河北工业大学

人工智能与数据科学学院

实验报告

**课程名称：** 计算机组成原理课程设计

**学 院：** 人工智能与数据科学学院

**系（专业）：** 计算机科学与技术

**班 级：** 计221

**学 号：** 225396 225392

**学生姓名：** 韩宇轩 邸相耀

2024 年 6月 19 日

**目录**

[1 启停、时序电路实验 1](#_Toc170289389)

[2 运算器数据通路实验 5](#_Toc170289390)

[3 总线、半导体静态存贮器实验 11](#_Toc170289391)

[4 模型计算机整机实验 17](#_Toc170289392)

### 1 启停、时序电路实验

**1.1 实验目的**

1.1.1 加深理解启停电路的功能，理解简单启停电路的工作原理和它的不足，深入理解改进后的启停电路的工作原理。理解改进后的启停电路对脉冲信号进行的有效控制。

1.1.2 按对启停电路的要求设计电路，并通过分频，组合所需的时序信号。

**1.2 实验原理**

1.2.1 启停电路需确保计算机工作从规定的第一个脉冲开始到最后一个脉冲结束。

1.2.2 使用RS触发器和维持阻塞原理来实现启停电路，确保脉冲完整。

1.2.3 启停电路通过触发器Q端变化控制脉冲输出，停机时置Q为0。

1.2.4 节拍脉冲电路利用分频计数器和停机信号设计，以适应不同工作模式。

1.2.5 防抖动电路采用双稳态触发器和单稳电路，确保操作准确性。

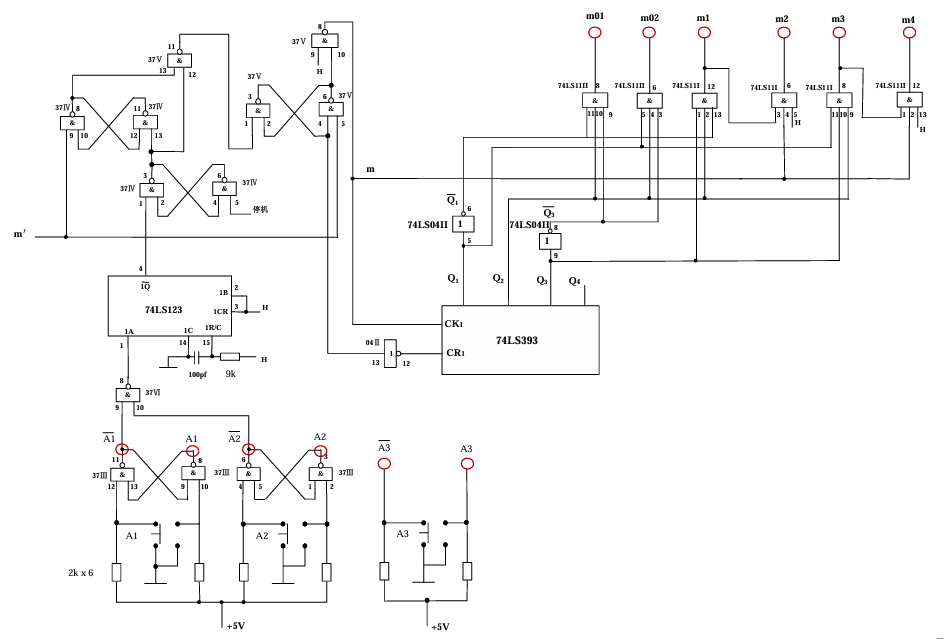
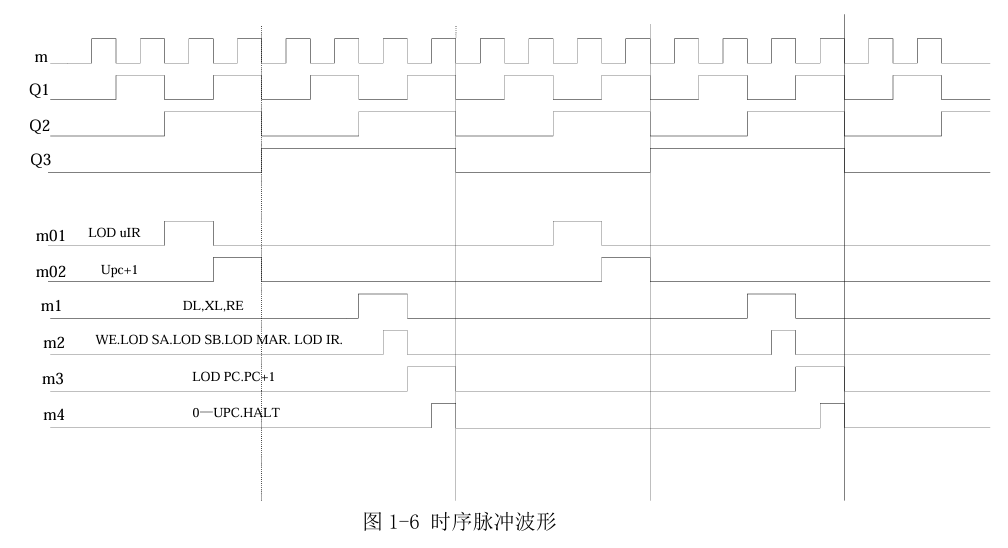
**1.3 实验内容**

1.3.1 设计启停电路，要求能实现手动停机和单周期自动停机。

1.3.2 节拍脉冲m经计数器分频后，用组合逻辑方法设计出能够产生如图1-6所示波形的时序脉冲电路。

1.3.3 画出时序电路的分频脉冲波形图，写出各工作脉冲的逻辑表达式；画出启停电路和时序电路的逻辑图，元件排列图并组装成电路。对分频计数器产生的分频信号Q1～Q3进行逻辑组合，产生一条微指令的微指令周期信号m01，m02，m1-m4,如图1-6所示。图中的时间顺序说明了一条微指令的执行过程；

1.3.4 m01从控制存储器取出一条并行微指令送入三个微指令寄存器uIR74LS273，发出执行这条微指令的23个微命令；m02使微地址指针μpc+1，为下一条微指令的送出做准备；经过2个m周期的延时，发出执行这条微指令的m1、m2、m3、m4脉冲。m1～m4的时序和作用如图1-6所示。节拍脉冲形成电路见图1-7所示。





**1.4 实验设备与器材**

仪器：计算机实验台和+5V直流稳压电源各一台。

**1.5 实验步骤**

1.5.1 设计启停电路

设计并实现一个能够手动停机和单周期自动停机的启停电路。

1.5.2 设计时序脉冲电路

利用计数器对节拍脉冲进行分频，并通过组合逻辑方法设计出能产生所需时序脉冲的电路，以满足实验要求的波形输出。

1.5.3 画出波形图与逻辑图

画出时序电路的分频脉冲波形图，并写出各工作脉冲的逻辑表达式；然后画出启停电路和时序电路的逻辑图、元件排列图，并组装成实际电路。

1.5.4 测试与验证

在搭建好的电路上进行测试，记录实验实测波形、启停情况，并对出现的现象进行分析。

**1.6 实验总结**

1.6.1 实验概述

本次实验为“启停、时序电路实验”，主要目的是加深对启停电路功能的理解，设计并改进启停电路，并通过分频和组合逻辑产生所需的时序信号。实验要求通过手动和自动两种方式实现机器的启停，以及设计能够产生特定时序波形的电路。

1.6.2 实验过程与遇到的困难

在实验过程中，我们首先遇到了设计启停电路的挑战。简单的启停电路存在脉冲不完整的问题，这会导致时序信号的不稳定。为了解决这个问题，我们采用了维持阻塞原理，并增加了几组与非门来改进电路，确保所有的脉冲信号波形完整。

此外，在设计时序电路时，我们也需要通过分频计数器产生符合要求的时序信号。实验中要求深入理解分频计数器的工作原理，并灵活应用组合逻辑设计方法。

在搭建电路时，还遇到了线路连接错误和元件选择不当的问题。这些问题导致实验初期电路无法正常工作，但通过仔细检查电路图和元件功能，最终找到了问题所在，并进行了修正。

1.6.3 解决方案

针对启停电路的问题，采用维持阻塞原理进行改进，确保脉冲信号的完整性。在时序电路的设计中，通过对分频计数器输出信号的逻辑组合，实现了所需时序信号的产生。针对实验过程中的问题，采取了以下解决方案：

1）仔细检查电路图：确保每个元件的连接都准确无误。

2）理解元件功能：深入理解元件的工作原理和特性，选择合适的元件进行电路搭建。

3）逐步调试：分步骤进行电路搭建和调试，逐步排查问题所在。

1.6.4 学到的理论知识与心得体会

通过这次实验，深入理解了启停电路的工作原理和时序电路的设计方法。实验过程中遇到了许多挑战，但通过不断学习和探索，逐渐掌握了相关知识，并成功完成了实验任务。

在实验中体会到了理论与实践相结合的重要性。理论知识是实验的基础，而实验则是检验理论知识的重要途径。通过亲手搭建电路并观察实验结果，更加深入地理解了相关理论知识的应用。

此外，还学会了如何面对和解决实验中的问题。在实验中遇到问题并不可怕，关键是要保持冷静，通过仔细分析和不断探索找到解决问题的方法。

总之，这次实验不仅学到了宝贵的理论知识，还锻炼了实践能力和解决问题的能力。

### 2 运算器数据通路实验

**2.1 实验目的**

2.1.1 熟悉74LS181函数功能发生器，提高应用器件在系统中应用的能力。

2.1.2 熟悉运算器的数据传送通路。

2.1.3 完成几种算术逻辑运算操作，加深对运算器工作原理的理解。

**2.2 实验原理**

2.2.1 74LS181功能：作为四位函数功能发生器（ALU），能完成32种算术逻辑运算，通过控制参数和模式选择算术或逻辑运算。

2.2.2 运算器数据通路：以ALU（74LS181）为核心，包括寄存器、移位门、数据总线等。操作数从总线送入ALU，运算结果经移位门送回总线。

2.2.3 移位功能：右移操作在ALU和总线间接入右移控制门，左移则通过ALU的特定指令模式实现。

2.2.4 电路设计：在ALU与总线间添加右移控制门，总线上连接必要设备。根据ALU输出判断信息，并设置指令寄存器存放控制信息。

**2.3 实验内容**

2.3.1 按运算器数据通路图设计一个能完成表所列出的八种补码运算指令的运算器。

表 运算器的 8 种指令功能表

|  |  |  |  |
| --- | --- | --- | --- |
| 指令 | 助记符 | 代码 | 功能 |
| 加 | ADD | 1001 | (Sa)+(Sb)→Sa |
| 减 | SUB | 0110 | (Sa)-(Sb)→Sa |
| 加 1 | INR | 0000 | (Sa)+1→Sa |
| 减 1 | DCR | 1111 | (Sa)-1→Sa |
| 逻辑与 | ANA | 1011 | (Sa)∧(Sb)→Sa |
| 取反 | CMA | 0101 | (Sb)→Sa |
| 右移 | SHR | 1010 | (Sb)R →Sa |
| 左移 | SHL | 1100 | (Sa)L →Sa |

2.3.2 完成元件选择，依据详细实验电路逻辑图，进行连线组装成电路。

2.3.3 在电路上进行表八种指令的操作(操作数、指令码由数据开关输入)，观察运算器数据通路上显示灯的运行结果。

**2.4 实验设备与器材**

2.4.1 计算机组成原理实验系统 1 台

2.4.2 实验使用的部件：

1）运算器（ALU）模块

2）控制器模块（CU）中的 74LS273(IR)，74LS244

3）数据输入开关 D7-D0

4）控制开关 K7、K6、K5、K0

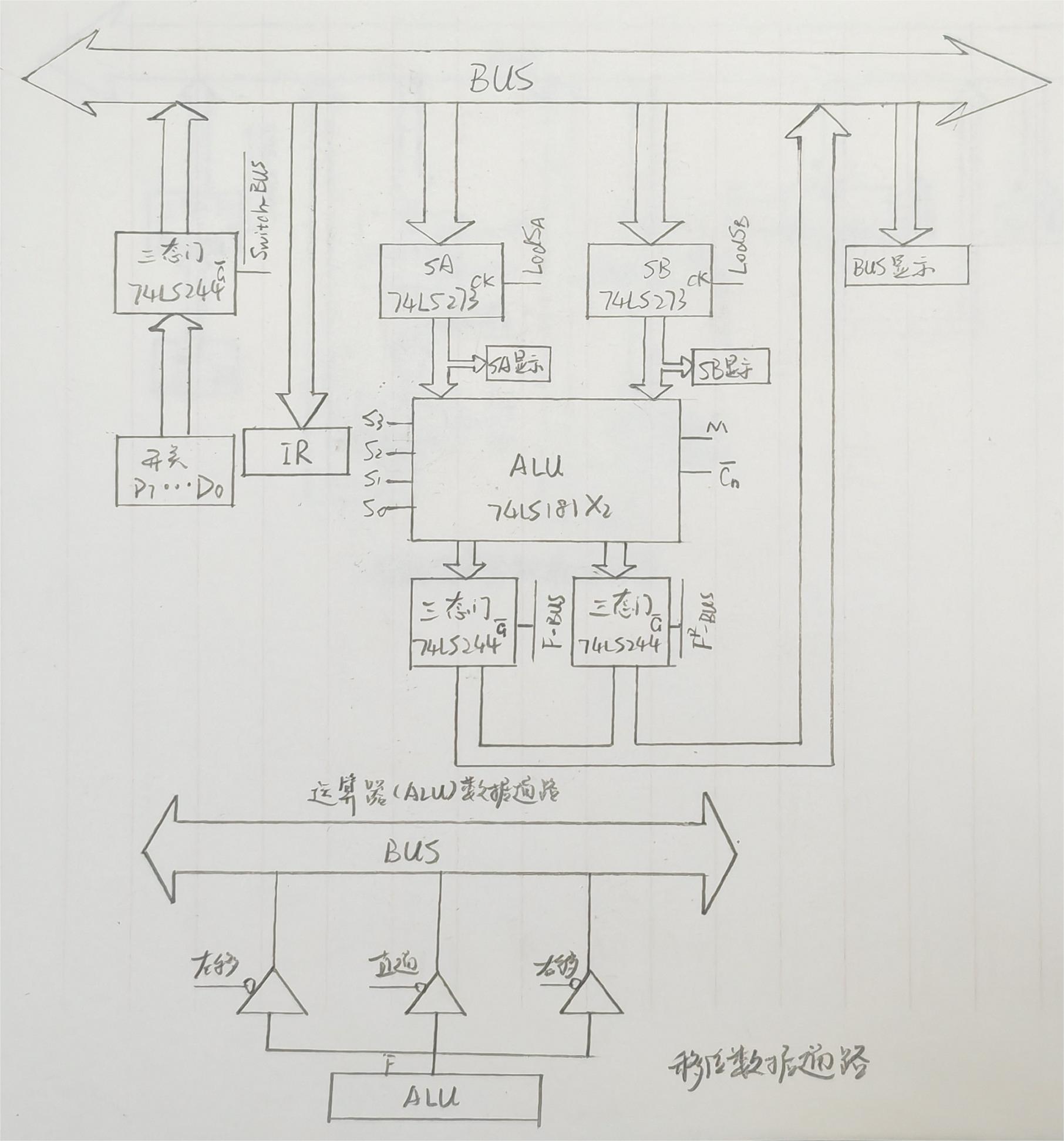
5）A3、A2、A1 按键。

2.4.3 5V 稳压电源一个

2.4.4 插接导线 15 根

**2.5 实验步骤**

2.5.1 数据通路图：



2.5.2 各开关功能：

|  |  |
| --- | --- |
| Button A1 | BUS -> SA |
| Button A2 | BUS -> SB |
| Button A3 | Clear SA & SB |
| K7 | DATA ->Bus |
| K6 | Lod IR (Cmd -> IR) |
| K5 | ALU -> Bus |
| K4 | ALU R-> Bus |
| K0 | Clear IR |
| IR7-IR2 | S3-S0, M, CN# |

2.5.3 指令周期（标蓝为输入部分，非指令周期）：

最初确保K7 = 1，K6 = 0，K5 = 1，K4 = 1，K0↓↑。

|  |  |
| --- | --- |
| Input Cmd | D7~D2输入指令 |
| Cmd –> IR | K7 = 0，K6↓↑，K7 = 1 |
| Input Data | D7~D0 输入数据 |
| Data -> SA | K7 = 0，Push A1，K7 = 1 |
| Input Data | D7~D0 输入数据 |
| Data -> SB | K7 = 0，Push A2，K7 = 1 |
| Output | K5 = 0，Push A1，K5 = 1 |

2.5.4 各操作指令：

|  |  |
| --- | --- |
| S3-S0, M, CN# | 算术 |
| 100101 | 加法 |
| 011000 | 减法 |
| 111101 | SA-1 |
| 000000 | SA+1 |
| 101100 | AND |
| 010111 | SB取反 |
| 101010 | =B（使能实验箱左三态门） |
| 101001 | =A |
| 110000 | SA左移一位 |
| 101010 | SB右移一位（使能实验箱右三态门） |

2.5.5 加法、减法、AND执行过程：

|  |  |
| --- | --- |
| Input Cmd | D7~D2输入指令 |
| Cmd –> IR | K7 = 0，K6↓↑，K7 = 1 |
| Input Data | D7~D0 输入数据 |
| Data -> SA | K7 = 0，Push A1，K7 = 1 |
| Input Data | D7~D0 输入数据 |
| Data -> SB | K7 = 0，Push A2，K7 = 1 |
| Output | K5 = 0，Push A1，K5 = 1 |

2.5.6 自增1、自减1、SA左移一位、BUS==SA操作执行过程：

|  |  |
| --- | --- |
| Input Cmd | D7~D2输入指令 |
| Cmd –> IR | K7 = 0，K6↓↑，K7 = 1 |
| Input Data | D7~D0 输入数据 |
| Data -> SA | K7 = 0，Push A1，K7 = 1 |
| Output | K5 = 0，Push A1，K5 = 1 |

2.5.7 SB取反、BUS==SB操作过程：

|  |  |
| --- | --- |
| Input Cmd | D7~D2输入指令 |
| Cmd –> IR | K7 = 0，K6↓↑，K7 = 1 |
| Input Data | D7~D0 输入数据 |
| Data -> SB | K7 = 0，Push A2，K7 = 1 |
| Output | K5 = 0，Push A1，K5 = 1 |

2.5.8 SB右移一位执行过程：

|  |  |
| --- | --- |
| Input Cmd | D7~D2输入指令 |
| Cmd –> IR | K7 = 0，K6↓↑，K7 = 1 |
| Input Data | D7~D0 输入数据 |
| Data -> SB | K7 = 0，Push A2，K7 = 1 |
| Output | K4 = 0，Push A1，K4 = 1 |

**2.6 实验总结**

2.6.1 实验概述

本次“运算器数据通路实验”的核心目的是通过实际电路的搭建，让我们更深入地理解74LS181函数功能发生器（ALU）的功能及其在实际应用中的作用。实验不仅加深了对运算器数据传送通路的理解，还涉及了几种基本的算术逻辑运算操作。整个实验流程包括了对实验目的与原理的学习、电路的设计与构建、实验操作的执行以及对实验结果的深入分析。

2.6.2 实验过程及遇到的困难与解决方案

在实验电路的搭建阶段，我们依据详细的实验电路图选择了适当的元件，并进行了仔细的线路规划。然而，实际操作中还是遇到了一些挑战：

遇到的困难：

1）复杂的线路连接导致出错率增加。

2）多元件引脚造成的混淆。

3）实验中频繁的开关和按键操作增加了操作难度。

解决方案：

1）在开始连线之前，对照电路图进行多次核查，确保每条线路的正确性。

2）针对频繁的操作，制定了一套合理的操作流程，并在实验中保持高度的专注和细心。

2.6.3 实验中学到的理论知识

通过这次实验，我们对运算器的工作原理和数据传送通路的构成有了更为深刻的理解。特别是对74LS181函数功能发生器（ALU）的应用和功能有了直观的体验。此外，我们也学会了如何根据实验的具体要求来选择合适的电子元件和构建有效的电路，以及如何通过实验来验证和应用所学的理论知识。

2.6.4 心得体会

这次实验让我们深刻体验到了实践操作在理论学习中的不可或缺的作用。亲手搭建电路并执行实验操作，使我们对运算器的工作原理和数据传送通路有了更为深刻的认识。实验中遇到的挑战也让我明白了耐心和细心在科研工作中的重要性。

我们更加确信，理论知识与实践操作是相辅相成的。仅有理论知识是远远不够的，通过实践操作我们才能真正地掌握和应用这些知识。这次实验不仅提高了我们的动手能力，还锻炼了我们解决问题的能力，为我们未来的学习和工作奠定了坚实的基础。

### 3 总线、半导体静态存贮器实验

3.1 实验目的

3.1.1 熟悉挂总线的逻辑器件的特性和总线传送的逻辑实现方法。

3.1.2 掌握半导体静态存储器的存取方法。

3.2 实验原理

3.2.1 总线传送概念

总线是计算机内部信息传输核心，用三态门等器件实现。

3.2.2 总线建立原则

确保互斥性和一致性，即同一时间仅一个发送源，且总线上器件类型一致。

3.2.3 静态存储器实验电路

采用单总线结构，通过控制时序区分地址和数据。存储器由存储体、地址译码器等组成。

3.2.4 关键器件

包括三态门器件74LS244、D触发器74LS273和计数器74LS161。

3.3 实验内容

3.3.1 总线传送实验

1）熟悉挂总线的逻辑器件特性，如三态门器件（如74LS244）和锁存器（如74LS373）在总线传送中的应用。

2）掌握建立总线的原则，包括互斥性和一致性。

3）通过实验操作，理解如何通过控制操作时序来区分送入总线的是地址还是数据。

3.3.2 半导体静态存储器实验

1）构建基于单总线结构的静态存储器实验电路，包括地址译码器、读写电路和控制电路等关键组件。

2）使用2K×8位的静态存储器（如HM6116）进行数据的读写操作。

3）学习如何通过操作开关和按键，将地址和数据通过三态传输门（如74LS244）挂到总线上，然后传输到相应的计数器、地址寄存器或存储器单元中。

3.4 实验设备与器材

3.4.1 计算机组成原理实验系统 1 台

3.4.2 实验使用的部件：

1）存储器（MEM）模块

2）辅助电路模块（AUX）中的 74LS244

3）控制模块（CU）中的 74LS273(IR)

4）数据输入开关 D7-D0

5）控制开关 K7-K0

6）A3-A0 按键。

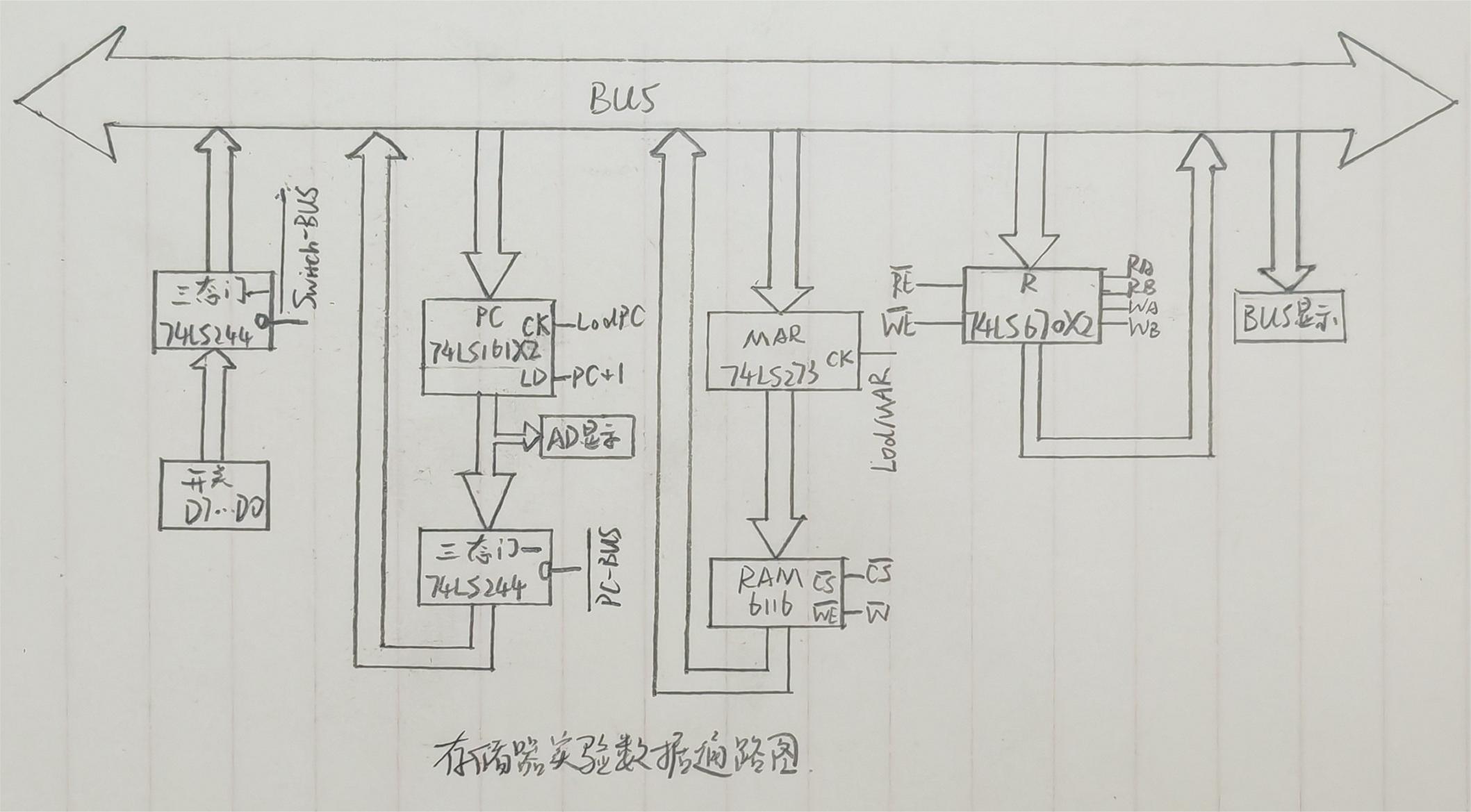
3.4.3 5V 稳压电源一个

3.4.4 插接导线17根

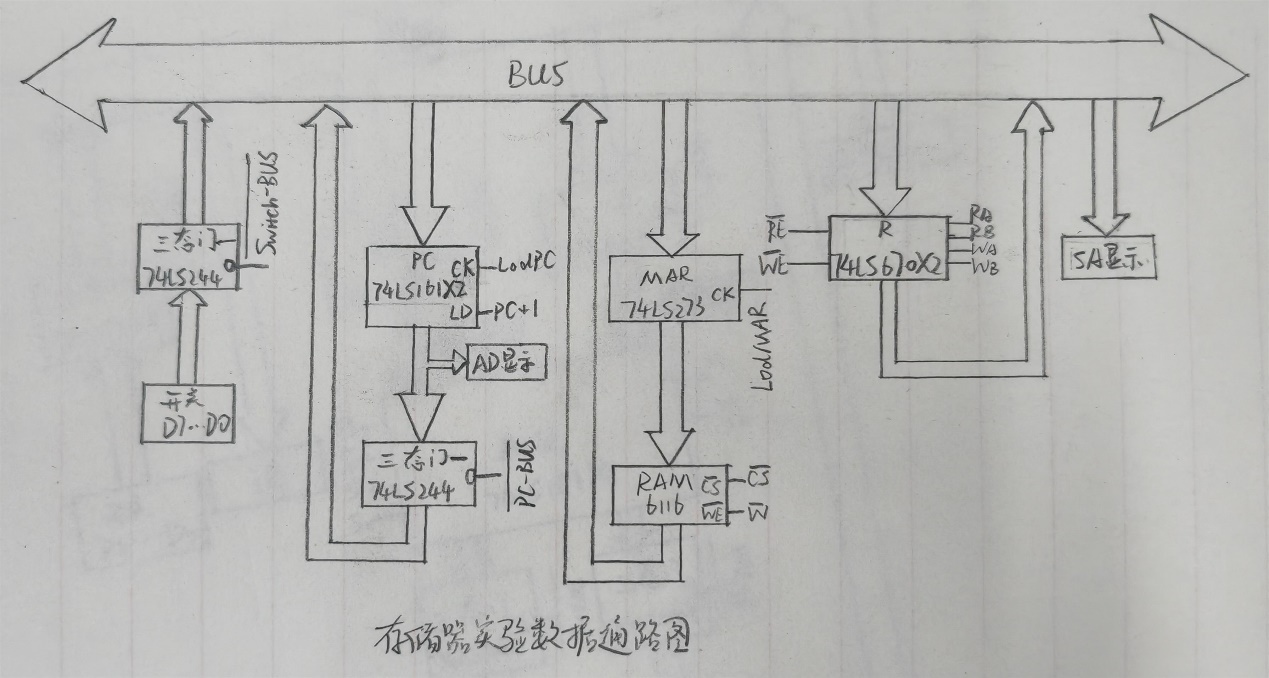
3.5 实验步骤

3.5.1 数据通路图：

基础：



拓展：



3.5.2 各开关功能：

|  |  |
| --- | --- |
| Button A1 | BUS -> PC |
| Button A2 | BUS -> MAR（存储器地址寄存器） |
| Button A3 | Clear PC & GR |
| K7 | DATA ->Bus |
| K6 | PC Mode（0=置位，1=计数） |
| K5 | PC –> BUS |
| K4 | （6116片选信号） |
| K3 | （6116的读写，数据流向控制线，0=写入，1=上总线） |
| K2 | （向GR中写） |
| K1 | （读GR中值） |
| K0 | LodIR |
| IR3-IR0 | WA, WB, RA, RB |

3.5.3 指令周期（标蓝为输入部分，非指令周期）：

最初确保：K7 = 1，K6 = 0，K5 = 1，K4 = 1，K3 = 1，K2 = 1，K1 = 1，K0 = 0。

上电时：按A3（清空PC和GR）

3.5.4 写入存储器：

|  |  |
| --- | --- |
| Input Address | D7~D2输入地址 |
| Address –> PC | K6 = 0，K7 = 0，Push A1，K7 = 1 |
| PC –> MAR，PC+1 | K6 = 1，K5 = 0，Push A2，K5 = 1， Push A1（此时PC自加1） |
| Input Data | D7~D0输入数据 |
| Write M | K3 = 0，K7 = 0， K4↓↑，K7 = 1 |

3.5.5 读取存储器：

|  |  |
| --- | --- |
| Input Address | D7~D2输入地址 |
| Address –> PC | K6 = 0，K7 = 0，Push A1，K7 = 1 |
| PC –> MAR，PC+1 | K6 = 1，K5 = 0，Push A2，K5 = 1， Push A1（此时PC自加1） |
| Read M | K3 = 1，K4=0（此时值在BUS上显示），K4 = 1 |

3.5.6 写入寄存器：

|  |  |
| --- | --- |
| Input Address | D3~D2输入地址（00=A，01=B，10=C） |
| Address –> IR（LodIR） | K7 = 0，K0↓↑，K7 = 1 |
| Input Data | D7~D0输入数据 |
| Write GR（从输入读） | K7 = 0，K2↓↑，K7 = 1 |
| Write GR（从M） | K3 = 1，K4 = 0，K2↓↑，K4 =1 |

3.5.7 读取寄存器：

|  |  |
| --- | --- |
| Input Address | D1~D0输入地址（00=A，01=B，10=C） |
| Address –> IR（LodIR） | K7 = 0，K0↓↑，K7 = 1 |
| Read GR | K1↓（此时值在BUS上显示），K1↑ |

3.5.8 拓展实验：

|  |  |
| --- | --- |
| Input Address | D7~D2输入地址0000 0001 |
| Address –> PC | K6 = 0，K7 = 0，Push A1，K7 = 1 |
| PC –> MAR，PC+1 | K6 = 1，K5 = 0，Push A2，K5 = 1， Push A1（此时PC自加1） |
| Input Data | D7~D0输入数据 1000 0000 |
| Write M | K3 = 0，K7 = 0，K4↓↑，K7 = 1 |
| PC –> MAR，PC+1 | K6 = 1，K5 = 0，Push A2，K5 = 1， Push A1（此时PC自加1） |
| Input Data | D7~D0输入数据 0100 0000 |
| Write M | K3 = 0，K7 = 0，K4↓↑，K7 = 1 |
| PC –> MAR，PC+1 | K6 = 1，K5 = 0，Push A2，K5 = 1， Push A1（此时PC自加1） |
| Input Data | D7~D0输入数据 0010 0000 |
| Write M | K3 = 0，K7 = 0，K4↓↑，K7 = 1 |
| PC –> MAR，PC+1 | K6 = 1，K5 = 0，Push A2，K5 = 1， Push A1（此时PC自加1） |
| Input Address | D7~D2输入地址 0000 0001 |
| Address –> PC | （K6 = 0，）K7 = 0，Push A1，K7 = 1 |
| PC –> MAR，PC+1 | K6 = 1，K5 = 0，Push A2，K5 = 1， Push A1（此时PC自加1） |
| Input Address | D3~D2输入地址00 |
| Address –> IR（LodIR） | K7 = 0，K0↓↑，K7 = 1 |
| Write GR（从M） | K3 = 1，K4 = 0，K2↓↑，K4 =1 |
| PC –> MAR，PC+1 | K6 = 1，K5 = 0，Push A2，K5 = 1， Push A1（此时PC自加1） |
| Input Address | D3~D2输入地址01 |
| Address –> IR（LodIR） | K7 = 0，K0↓↑，K7 = 1 |
| Write GR（从M） | K3 = 1，K4 = 0，K2↓↑，K4 =1 |
| PC –> MAR，PC+1 | K6 = 1，K5 = 0，Push A2，K5 = 1， Push A1（此时PC自加1） |
| Input Address | D3~D2输入地址10 |
| Address –> IR（LodIR） | K7 = 0，K0↓↑，K7 = 1 |
| Write GR（从M） | K3 = 1，K4 = 0，K2↓↑，K4 =1 |
| Input Address | D1~D0输入地址00 |
| Address –> IR（LodIR） | K7 = 0，K0↓↑，K7 = 1 |
| Read GR | K1↓（此时1000 0000在BUS上显示），K1↑ |
| Write SA（SA的CR#接A3#，SA的CK临时选择接入了D7） | D7↓↑（SA的灯显示出来了1000 0000，表示已存入） |
| Input Address | D1~D0输入地址01 |
| Address –> IR（LodIR） | K7 = 0，K0↓↑，K7 = 1 |
| Read GR | K1↓（此时0100 0000在BUS上显示），K1↑ |
| Write SB（SB的CR#接A3#，SA的CK临时选择接入了D6） | D6↓↑（SA的灯显示出来了0100 0000，表示已存入） |
| Input Address | D1~D0输入地址10 |
| Address –> IR（LodIR） | K7 = 0，K0↓↑，K7 = 1 |
| Read GR | K1↓（此时0010 0000在BUS上显示），K1↑ |

3.6 实验总结

通过本次总线、半导体静态存贮器实验，我们深入了解了计算机内部信息的传送和加工过程，对总线传送和半导体静态存储器的存取方法有了更深刻的理解。

在总线传送方面，我们熟悉了挂总线的逻辑器件的特性和总线传送的逻辑实现方法。通过实验操作，我们深入理解了如何通过三态门器件（如74LS244）来实现信息的总线传送，这种方法不仅有效地减少了信息传输线的数量，还显著提高了信息传送的效率和可靠性。同时，我们也掌握了建立总线的几个重要原则，如互斥性和一致性，这对于确保总线系统的稳定运行至关重要。

在半导体静态存储器方面，我们通过实验操作掌握了其存取方法。我们成功地通过控制开关和按键将数据和地址信息送入存储器，并从存储器中准确地读取了数据。这一过程让我对半导体静态存储器的工作原理有了直观的认识，并加深了我对计算机内部数据存储和读取过程的理解。

此外，本次实验还让我们对实验中所使用的各个芯片的逻辑功能有了更深入的了解，如八三态门74LS244、八D触发器74LS273以及同步四位计数器74LS161等。这些芯片在构建存储器实验电路中扮演着关键角色，对于实现数据的稳定存储和快速读取具有重要意义。

通过本次实验，我们不仅加深了对计算机组成原理的理解，还提高了自己的动手能力和实践能力。同时，我们也意识到在实验过程中需要保持严谨的态度和细致的操作，以确保实验的准确性和安全性。

总的来说，本次实验对我们而言是一次宝贵的学习经历。它不仅让我们对总线传送和半导体静态存储器的存取方法有了更深刻的认识，还为我们后续的学习和研究打下了坚实的基础。

### 4 模型计算机整机实验

4.1 实验目的

4.1.1 熟悉启停以及时序电路的特性，以及时序控制下，各个元器件的逻辑方法。

4.1.2 掌握数据格式原理，指令设计，模型计算机指令与地址的存取方法。

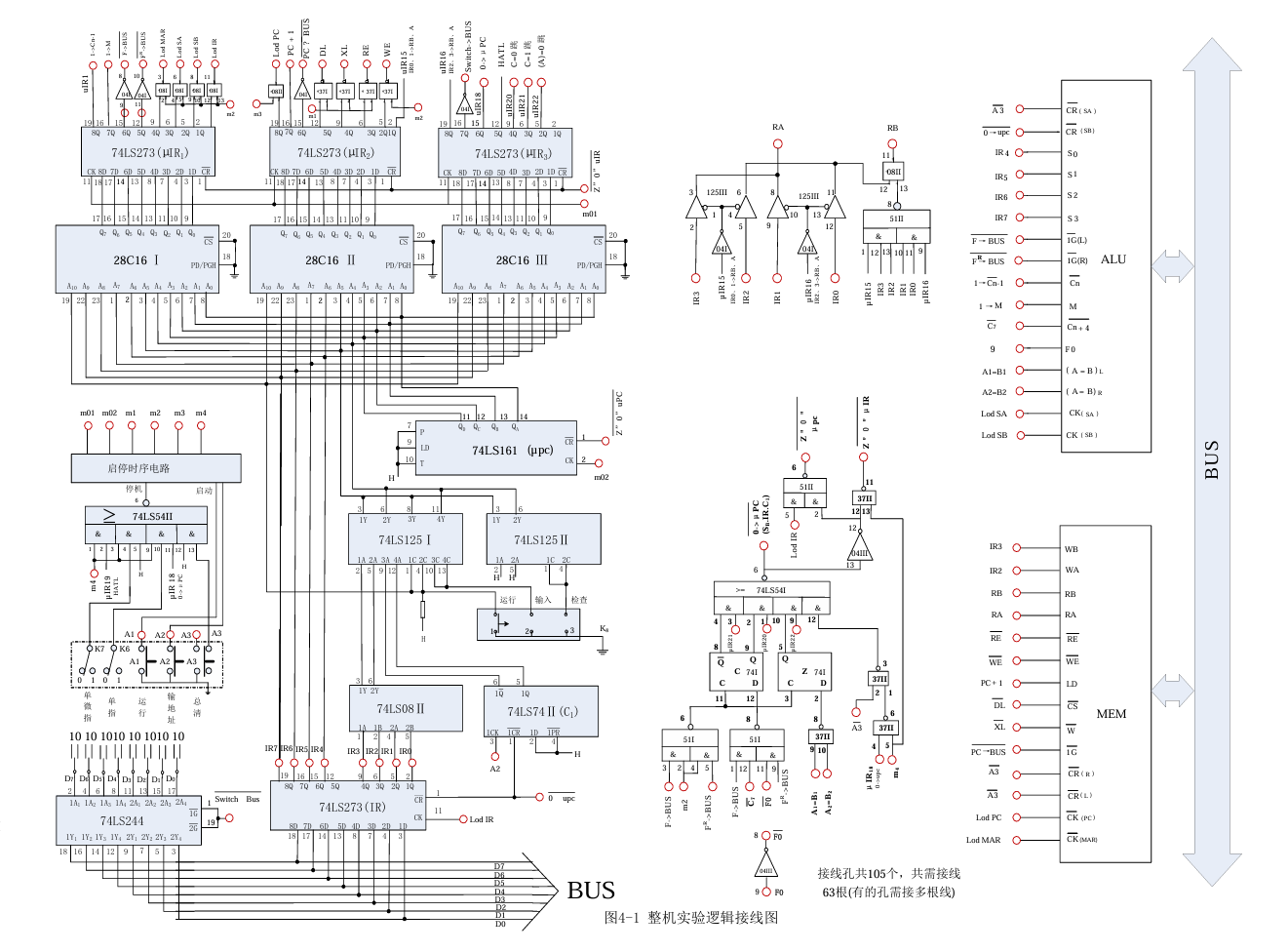
4.1.3 明白模型机的工作原理，学会自己查看微指令和数据流向。

4.2 实验原理

时序部件用来产生计算机在执行指令过程中的时序信号，计算机工作的过程中是一个指令周期接一个指令周期，在一个指令周期内部是一个机器周期接一个机器周期，机器周期内容是一个节拍接一个节拍的工作。在时序下产生有序的节拍信号，并进行机器执行。

计算机需要完备的指令系统，有着数据传输，算术逻辑运算、程序控制、输入输出四种功能的指令。

本实验我采用实验台上连线实验，原理图如下：



ALU、MEM接法按照图示即可，但需要注意的是RB要接实验台右侧的RA，RA接RB（此处检查问题浪费了半个下午，在老师的指导下成功跑通本次拓展实验所写代码）。另外就是将实验台名称一样的接口用导线连接（注意反信号）。

4.3 实验内容

4.3.1 基础实验内容

连接导线，确认所连能够正常执行命令，输入样例程序，检查内存所存是否有误。

在输入程序、检查内存后，就可以根据以下步骤运行程序。

（1）将要执行的程序入口地址值置于D7-D0 数据开关上。

（2）置K8于位置“1”。根据需要，可合上单微指开关K7或合上单指开关K6，进行单步调试，也可以K7、K6 全打开，连续运行。

（3）按A1键，机器开始执行程序。

（4）如果机器处于单指或单微指运行，则执行一条机器指令或一条微指令后，机器都会自动停机；这时需再次按A1，每按一次A1，机器都只执行一条指令或一条微指令。

在任何时候按A3总清键，机器立即回到初始状态。停机时，地址灯内容为最末执行指令的内存单元地址。

注意：当输出内容和自己的预想不符的时候，一定要用测笔测试一下输出信号，查查是否是自己的连线问题，防止自己检查很久代码之后，也没有发现错误之所在，最后结果是硬件问题。

4.3.1 拓展实验内容

输入三个数，当三个数的和大于10时，输出三个数中的最大值，当小于10时，输出三个数中的最小值。结果等于10输出最大或最小值都可以。

4.4 实验设备与器材

4.4.1 计算机组成原理实验系统 1 台

4.4.2 实验使用的部件：

1）数据输入开关 D7-D0

2）控制开关 K8，K7，K6

3）A3-A0 按键

4）实验所需全部芯片

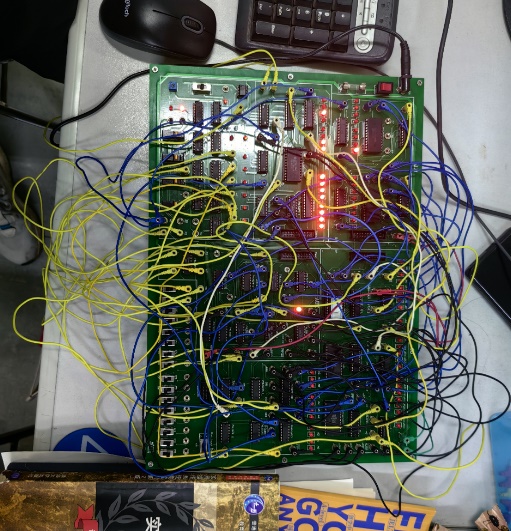
5）各器件显示灯显示输入输出信息

4.4.3 5V 稳压电源一个

4.4.4 插接导线若干根

4.5 实验步骤(该部分为重点阐述内容)

4.5.1 按照电路图连接使能各处电路器件



4.5.2 各按钮、Key的作用：

|  |  |
| --- | --- |
| 操作按钮 | 功能 |
| Button A1 | PC –> MAR，PC+1（单（微）指令模式时逐步调试功能） |
| Button A2 | 写入PC |
| Button A3 | Reset |
| K8 | Mode（Run，Input，Check） |
| K7 | 1 = 单微指令模式 |
| K6 | 1 = 单指令模式（K6和K7全为1时为连续模式） |

4.5.3 各基础操作

**4.5.3.1 输入地址**

K8 = Input，K6 = 1，Push A3（复原），Input Addr，Push A2（PC->MAR）

**4.5.3.2 输入程序模式**

K8 = Input，K6 = 1，Input Cmd，Push A1（Data->MAR，PC+1，Show Cmd & Addr）

此处可继续Input Cmd，Push A1（相当于紧跟上面地址写入Cmd），或重新输入地址，在另外的地址写入程序。

**4.5.3.3 运行程序模式**

先在K8 = Input模式输入地址，K8 = Run，（此处K7 = 1，K6 = 0是单微指令模式运行，K7 = 0，K6 = 1是单指令运行，K7 = 1，K6 = 1是连续运行），Push A1（启动程序，单指令、单微指令下时，按一下执行一条指令、单微指令）。

**4.5.3.4 检查内存模式**

先在K8 = Input模式输入地址，K8 = Check，Push A1（开始检查，显示地址和该地址内指令，其余类同运行程序模式下功能）

4.5.4 拓展实验流程：

在验收完毕基础内容之后，开始设计拓展实验。

**4.5.4.1 实验代码思路：**

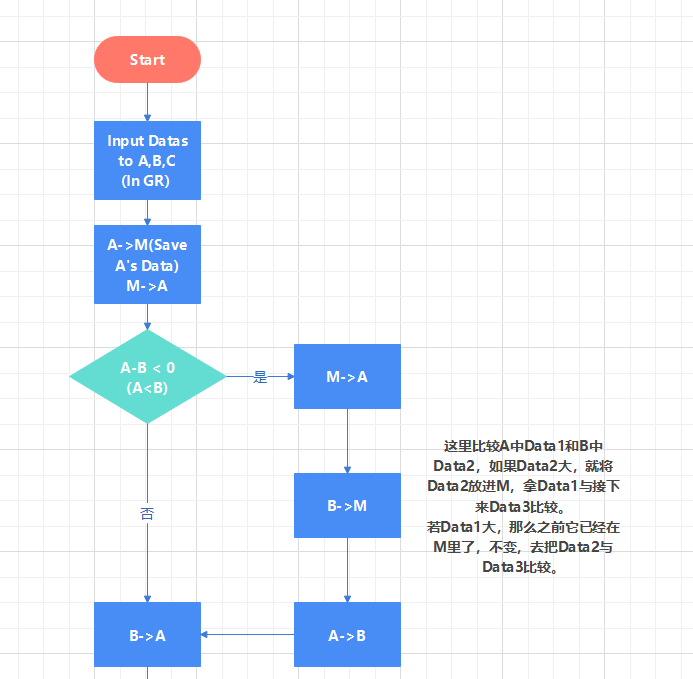
先进行输入三个数字，分别存入寄存器组A、B、C中。

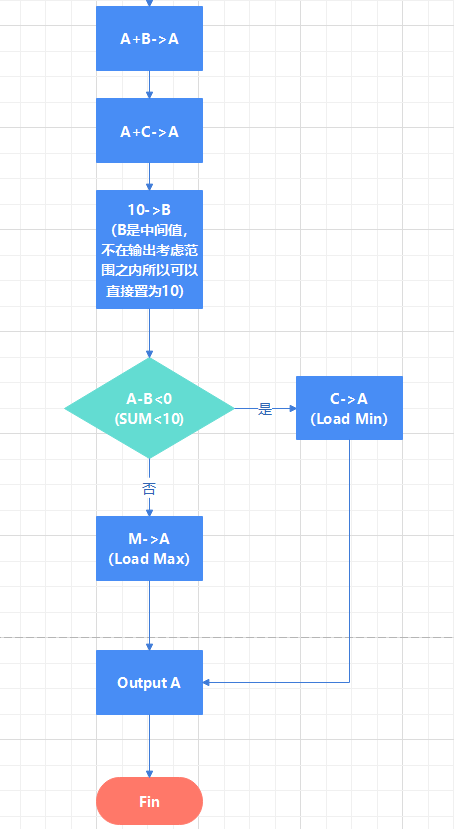
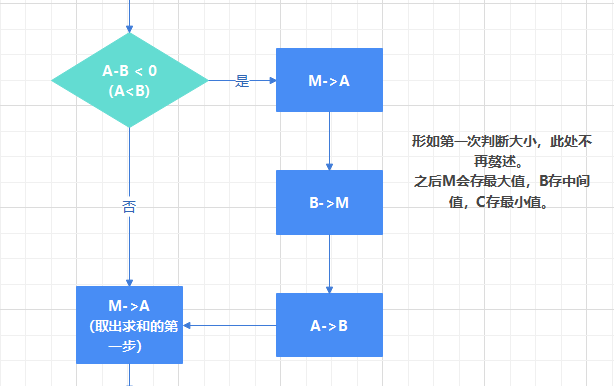
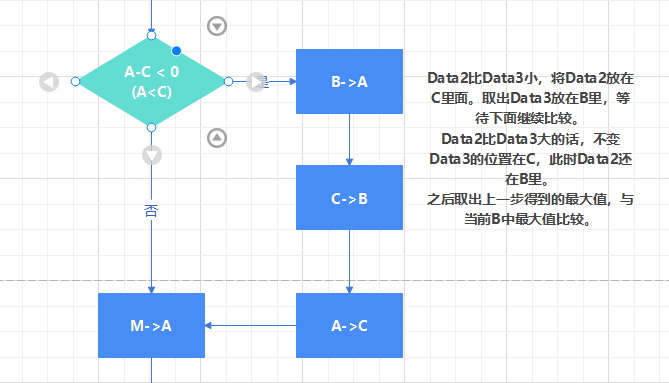
进行排序，首先是两两比较，较大的存M，较小的继续进行比较，第二次比较，小的放入C寄存器，比较大的拿出来，和M中第一次比较得到的较大值比较，这次比较得到的较大的数字就是最后的最大数字（这里比较采用了JCZ指令，发现如果A-B小于0时，C=0，实现跳转）。

此时排序完毕M存入最大值，寄存器C存入最小值，B是中间值。利用A寄存器进行求和计算，将没用的B信息置为10，与A中Sum比较大小，选择输出最大值还是最小值。

**4.5.4.2 实验代码流程图**

如图：





4.5.4.3 实验二进制代码

|  |  |  |  |
| --- | --- | --- | --- |
| 指令序号 | 操作 | 二进制码 | 功能 |
|  | IN | 0100 0000 | 1->A |
|  | IN | 0100 0100 | 2->B |
|  | IN | 0100 1000 | 3->C |
|  | STA | 1000 0000 | A->M |
|  |  | 1000 0000 |  |
|  | LDA | 0111 0000 | M->A |
|  |  | 1000 0000 |  |
|  | SUB | 0110 0001 | A-B->A |
|  | JCZ | 1101 1111 | <0,JMP 38 |
|  |  | 0010 0110 |  |
|  | MOV | 0010 0001 | B->A |
|  | SUB | 0110 0010 | A-C->A |
|  | JCZ | 1101 1111 | <0,JMP 45 |
|  |  | 0010 1101 |  |
|  | LDA | 0111 0000 | M->A |
|  |  | 1000 0000 |  |
|  | SUB | 0110 0001 | A-B->A |
|  | JCZ | 1101 1111 | <0,JMP 50 |
|  |  | 0011 0010 |  |
|  | LDA | 0111 0000 | M->A |
|  |  | 1000 0000 |  |
|  | ADD | 1001 0001 | A+B->A |
|  | ADD | 1001 0010 | A+C->A |
|  | MVI | 0001 0100 | 10->B |
|  |  | 0000 1010 |  |
|  | SUB | 0110 0001 | A-B->A |
|  | JCZ | 1101 1111 | <0,JMP 34 |
|  |  | 0010 0010 |  |
|  | LDA | 0111 0000 | M->A |
|  |  | 1000 0000 |  |
|  | OUT | 0100 0011 | LIGHT |
|  | JMP | 1101 0000 | RETURN |
|  |  | 0000 0001 |  |
|  | MOV | 0010 0010 | C->A |
|  | OUT | 0100 0011 | LIGHT |
|  | JMP | 1101 0000 | RETURN |
|  |  | 0000 0001 |  |
|  | LDA | 0111 0000 | M->A |
|  |  | 1000 0000 |  |
|  | STA | 1000 0001 | B->M |
|  |  | 1000 0000 |  |
|  | MOV | 0010 0100 | A->B |
|  | JMP | 1101 0000 | JMP 11 |
|  |  | 0000 1011 |  |
|  | MOV | 0010 0001 | B->A |
|  | MOV | 0010 0110 | C->B |
|  | MOV | 0010 1000 | A->C |
|  | JMP | 1101 0000 | JMP 15 |
|  |  | 0000 1111 |  |
|  | LDA | 0111 0000 | M->A |
|  |  | 1000 0000 |  |
|  | STA | 1000 0001 | B->M |
|  |  | 1000 0000 |  |
|  | MOV | 0010 0100 | A->B |
|  | JMP | 1101 0000 | JMP 20 |
|  |  | 0001 0100 |  |

4.6 实验总结

在这次的实验中，我们对启停电路及时间序列电路的特性有了更加深刻的理解。通过实操，我们观察了时钟在整个程序运行时的重要性，这使我们对整个时序控制的逻辑方法有了明晰的认识。

实验的开始是对启停电路的探索。通过实际接线并操作时钟使能按钮，我们观察到了各处LED指示灯的状态变化，这种直观的感受增强了我对启停电路以及模型PC机工作方式的理解。随后，我们进行了时序电路的实验，手动给出脉冲，同时也体会到了控制信号传递的精确性和复杂性。

此次实验最大的收获，是我对模型计算机工作原理的理解更加深入。通过查看微指令以及数据流向，编写类似于汇编的指令，并手动写入存储器，我们开始学会如何解析和理解微操作的执行过程，并对汇编语言以及冯诺依曼式计算机结构有了更深刻的理解。这对于我们的学习和未来的科研工作都具有重要的意义。

数据格式的处理原理以及指令设计部分对我们来说同样充满挑战。通过对不同指令的模拟执行，并将多个指令执行，完成综合实验，我掌握了如何在模型计算机中存取指令与地址的方法。这不仅考验了我们的逻辑思维能力，也锻炼了我们的实践操作技能。

通过本次实验，我们深刻感受到了时序控制在计算机组成中的重要作用。从理论到实践的转变过程中，每一步操作都让我更加贴近实际工作状态，每一次问题的解决都提升了我们解决问题的能力。此外，我们加深了对计算机行业合作和指导的重要性的理解，这将是我们宝贵的学术财富，也是未来不断探索和进步的动力。