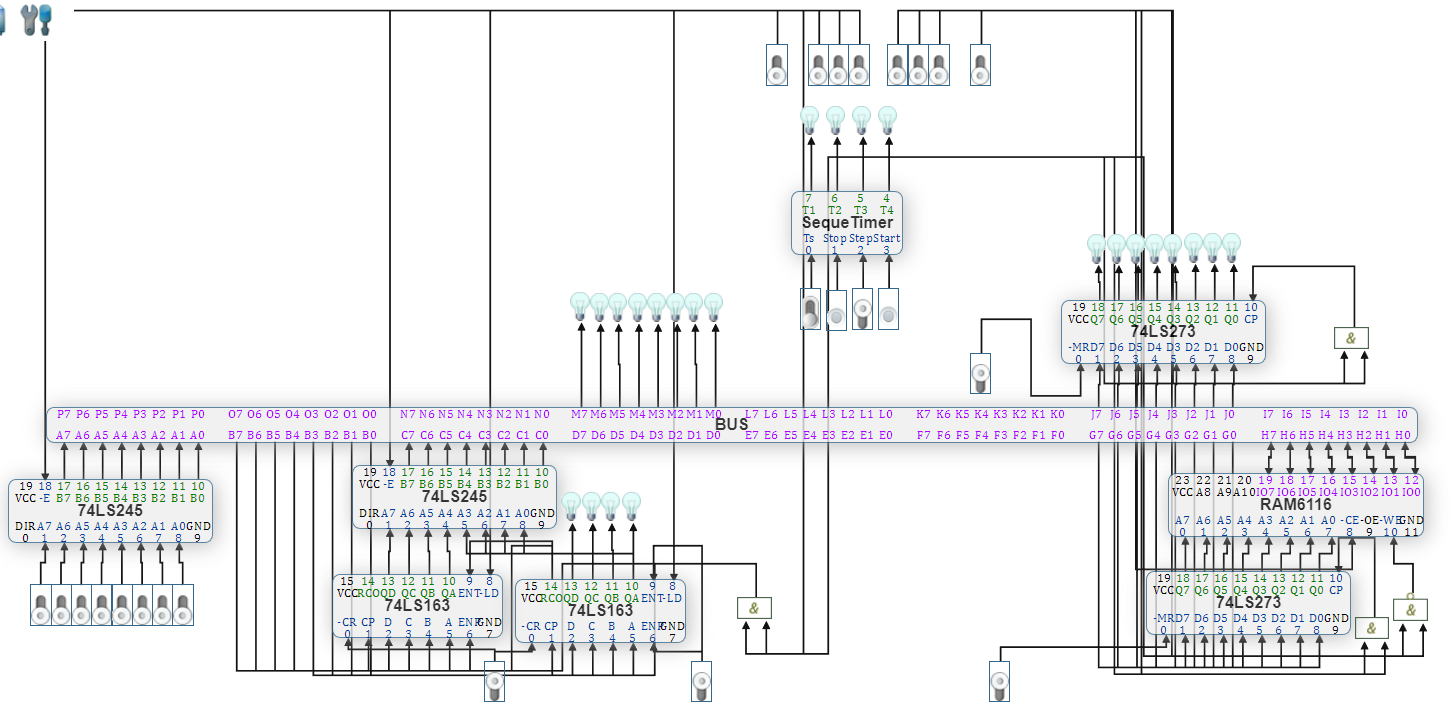
为方便实验进行．将图 7.1 中的所有控制信号归纳到表 7-3 中。



### 实验内容与步骤

1. 运行虚拟实验系统，导入实验电路图，接好表 7-3 中列出的所有控制信号线，并仔细检查一遍，确保连接正确。连接好的电路截图如下：

（请在此处粘贴电路截图）



1. 阅读表7-2的程序，并回答问题：此程序的功能是什么？

此程序的功能是：**通过数据开关输入一个数，然后和08H存储单元的数据相加，然后结果存储到09H中，循环执行这段。**

1. 电路预设置：将计数器的 CR 、ENT、ENP 置1，IR、AR的MR 置1 。时序发生器的

step 置1 。

4. 程序起始地址→PC，地址从数据开关输入。具体步骤如下：

(1）设计好要使用的微命令，**填入表7-4中．**

表7-4 程序起始地址→PC微命令

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 功能 | 微命令 | | | | | | | |
| SW-BUS | PC-BUS | LOAD | LDPC | LDAR | CE | WE | LDIR |
| 数据开关→PC | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 |

（2) 打开电源

（3) 设置控制信号1：数据开关→PC(此处填写微命令序列：01010100)；将数据开关设置为地址00H，单击时序发生器的 start 按钮，等待一个CPU 周期，此时PC 被置为 00H 。

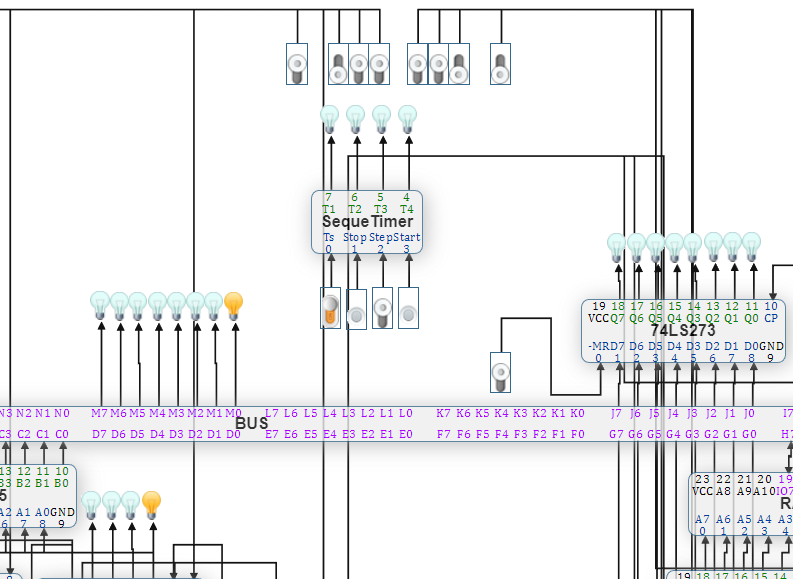
5. 取指令：以当前 PC 的值作为地址，取出存储器中相应的指令，放入指令寄存器 IR ，同时 PC+1 。具体步骤如下：

（1） 设计好要使用的微命令，**填入表 7-5 中**

表7-5 取指令微命令

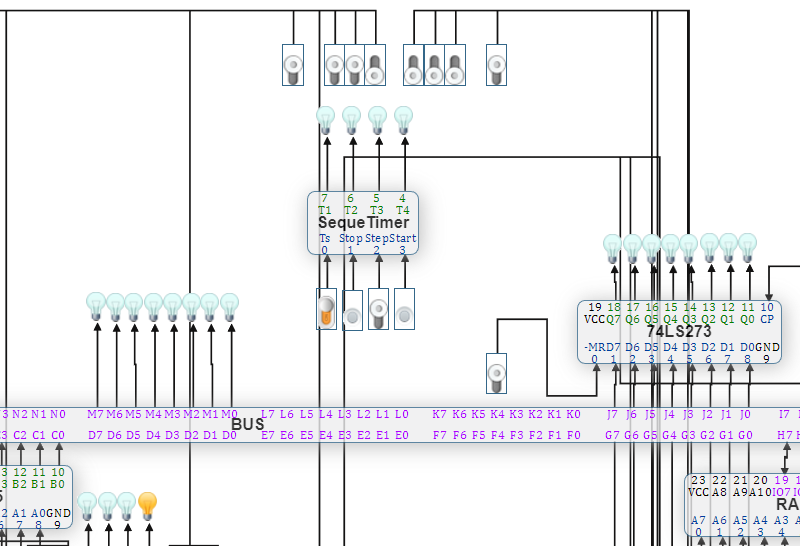
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 功能 | 微命令 | | | | | | | |
| SW-BUS | PC-BUS | LOAD | LDPC | LDAR | CE | WE | LDIR |
| PC→AR，PC+1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 |
| RAM→BUS,BUS→IR | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 |

（2）设置控制信号：PC→AR ，PC+1→(此处填写微命令序列：10111100)；单击 start 按钮。等待一个 CPU 周期，此时 PC 的值存入 AR ，而后 PC 递增 1 。

（请在此处粘贴你绘制的实验电路指示灯状态）

（3）设置控制信号：RAM→IR(此处填写微命令序列：11100001)；单击 Start 按钮，等待一个 CPU 周期，此时 00H 地址处的 IN 指令被取出放入了 IR 。注意：由于 IN 指令为全零，所以此时指示灯不会点亮。

（请在此处粘贴你绘制的实验电路指示灯状态）



6. 重复执行一次表 7-5 中的 2 组微命令，读出 PC 所指单元内容（即下一条指令 ADD 的操作码部分）到 IR 。

注意：多次重复执行表 7-5 中的 2 组微命令，可以把后面存储单元内容依次读入 IR 中，但是，程序执行时只需要把指令的操作码读入 IR ，指令的操作数或操作数地址不需要读入 IR ，而应送至其他寄存器。

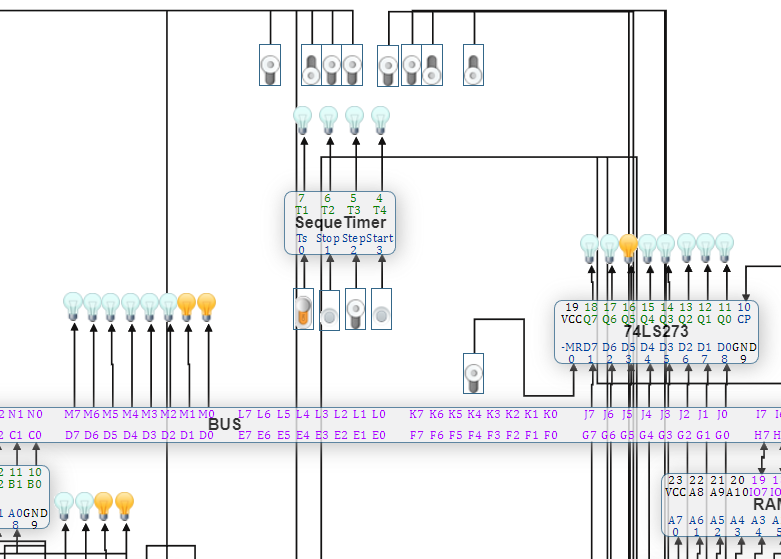
7. 读 ADD 指令的操作数地址，在 LED 上显示，同时 PC+1 。注意：上一步取 ADD 指令后，PC 的值增加了 1 ，当前 PC 的值为 02H ，指向 ADD 指令操作数的地址，即 PC 的值是操作数地址的地址。具体步骤如下：

（1）设计好要使用的微命令，**填入表 7-6 中。**

表7-6读ADD指令的操作数地址地址微指令

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 功能 | 微命令 | | | | | | | |
| SW-BUS | PC-BUS | LOAD | LDPC | LDAR | CE | WE | LDIR |
| PC→AR,PC+1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 |
| RAM→BUS | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |

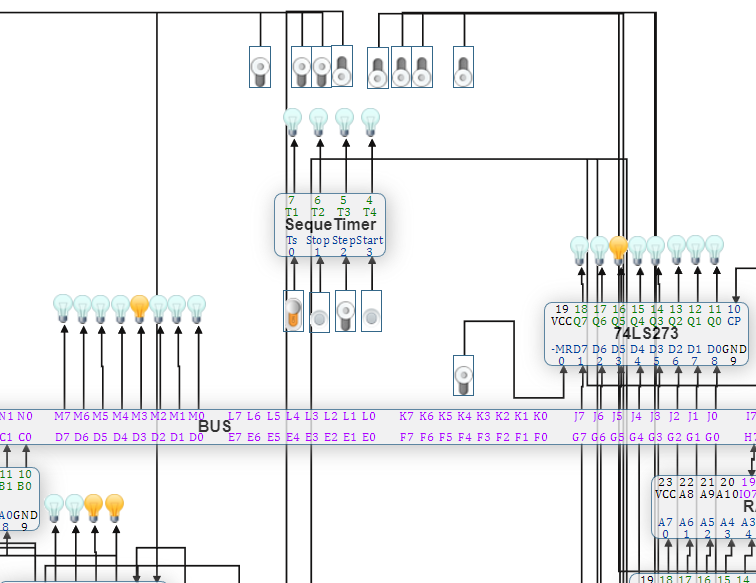
1. 设置控制信号：PC→AR，PC+1(此处填写微命令序列：10111100) 单击Start按钮．等待一个 CPU 周期。



（3）设置控制信号：RAM→BUS(此处填写微命令序列：11100000)；

说明：本实验只关注取指令阶段， 此处只读出了操作数地址，没有从指定地址中读出操作数，也没有真正执行加法操作，下同。

（请在此处粘贴你绘制的实验电路指示灯状态）



8. 用同样的方法取 STA 指令到 IR ， PC+1：读出 STA 指令的操作数地址，PC+1 。

9. 取JMP指令并执行，步骤如下：

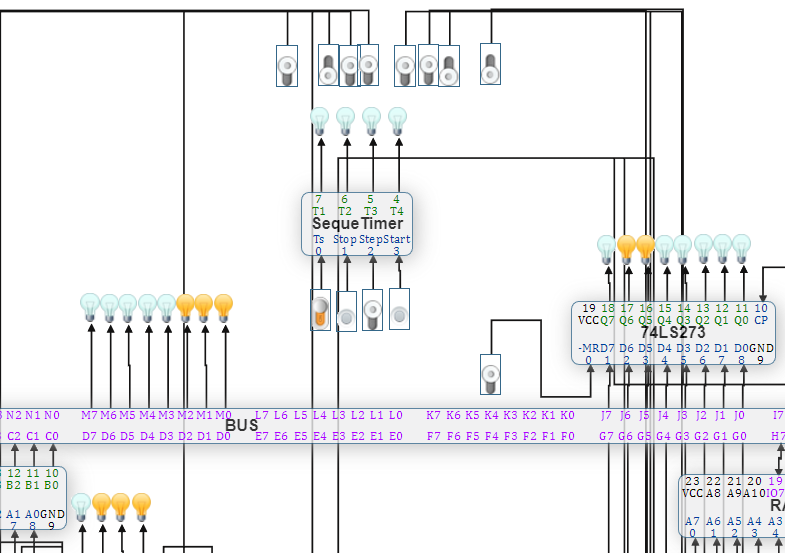
（1）取指令的微命令上面已经有了，需要补充执行跳转的微命令组合，即：地址改写当前 PC 值的微命令，**请设计微命令，填入表 7-7 中。**

表7-7取JMP指令并执行微命令

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 功能 | 微命令 | | | | | | | |
| SW-BUS | PC-BUS | LOAD | LDPC | LDAR | CE | WE | LDIR |
| PC→AR,PC+1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 |
| RAM→BUS,BUS→PC | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |

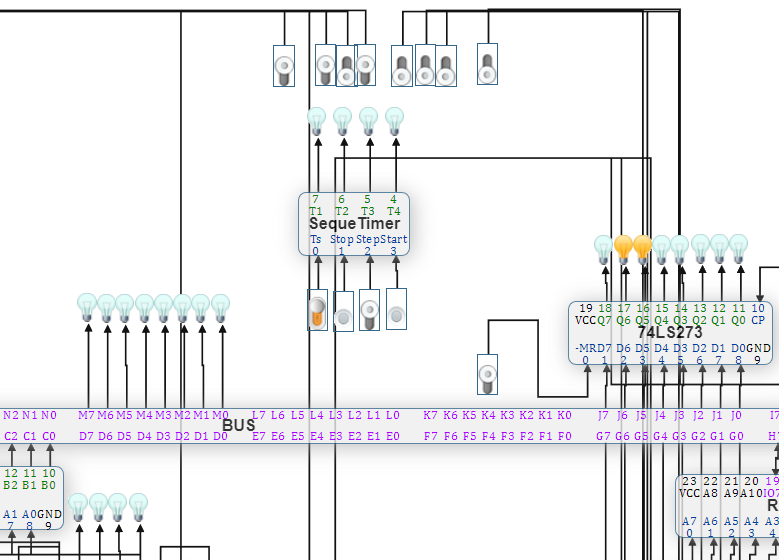
（2）用前面的方法取JMP指令到IR，PC+1。

（3）设置控制信号：PC→AR,PC+1 (此处填写微命令序列：10111100) 单击Start按钮。等待一个 CPU 周期。



（4）设置控制信号：RAM→BUS，BUS→PC (此处填写微命令序列：11010000), 单击 Start 按钮。等待一个 CPU 周期。 PC 值已被设为 00H 。

（请在此处粘贴你绘制的实验电路指示灯状态）

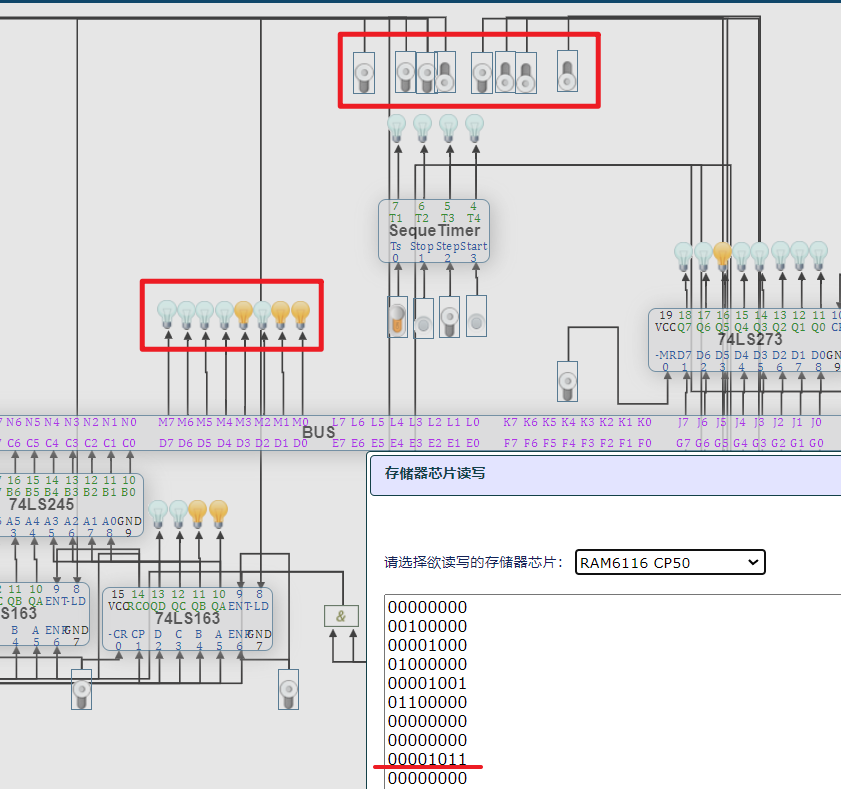


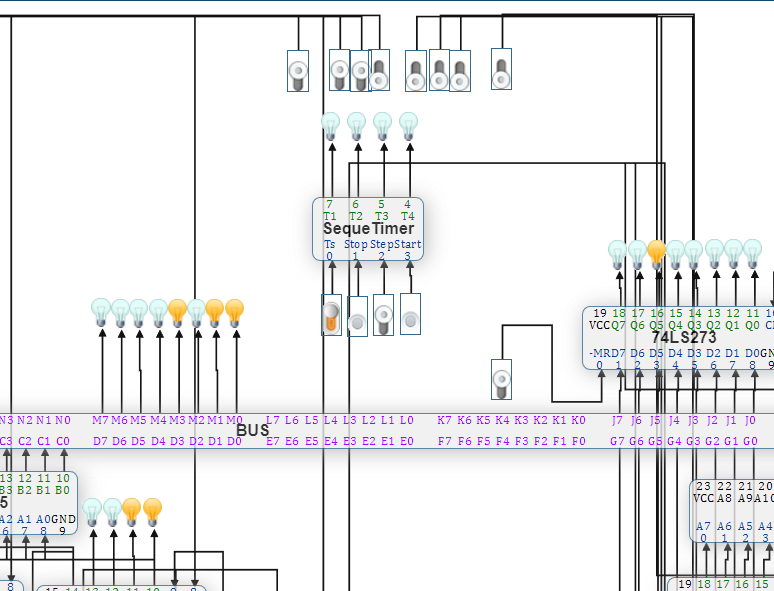
10. 尝试在读ADD指令的操作数地址时，不仅仅读出操作数地址，也把真正的操作数从存储器读出来，并在 LED 上显示。

（请在此处粘贴你绘制的实验电路指示灯状态）

此时ADD操作数的地址存到了AR中，通过11101000指令，将ADD操作数地址RAM->BUS, BUS->AR，就可以将操作数地址传入到AR中。

通过11100000, RAM->BUS,就可以在总线上显示出操作数的真实值。





### 思考与分析

1. 计算机开机的时候，一条指令都没有执行之前，程序计数器PC的值是如何设置的？

刚开机时，PC的值应该设置为第一条指令的地址，CPU通过访问基本输入输出系统BIOS-ROM获取第一条指令定制。

1. 本实验中，程序计数器是否一直指向下一条要执行的指令？

不是，在执行完ADD操作后，PC指向的是ADD操作数的地址，而不是下一条要执行的指令。

1. 程序计数器与微程序有什么联系？

微程序由若干为微指令组成，而程序计数器的变化（程序计数器输出，然后变为下一个要进行的指令）是通过执行微指令实现的。