**暨南大学本科实验报告专用纸**

课程名称 计算机组成原理实验 指导教师 王勇杰 成绩

实验项目名称 数据通路实验 实验项目编号 080600643

实验项目实验地点B404 学院智能科学与工程学院专业 人工智能

学生姓名王志涛学号2021102259 实验时间 2023 年 4 月 21 日**实验目的**

1.1将微程序控制器同执行部件（整个数据通路）联机，组成一台模型计算机。

1.2用微程序控制器控制模型计算机的数据通路。

1.3通过TEC-5执行由8条机器指令组成的简单程序，掌握机器指令与微指令的关系，牢固建立计算机的整机概念。

1. **实验电路**

2.1本次实验将前面几个实验中的所有电路，包括运算器、存储器、通用寄存器堆、微程序控制器等模块组合在一起，构成一台简单的模型机。因此，在基本实验中，这是最复杂的一个实验，也将是最有收获的一个实验。

2.2在前面的实验中，实验者本身作为“控制器”，完成了对数据通路的控制。而在本次实验中，数据通路的控制将交由微程序控制器来完成。TEC-5从内存中取出一条机器指令到执行指令结束的一个指令周期，是由微程序完成的，即一条机器指令对应一个微程序序列。



1. **实验任务与过程**

3.1.本组的寄存器数据是R0=35H，R1=43H，R2=10H，R3=07H。存储器10H单元的内容为55H。选择这组数据的目的是执行ADD R1,R0指令时不产生进位C，从而在执行JC R3指令时不产生跳转，而是顺序执行。

* + 设置通用寄存器R0、R1、R2和R3的值

在本操作中，我们打算使R0=35H，R1=43H，R2=10H，R3=07H

1．令DP=0，DB=0，使系统处于连续运行状态。令SWC=0，SWB=1，SWA=1，使系统处于写寄存器状态WRF。按CLR#按钮，使实验系统处于初始状态。

1. 在SW7—SW0上设置一个存储器地址，该存储器地址供设置通用寄存器使用。该存储器地址最好是不常用的一个地址，以免设置通用寄存器操作破坏重要的存储器单元的内容。例如可将该地址设置为0FFH。按一次QD按钮，将0FFH写入左端口地址寄存器AR。
2. 在SW7—SW0上设置00H，作为通用寄存器R0的寄存器号。按一次QD按钮，将00H写入指令寄存器IR。
3. 在SW7—SW0设置35H，按一次QD按钮，将35H写入IR指定的R0寄存器。
4. 在SW7—SW0上设置01H，作为通用寄存器R1的寄存器号。按一次QD按钮，将01H写入指令寄存器IR。
5. 在SW7—SW0设置43H，按一次QD按钮，将43H写入IR指定的R1寄存器。
6. 在SW7—SW0上设置02H，作为通用寄存器R2的寄存器号。按一次QD按钮，将02H写入指令寄存器IR。
7. 在SW7—SW0设置10H，按一次QD按钮，将10H写入IR指定的R2寄存器。
8. 在SW7—SW0上设置03H，作为通用寄存器R3的寄存器号。按一次QD按钮，将03H写入指令寄存器IR。
9. 在SW7—SW0设置07H，按一次QD按钮，将07H写入IR指定的R3寄存器。

设置寄存器内容完毕。按CLR#按钮，使系统恢复到初始状态。

注意：设置完R0、R1、R2、R3的值后，最好用读寄存器控制台操作检查一下写入的内容是否正确。读寄存器内容的控制台操作见第一章的第七部分控制台。

* + 存程序机器代码

本操作中，我们从00地址开始存8个机器代码：01H，5CH，39H，4AH，22H，1EH，78H，60H。在10H单元存入55H，作为10H单元的初值，以检查LDA和STA两条指令的作用。

* + 1. 令DP = 0，DB = 0，使实验系统处于连续运行状态。令SWC = 0、SWB = 0、SWA = 1，使实验系统处于写双端口存储器工作方式WRM。按CLR#按钮，使实验系统处于初始状态。

2．置SW7—SW0为00H，按QD按钮，将00H写入左端口地址寄存器AR。

3．置SW7—SW0 为01H，按QD按钮，将01H写入存储器00H单元。AR自动加1，变为01H。

4．置SW7—SW0 为5CH，按QD按钮，将5CH写入存储器01H单元。AR自动加1，变为02H。

5．置SW7—SW0 为39H，按QD按钮，将39H写入存储器02H单元。AR自动加1，变为03H。

6．置SW7—SW0 为4AH，按QD按钮，将4AH写入存储器03H单元。AR自动加1，变为04H。

7．置SW7—SW0 为22H，按QD按钮，将22H写入存储器04H单元。AR自动加1，变为05H。

8．置SW7—SW0 为1EH，按QD按钮，将1EH写入存储器05H单元。AR自动加1，变为06H。

7．置SW7—SW0 为78H，按QD按钮，将78H写入存储器06H单元。AR自动加1，变为07H。

8．置SW7—SW0 为60H，按QD按钮，将60H写入存储器07H单元。AR自动加1，变为08H。

9．按CLR#按钮，使实验系统恢复到初始状态。

10．置SW7—SW0为10H，按QD按钮，将10H写入左端口地址寄存器AR。

11．置SW7—SW0 为55H，按QD按钮，将55H写入存储器10H单元。AR自动加1，变为11H。

往存储器输入程序和数据结束，按CLR#按钮，使实验系统恢复到初始状态。

注意：设置完存储器的程序和数据后，最好用存储器器控制台操作检查一下写入的内容是否正确。读存储器内容的控制台操作见第一章的第七部分控制台。

●用单拍（DP）方式执行一遍程序。

置SWC=0,SWB=0,SWA=0,DP=1,DB=0,使实验系统处于单拍运行状态。置SW7—SW0=00H，使程序从地址00H开始执行。按CLR#按钮，使实验系统处于初始状态，然后一次一次按QD按钮，使程序一拍一拍的执行。

在单拍执行过程中，首先要随时监测AR、PC、µA5—µA0和IR的值，以判定程序执行到何处，正在执行哪条指令和微指令。对照微程序流程图，可以判断出微指令的地址和正在进行的微操作。程序执行的结果如下：

初值：R0=35H，R1=43H，R2=10H，R3=07H。存储器10H单元的内容为55H。

1．ADD R1,R0

执行结果 R0=35H，R1=78H，R2=10H，R3=07H。存储器10H单元的内容为55H。无进位C。

2．JC R3

执行结果 R0=35H，R1=78H，R2=10H，R3=07H。存储器10H单元的内容为55H。PC为02H。进位C不变。

3．STA R1,[R2]

执行结果 R0=35H，R1=78H，R2=10H，R3=07H。存储器10H单元的内容为78H。

4．LDA R2,[R2]

执行结果 R0=35H，R1=78H，R2=78H，R3=07H。存储器10H单元的内容为78H。

5．AND R2,R0

执行结果 R0=35H，R1=78H，R2=30H，R3=07H。存储器10H单元的内容为78H。

6．SUB R2,R3

执行结果 R0=35H，R1=78H，R2=29H，R3=07H。存储器10H单元的内容为78H。进位C为1。

7．OUT R2

执行结果 R0=35H，R1=78H，R2=29H，R3=07H。存储器10H单元的内容为78H。可在数据总线DBUS指示灯上观察到29H。

8．STP

执行结果 R0=35H，R1=78H，R2=29H，R3=07H。存储器10H单元的内容为78H。

最后的执行结果可通过控制台的读寄存器操作和读存储器操作观察到，在观察最后结果之前，首先应按CLR#按钮，使实验系统处于初始状态。

●用连续方式执行一遍程序。

1．由于上面的单拍执行程序，已破坏了寄存器R1、R2和存储器10单元的内容（程序没有破坏），因此需要重新设置寄存器R1、R2和存储器10单元的值。初值：R0=35H，R1=43H，R2=10H，R3=07H。存储器10H单元的内容为55H。

2．置SWC=0,SWB=0,SWA=0,DP=0,DB=0,使实验系统处于连续运行状态。置SW7—SW0=00H，使程序从地址00H开始执行。按CLR#按钮，使实验系统处于初始状态，然后按一次QD按钮，则程序自动连续运行到地址为07H的STP指令。

执行结果 R0=35H，R1=78H，R2=29H，R3=07H。存储器10H单元的内容为78H。

最后的执行结果可通过控制台的读寄存器操作和读存储器操作观察到，在观察最后结果之前，首先应按CLR#按钮，使实验系统处于初始状态。

3.2存程序代码，设置通用寄存器R0、R1、R2和R3的第二组值及存储器相关单元的数据。

本组的寄存器数据是R0=86H，R1=88H，R2=10H，R3=07H。存储器10H单元的内容为55H。选择这组数据的目的是执行ADD R1,R0指令时产生进位C，从而在执行JC R3指令时产生跳转，而不是顺序执行。

* + 设置通用寄存器R0、R1、R2和R3的值

在本操作中，我们打算使R0=86H，R1=88H，R2=10H，R3=07H

1．令DP=0，DB=0，使系统处于连续运行状态。令SWC=0，SWB=1，SWA=1，使系统处于写寄存器状态WRF。按CLR#按钮，使实验系统处于初始状态。

2．在SW7—SW0上设置一个存储器地址，该存储器地址供设置通用寄存器使用。该存储器地址最好是不常用的一个地址，以免设置通用寄存器操作破坏重要的存储器单元的内容。例如可将该地址设置为0FFH。按一次QD按钮，将0FFH写入左端口地址寄存器AR。

3．在SW7—SW0上设置00H，作为通用寄存器R0的寄存器号。按一次QD按钮，将00H写入指令寄存器IR。

4．在SW7—SW0设置86H，按一次QD按钮，将86H写入IR指定的R0寄存器。

5．在SW7—SW0上设置01H，作为通用寄存器R1的寄存器号。按一次QD按钮，将01H写入指令寄存器IR。

6．在SW7—SW0设置88H，按一次QD按钮，将88H写入IR指定的R1寄存器。

7．在SW7—SW0上设置02H，作为通用寄存器R2的寄存器号。按一次QD按钮，将02H写入指令寄存器IR。

8．在SW7—SW0设置10H，按一次QD按钮，将10H写入IR指定的R0寄存器。

9．SW7—SW0上设置03H，作为通用寄存器R3的寄存器号。按一次QD按钮，将03H写入指令寄存器IR。

10．在SW7—SW0设置07H，按一次QD按钮，将07H写入IR指定的R3寄存器。

设置寄存器内容完毕。按CLR#按钮，使系统恢复到初始状态。

注意：设置完R0、R1、R2、R3的值后，最好用读寄存器控制台操作检查一下写入的内容是否正确。读寄存器内容的控制台操作见第一章的第七部分控制台。

* + 存程序机器代码

本操作中，我们从00地址开始存8个机器代码：01H，5CH，39H，4AH，22H，1EH，78H，60H。在10H单元存入55H，作为10H单元的初值，以检查LDA和STA两条指令的作用。

* + 1. 令DP = 0，DB = 0，使实验系统处于连续运行状态。令SWC = 0、SWB = 0、SWA = 1，使实验系统处于写双端口存储器工作方式WRM。按CLR#按钮，使实验系统处于初始状态。

2．置SW7—SW0为00H，按QD按钮，将00H写入左端口地址寄存器AR。

3．置SW7—SW0 为01H，按QD按钮，将01H写入存储器00H单元。AR自动加1，变为01H。

4．置SW7—SW0 为5CH，按QD按钮，将5CH写入存储器01H单元。AR自动加1，变为02H。

5．置SW7—SW0 为39H，按QD按钮，将39H写入存储器02H单元。AR自动加1，变为03H。

6．置SW7—SW0 为4AH，按QD按钮，将4AH写入存储器03H单元。AR自动加1，变为04H。

7．置SW7—SW0 为22H，按QD按钮，将22H写入存储器04H单元。AR自动加1，变为05H。

8．置SW7—SW0 为1EH，按QD按钮，将1EH写入存储器05H单元。AR自动加1，变为06H。

7．置SW7—SW0 为78H，按QD按钮，将78H写入存储器06H单元。AR自动加1，变为07H。

8．置SW7—SW0 为60H，按QD按钮，将60H写入存储器07H单元。AR自动加1，变为08H。

9．按CLR#按钮，使实验系统恢复到初始状态。

10．置SW7—SW0为10H，按QD按钮，将10H写入左端口地址寄存器AR。

11．置SW7—SW0 为55H，按QD按钮，将55H写入存储器10H单元。AR自动加1，变为11H。

往存储器输入程序和数据结束，按CLR#按钮，使实验系统恢复到初始状态。

注意：设置完存储器的程序和数据后，最好用存储器器控制台操作检查一下写入的内容是否正确。读存储器内容的控制台操作见第一章的第七部分控制台。

●用单拍（DP）方式执行一遍程序。

置SWC=0,SWB=0,SWA=0,DP=1,DB=0,使实验系统处于单拍运行状态。置SW7—SW0=00H，使程序从地址00H开始执行。按CLR#按钮，使实验系统处于初始状态，然后一次一次按QD按钮，使程序一拍一拍的执行。

在单拍执行过程中，首先要随时监测AR、PC、µA5—µA0和IR的值，以判定程序执行到何处，正在执行哪条指令和微指令。对照微程序流程图，可以判断出微指令的地址和正在进行的微操作。程序执行的结果如下：

初值：R0=86H，R1=88H，R2=10H，R3=07H。存储器10H单元的内容为55H。

1．ADD R1,R0

执行结果 R0=86H，R1=0EH，R2=10H，R3=07H。存储器10H单元的内容为55H。有进位C。

2．JC R3

执行结果 R0=86H，R1=0EH，R2=10H，R3=07H。存储器10H单元的内容为55H。PC为07H。进位C不变。

3．STP

执行结果 R0=86H，R1=0EH，R2=10H，R3=07H。存储器10H单元的内容为55H，进位C为1。

最后的执行结果可通过控制台的读寄存器操作和读存储器操作观察到，在观察最后结果之前，首先应按CLR#按钮，使实验系统处于初始状态。

●用连续方式执行一遍程序。

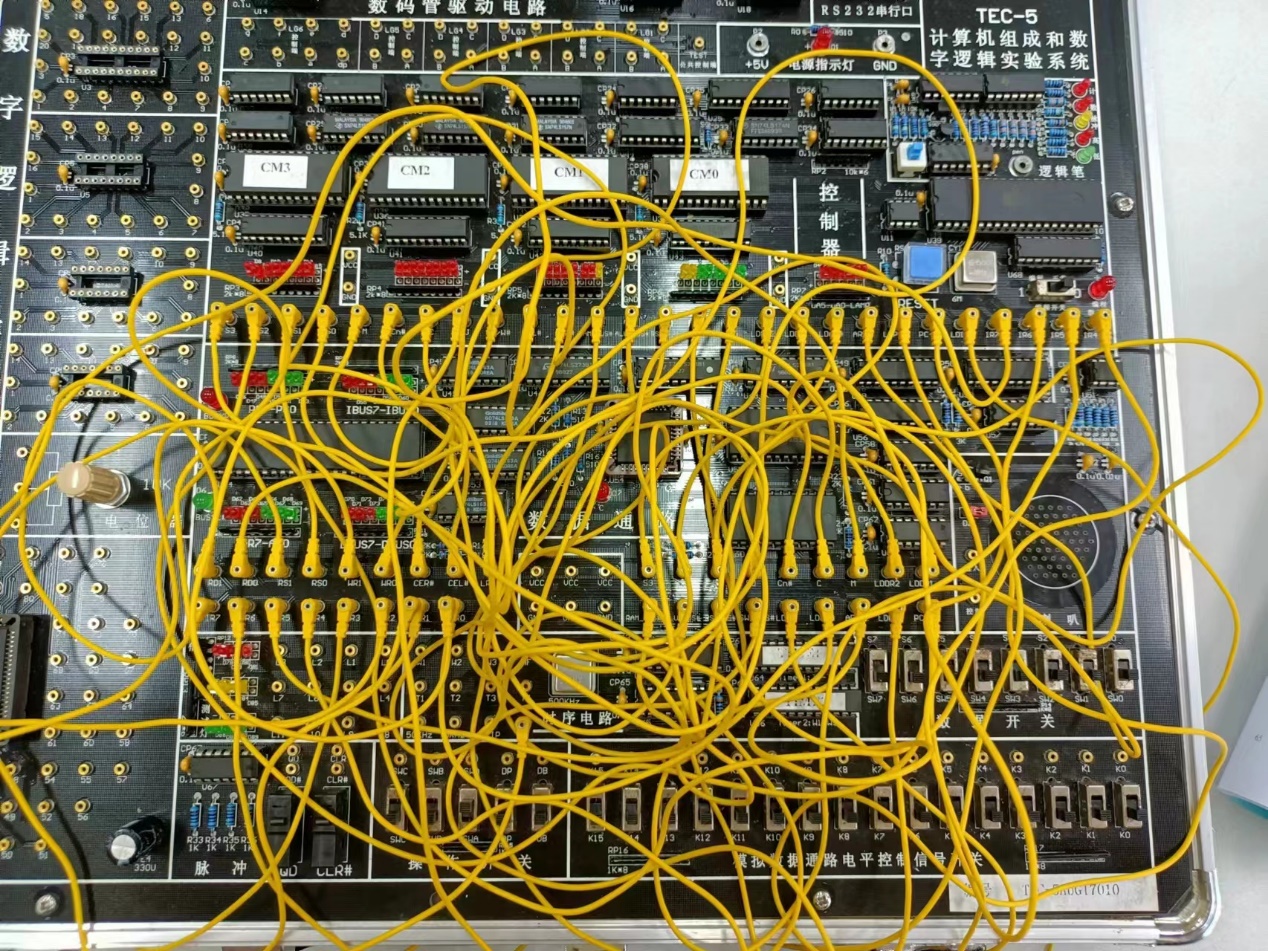
1．由于上面的单拍执行程序，已破坏了寄存器R1内容（程序没有破坏），因此需要重新设置寄存器R1的值。初值：R0=86H，R1=88H，R2=10H，R3=07H。存储器10H单元的内容为55H。

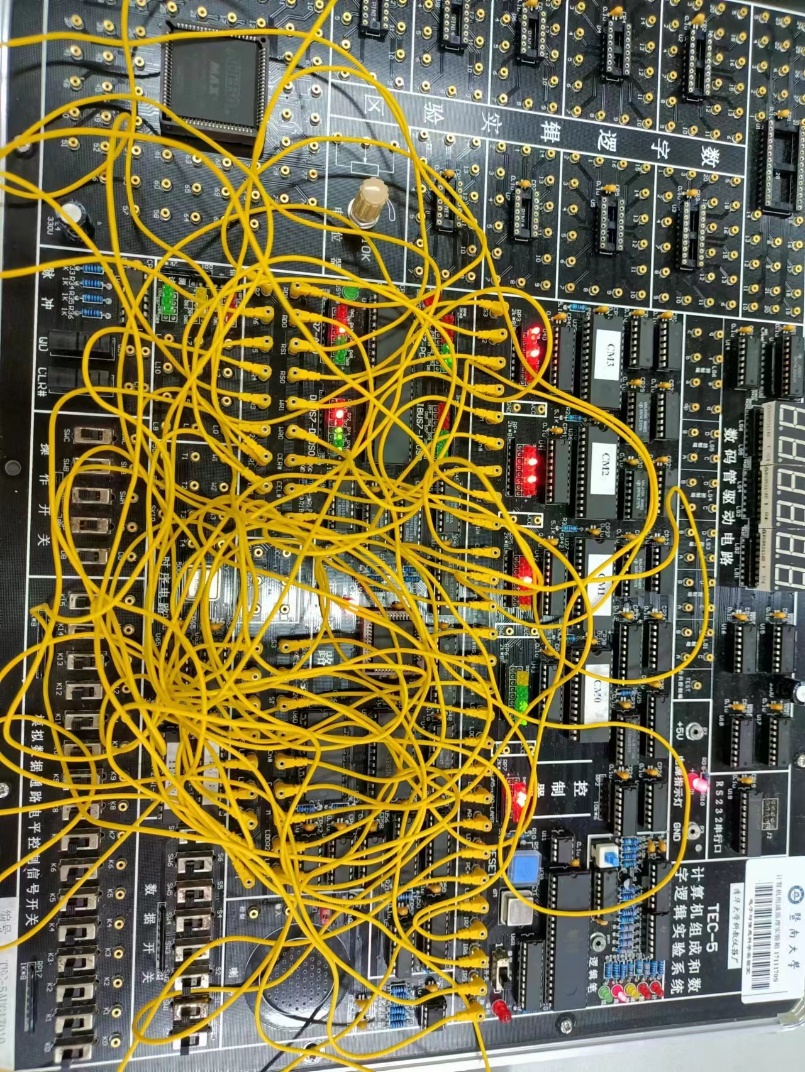
2．置SWC=0,SWB=0,SWA=0,DP=0,DB=0,使实验系统处于连续运行状态。置SW7—SW0=00H，使程序从地址00H开始执行。按CLR#按钮，使实验系统处于初始状态，然后按一次QD按钮，则程序自动连续运行到地址为07H的STP指令。

执行结果 R0=86H，R1=0EH，R2=10H，R3=07H。存储器10H单元的内容为55H。

最后的执行结果可通过控制台的读寄存器操作和读存储器操作观察到，在观察最后结果之前，首先应按CLR#按钮，使实验系统处于初始状态。

3.3实验过程图





1. **实验评估**

在这次实验中，我学到了以下几点：

1.我掌握了机器指令和微指令之间的关系。机器指令是由程序员编写的指令，描述了要执行的任务和操作的顺序。而微指令则是机器指令的内部细分，用于控制计算机硬件的具体操作。通过解码机器指令并将其转换为对应的硬件操作序列，微指令实现了机器指令的执行。

2.我建立起了计算机的整体概念。在我们的模型机中，包括控制台、微程序控制器、时序发生器和数据通路这四个部分。控制台提供开关功能，微程序控制器提供指示灯信号和控制信号，时序发生器提供时序信号，而数据通路则负责指令代码和条件信号的传输。这些部件组成了一个模型计算机，能够基本代表现实计算机的基本逻辑结构。

3.我明白了程序在计算机内部的运行机制。首先，我们需要设置数据段，即给通用寄存器设置初始值，为程序的运行提供必要的数据。其次，我们设置代码段，将机器代码存储在双端口存储器的相应存储单元中，类比计算机内部的内存。最后，我们可以通过单拍方式或连续方式运行程序。根据时序信号，程序按顺序执行代码。每执行一条机器指令，都会使用相应的数据（如果该指令需要数据）

4.我了解了代码的顺序执行和跳转执行。顺序执行时，指令地址寄存器的值会自动加一，从而使计算机能够高效地转到下一条指令的执行。而跳转执行则根据条件信号设置指令地址寄存器的值，从而改变程序的执行流程。顺序指令按照编写顺序逐条执行，不改变程序的控制流程，而跳转指令可以改变程序的执行顺序，实现条件判断、循环控制和函数调用等功能。这两种指令类型的组合使用能实现复杂的控制逻辑和功能，使程序具备灵活性和实现复杂逻辑的能力。

在本次实验中，我深入学习了计算机的指令执行过程和计算机内部的工作原理。通过掌握机器指令和微指令的关系，我了解到机器指令是程序员编写的指令，而微指令则是机器指令的内部细分，用于具体控制计算机硬件的操作。这种理解为我进一步学习计算机体系结构和指令集设计提供了基础。

在实验过程中，我建立了对计算机整体概念的认识。我了解到计算机由控制台、微程序控制器、时序发生器和数据通路等部件组成，每个部件都有着特定的功能和作用。这种整体概念的建立使我能够更好地理解计算机的基本逻辑结构，并将其与实际计算机相对应。

此外，我对程序在计算机内部的运行机制有了更深入的了解。通过设置数据段和代码段，并 掌握了单拍方式和连续方式的运行方式，我学会了顺序执行代码的过程。我还了解到跳转指令的重要性，它能够改变程序的执行流程，实现条件判断、循环控制和函数调用等功能。这对于编写高效、灵活的程序至关重要。

这是在实验室里做的最后一个实验，对我来说，它给我留下了深刻的体会。

这次实验是我进行的五个实验中最耗时的一个。由于实验中涉及到的连线比其他四个实验都要更为复杂，我必须非常仔细地进行操作，以免出现接线错误导致实验无法按照预期进行。然而，尽管我尽力避免错误，我还是遇到了许多问题。我怀疑这些问题与接线有关，但我反复检查后发现接线并没有错误，然而实验结果仍然与我的预期不符。这复杂的连线给我带来了巨大的挑战。

基于之前的经验，我开始怀疑可能是设备本身存在问题。因此，我尝试更换了多台设备，但每次更换设备都需要花费大量时间重新进行连线。我知道，一丝不慎都可能导致实验失败。

经过多次尝试之后，我已经习惯了为了一个实验尝试多个设备的过程。虽然这个过程曲折艰辛，但最终我获得了令人欣慰的结果，实验也顺利地按照预期完成了。

总结起来，这次实验对我来说是一次宝贵的经历。它教会了我耐心和细致，让我学会在面对复杂问题时不轻易放弃。通过这次实验，我更深刻地体会到了实验过程中的挑战和困难，并学会了在面对困难时保持冷静并持续努力。这种经验将对我的未来学习和职业发展产生积极的影响，让我更加自信地面对各种挑战。同时，我也明白了实验中设备的重要性，准备工作的细致程度和设备的可靠性都会直接影响实验结果的准确性。因此，我会更加注重实验前的准备工作，并在实验中更加细致地检查设备，以确保实验能够顺利进行。