



# 《计算机组成原理》实验报告

实验名称	控制器实验	实验时间	2024 年 10 月 25 日
------	-------	------	------------------

## 1 实验目的

- (1) 掌握时序发生器的原理及其设计方法。
- (2) 熟悉 FPGA 应用设计及 EDA 软件的使用。
- (3) 掌握微程序控制器的组成原理。
- (4) 掌握微程序的编制、写入，观察微程序的运行过程。
- (5) 基于数据通路图，掌握微程序控制器的工作原理。
- (6) 基于微程序流程图，掌握微程序控制器的工作原理。
- (7) 基于信号时序图，掌握微程序控制器的工作原理。

## 2 实验主要内容及过程

### 时序发生器设计实验

(1) 参照上面的实验原理，用 VHDL 语言来具体设计一个时序发生器。使用 Quartus 软件编辑 VHDL 文件并进行编译，时序发生器在 FPGA 芯片中对应的引脚如图 3-1-3 所示，框外文字表示连接标号，框内文字表示该引脚的含义（本实验例程见‘安装路径\FPGA\Timer\Timer.qpf’工程）

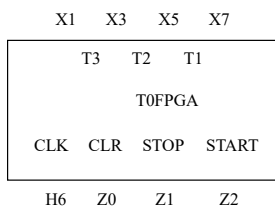


图 3-1-3 实验接线图

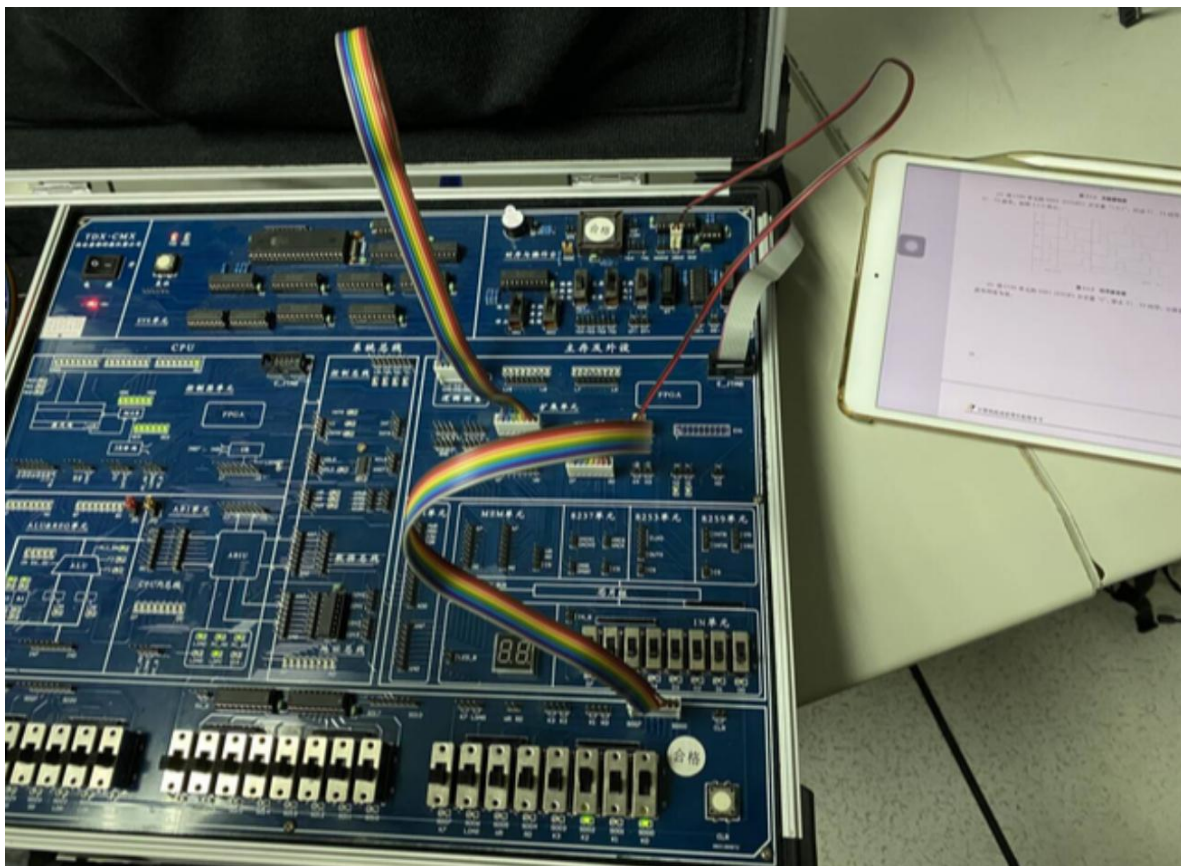


图 3.1.1

(2) 关闭实验系统电源，按图 3-1-4 连接实验电路，并检查无误，图中将用户需要连接的信号用圆圈标明。实验实际接线图如3.1.1所示。

(3) 打开实验系统电源，将下载电缆插入扩展单元的 E\_JTAG 口，把生成的 SOF 文件下载到扩展单元中去，扩展单元介绍见实验 1.2。下载过程图如3.1.2所示。

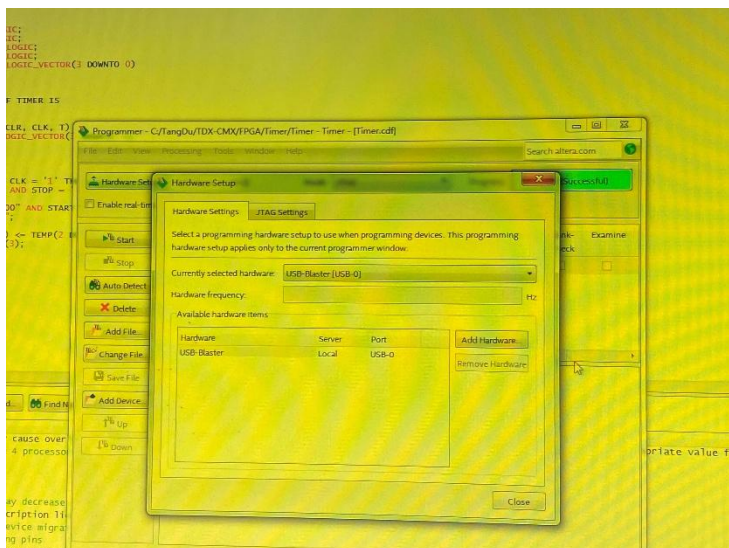


图 3.1.2

(4) 将 CON 单元的 SD02 (START) SD01 (STOP) 开关置‘1’，拨动SD00 (CLR) 置 ‘0’然后置‘1’，使T1...T4 输出低。运行联机软件，选择“【波形】—【打开】”打开逻辑示波器窗口，然后选择“【波形】—【运行】”启动逻辑示波器，逻辑示波器窗口显示 T1...T4 四路时序信号波形。如图3.1.3所示：

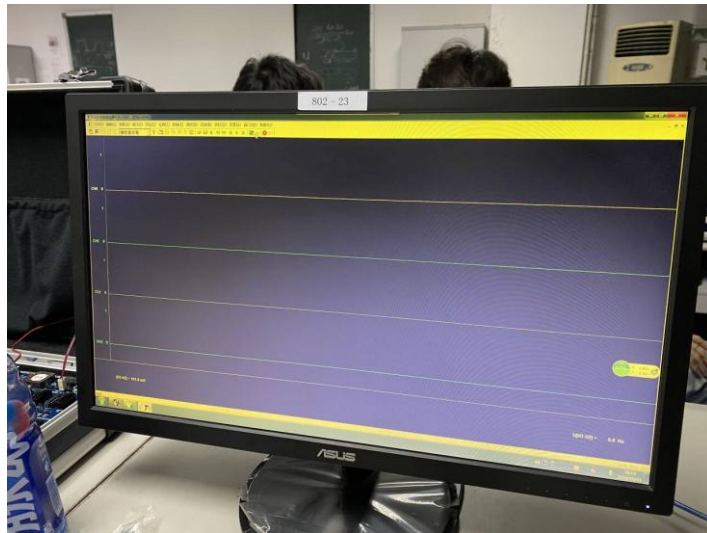


图 3.1.3

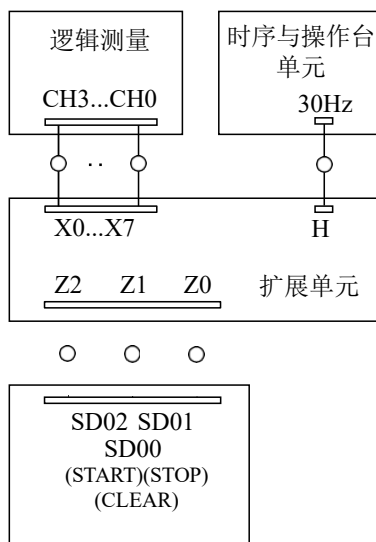


图 3-1-4 实验接线图

(5) 将 CON 单元的 SD02 (START) 开关置‘1-0-1’, 启动T1...T4 时序, 示波器窗口显示 T1...T4 波形, 如图 3-1-5 所示。

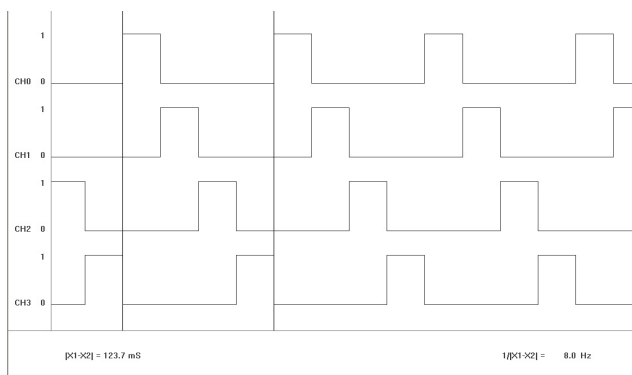


图 3-1-5 时序波形图

实验结果如图 3.1.4 所示。

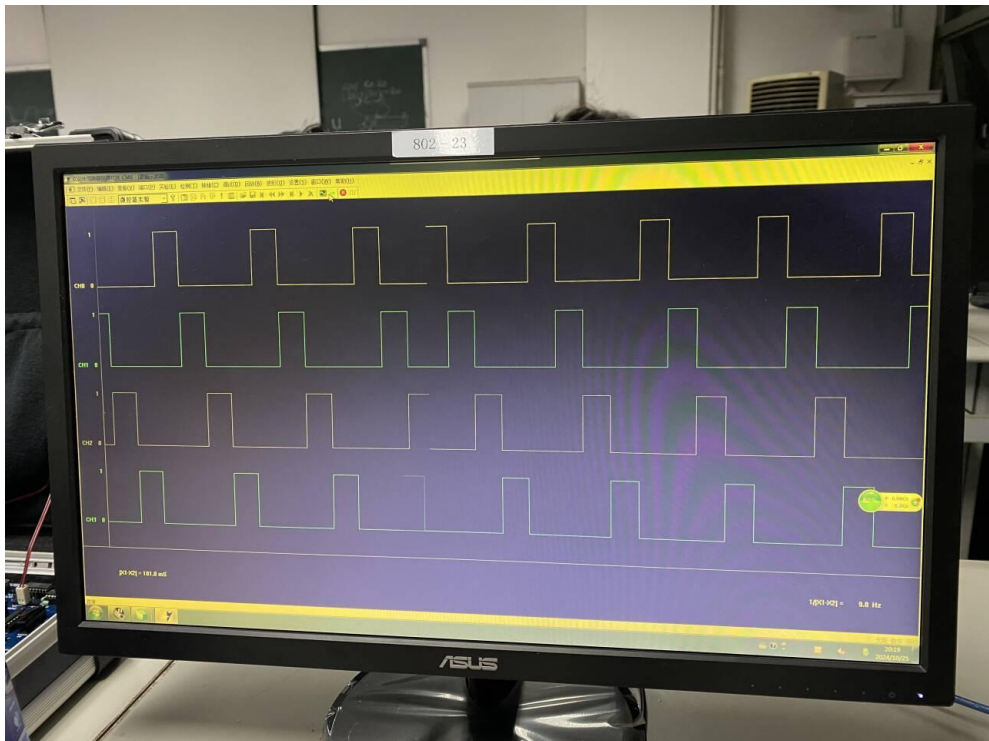


图 3.1.4

(6) 将CON 单元的 SD01 (STOP)开关置‘0’, 停止T1...T4 时序, 示波器窗口显示 T1...T4 波形均变为低。实验结果如图 3.1.5 所示。

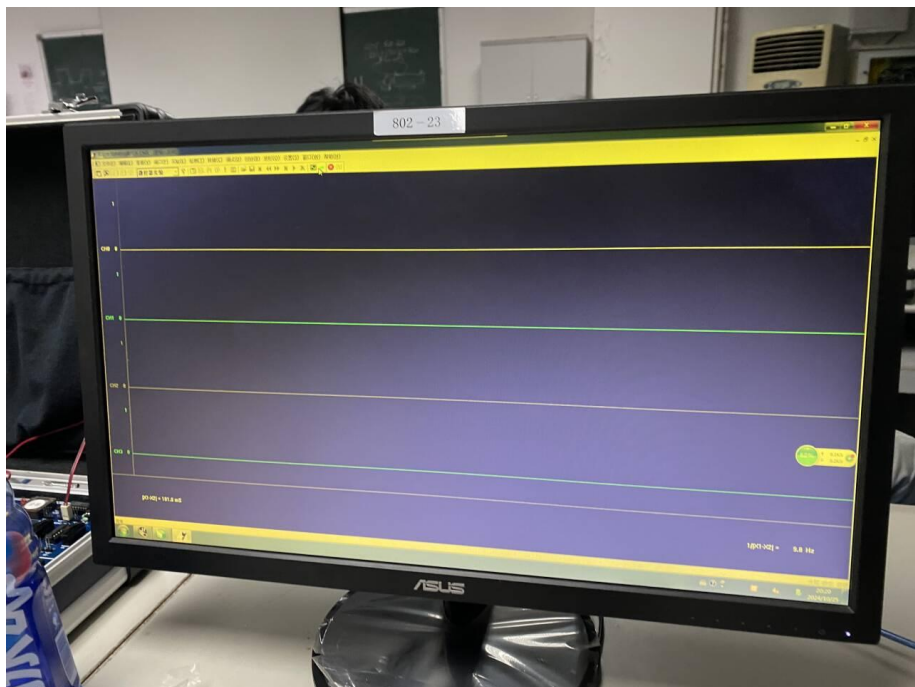


图 3.1.45



## 微程序控制器实验

1. 把时序与操作台单元的“MODE”用短路块短接，使系统工作在四节拍模式，JP1 用短路块将 1、2 短接，按图 3-2-10 所示连接实验线路，仔细查线无误后接通电源。如果有‘滴’报警声，说明总线有竞争现象，应关闭电源，检查接线，直到错误排除。

实验截图如3. 2. 1所示：

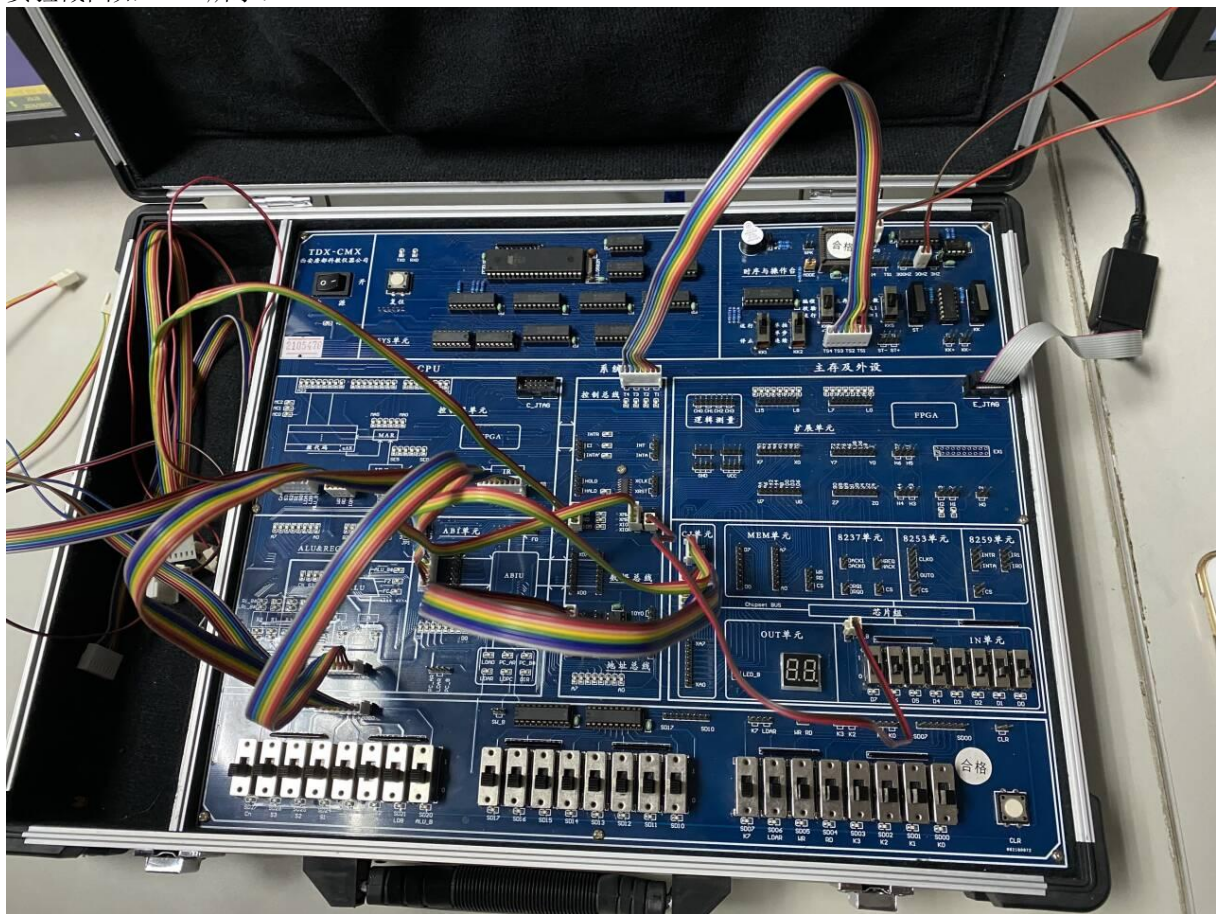


图 3.2.1

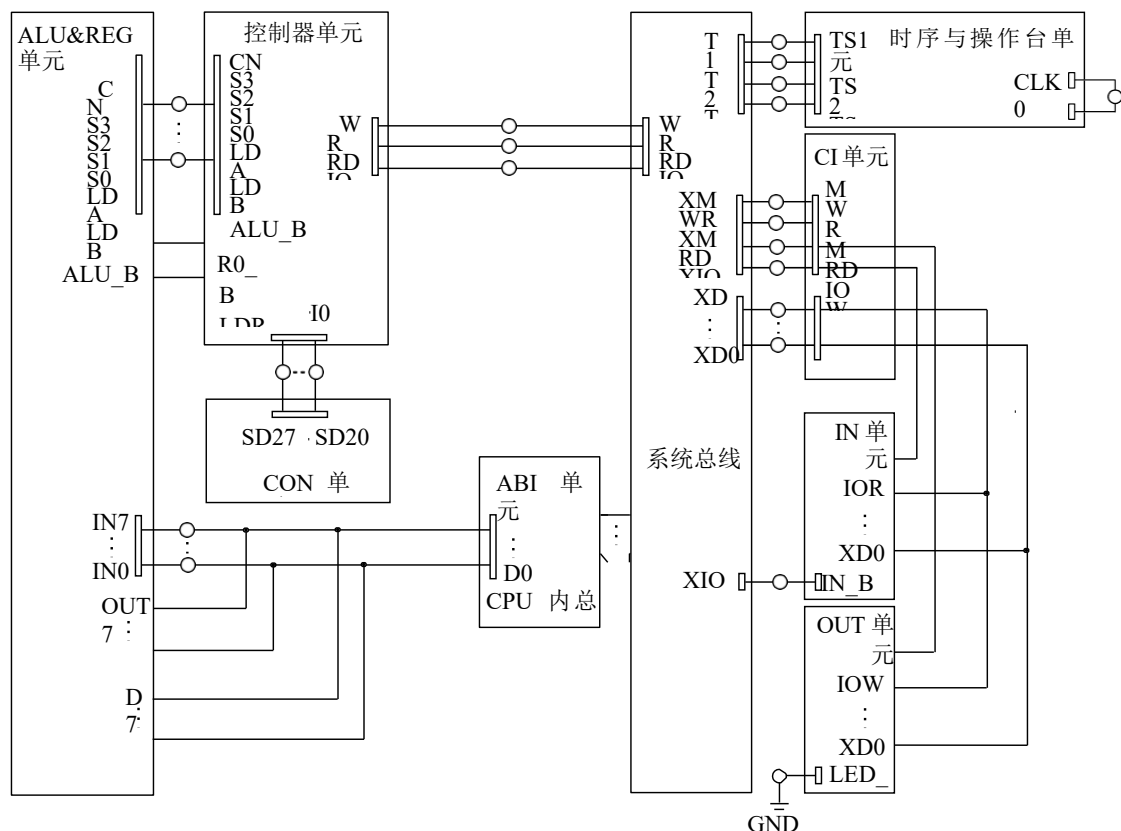


图 3-2-10 实验接线图

2. 对微控器进行读写操作，分两种情况：手动读写和联机读写。

### 1) 手动读写

#### (1) 手动对微控器进行编程（写）

① 将时序与操作台单元的开关 KK1 置为‘停止’档, KK3 置为‘编程’档, KK4 置为‘控存’档, KK5 置为‘置数’档。

② 使用 CON 单元的 SD15——SD10 给出微地址, IN 单元给出低 8 位应写入的数据, 连续两次按动时序与操作台的开关 ST, 将 IN 单元的数据写到该单元的低 8 位。

③ 将时序与操作台单元的开关 KK5 置为‘加 1’档。

④ IN 单元给出中 8 位应写入的数据, 连续两次按动时序与操作台的开关 ST, 将 IN 单元的数据写到该单元的中 8 位。IN 单元给出高 8 位应写入的数据, 连续两次按动时序与操作台的开关 ST, 将 IN 单元的数据写到该单元的高 8 位。

⑤ 重复①、②、③、④四步, 将表 3-2-2 的微代码写入 E2ROM 芯片中。

#### (2) 手动对微控器进行校验（读）

① 将时序与操作台单元的开关 KK1 置为‘停止’档, KK3 置为‘校验’档, KK4 置为‘控存’档, KK5 置为‘置数’档。



② 使用 CON 单元的 SD15——SD10 给出微地址，连续两次按动时序与操作台的开关 ST，MC 单元的指数数据指示灯 M7——M0 显示该单元的低 8 位。

③ 将时序与操作台单元的开关 KK5 置为‘加 1’档。

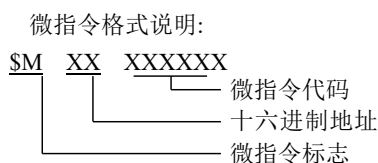
④ 连续两次按动时序与操作台的开关 STMC 单元的指数数据指示灯 M15——M8 显示该单元的中 8 位，MC 单元的指数数据指示灯 M23——M16 显示该单元的高 8 位。

⑤ 重复①、②、③、④四步，完成对微代码的校验。如果校验出微代码写入错误，重新写入、校验，直至确认微指令的输入无误为止。

## 2) 联机读写

### (1) 将微程序写入文件

联机软件提供了微程序下载功能，以代替手动读写微控器，但微程序得以指定的格式写入到以 TXT 为后缀的文件中，微程序的格式如下：



如\$M 1F 112233，表示微指令的地址为 1FH，微指令值为 11H（高）22H（中）33H（低）本次实验的微程序如下，其中分号‘；’为注释符，分号后面的内容在下载时将被忽略掉。

### (2) 写入微程序

用联机软件的“【转储】—【装载】”功能将该格式(\*.TXT)文件装载入实验系统。装入过程中，在软件的输出区的‘结果’栏会显示装载信息，如当前正在装载的是机器指令还是微指令，还剩多少条指令等。

### (3) 校验微程序

选择联机软件的“【转储】—【刷新指令区】”可以读出下位机所有的机器指令和微指令，并在指令区显示。检查微控器相应地址单元的数据是否和表 3-2-2 中的十六进制数据相同，如果不同，则说明写入操作失败，应重新写入，可以通过联机软件单独修改某个单元的微指令，先用鼠标左键单击指令区的‘微存’TAB 按钮，然后再单击需修改单元的数据，此时该单元变为编辑框，输入 6 位数据并回车，编辑框消失，并以红色显示写入的数据。

```
; //***** //
; // //
; //          微控器实验指令文件 //
; // //
; //          By TangDu CO.,LTD //
; // //
; //***** //
; //**** Start Of MicroController Data **** //
$M 00 000001 ; NOP
$M 01 007070 ; CON(INS)->IR, P<1>
$M 04 002405 ; R0->B
$M 05 04B201 ; A加 B->R0
$M 30 001404 ; R0->A
```





```
$M 32 183001 ; IN->R0
$M 33 280401 ; R0->OUT
$M 35 000035 ; NOP
; //***** End Of MicroController Data ***** //
```

### 3. 运行微程序

本实验支持两种方式运行：本机运行（不需电脑）和联机运行（需要电脑）其中联机运行方式支持数据通路图的观测，支持微程序流程图的观测，支持信号时序图的观测。

转储完成截图如3.2.2所示：

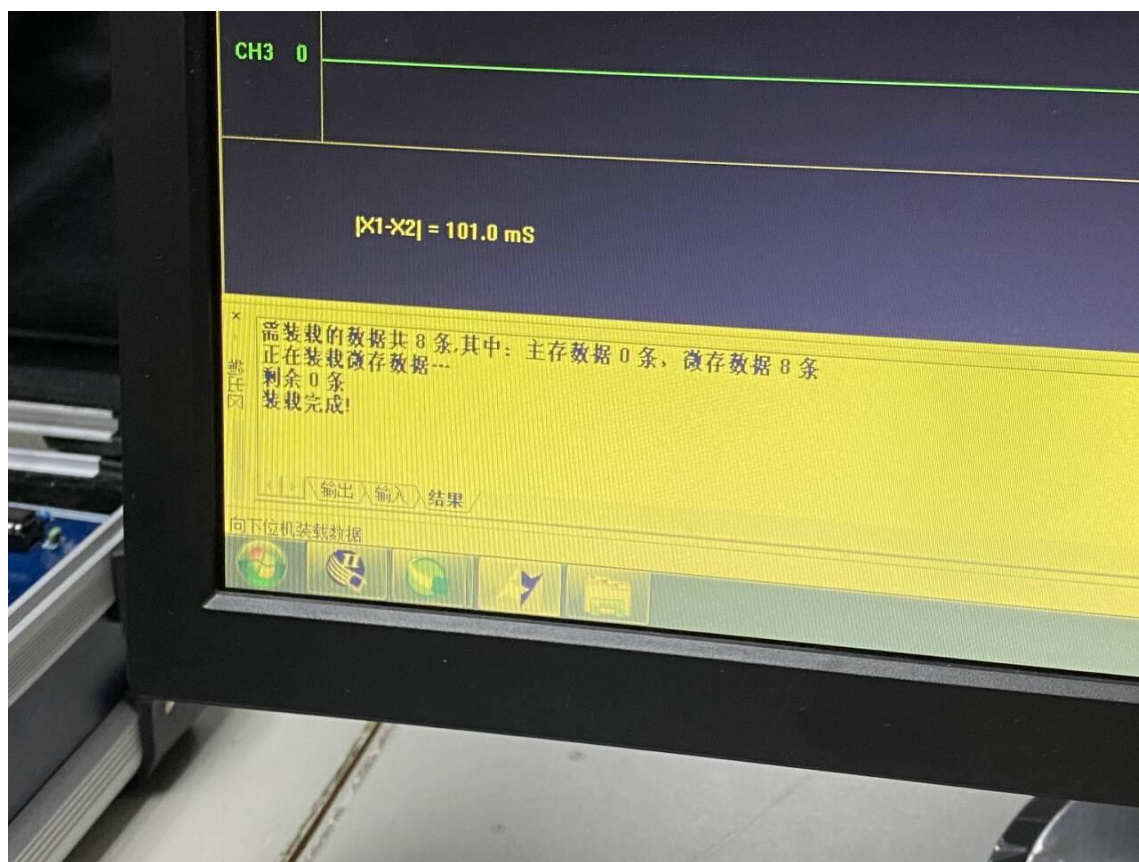


图 3.2.2

#### 1) 本机运行

① 将时序与操作台单元的开关 KK1、KK3 置为‘运行’档，按动 CON 单元的 CLR 按钮，将微地址寄存器（MAR）清零，同时也将指令寄存器（IR）、ALU 单元的暂存器 A 和暂存器 B 清零。

② 将时序与操作台单元的开关 KK2 置为‘单拍’档，然后按动 ST 按钮，体会系统在 T1、T2、T3、T4 节拍中各做的工作。T2 节拍微控器将后续微地址（下条执行的微指令的地址）打入微地址寄存器，当前微指令打入微指令寄存器，并产生执行部件相应的控制信号；T3、T4 节拍根据 T2 节拍产生的控制信号做出相应的执行动作，如果测试位有效，还要根据机器指令及当

前微地址寄存器中的内容进行译码，使微程序转入相应的微地址入口，实现微程序的分支。

③ 按动 CON 单元的 CLR 按钮，清微地址寄存器（MAR）等，并将时序与单元的开关 KK2 置为‘单拍’档。

④ 置 IN 单元数据为 00100011，按动 ST 按钮，如图3.2.3所示：

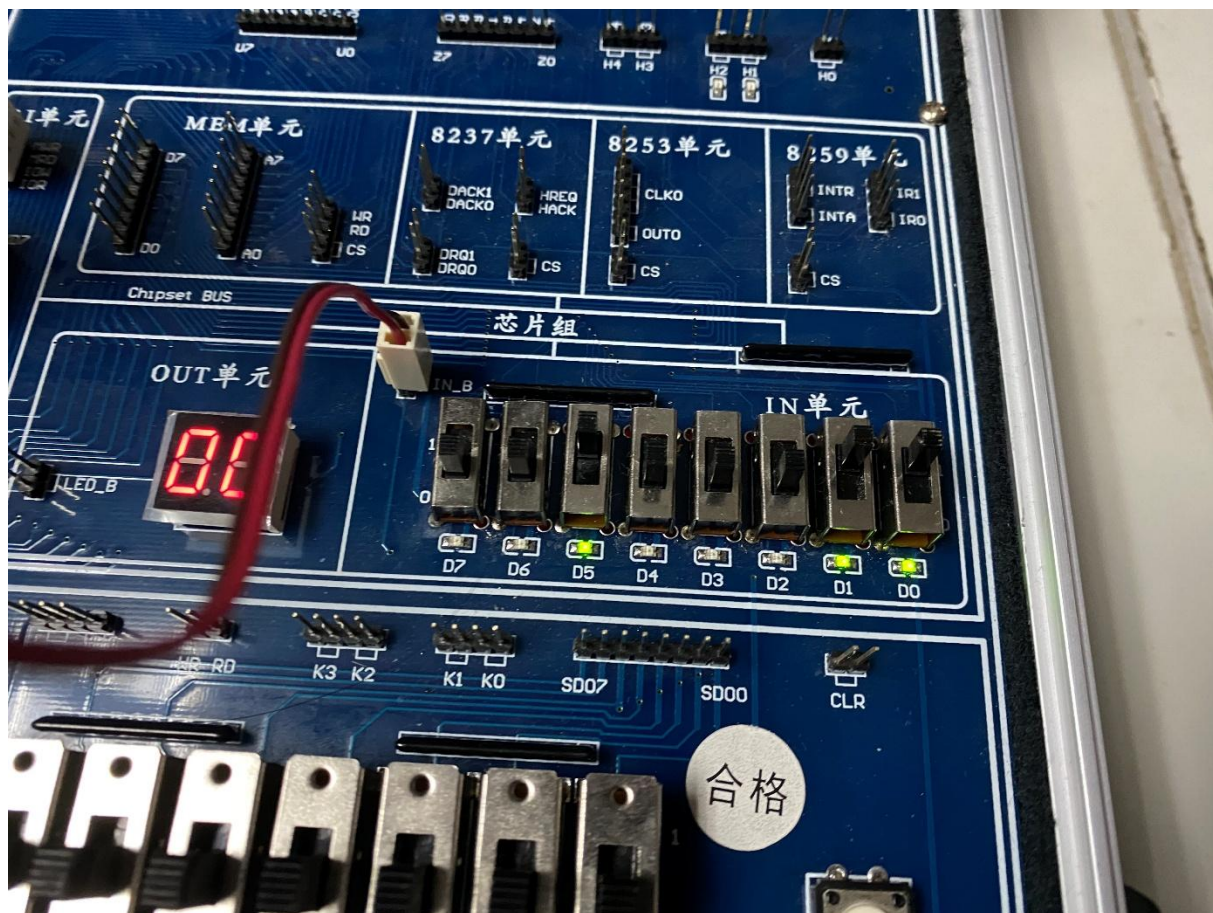


图 3.2.3

当 MC 单元后续微地址显示为 000001 时，在 CON 单元的 SD27...SD20 模拟给出 IN 指令 00100000 并继续单步执行。

结果如图3.2.4所示：

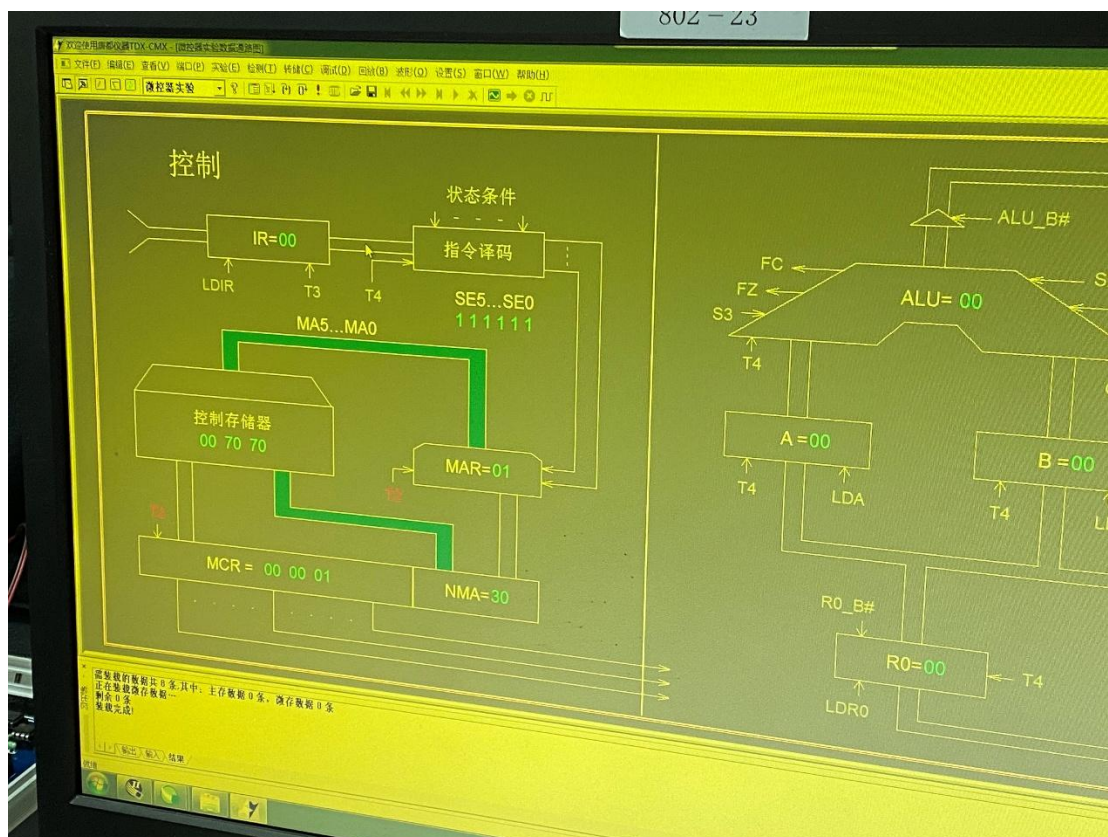


图 3.2.4

当 MC 单元后续微地址显示为 000001 时,说明当前指令已执行完;在 CON 单元的 SD27...SD20 给出 ADD 指令 00000000,该指令将会在下个 T3 被打入指令寄存器 (IR),它将 R0 中的数据和其自身相加后送 R0;接下来在 CON 单元的 SD27...SD20 给出 OUT 指令 00110000 并继续单步执行,如图 3.2.5 所示:



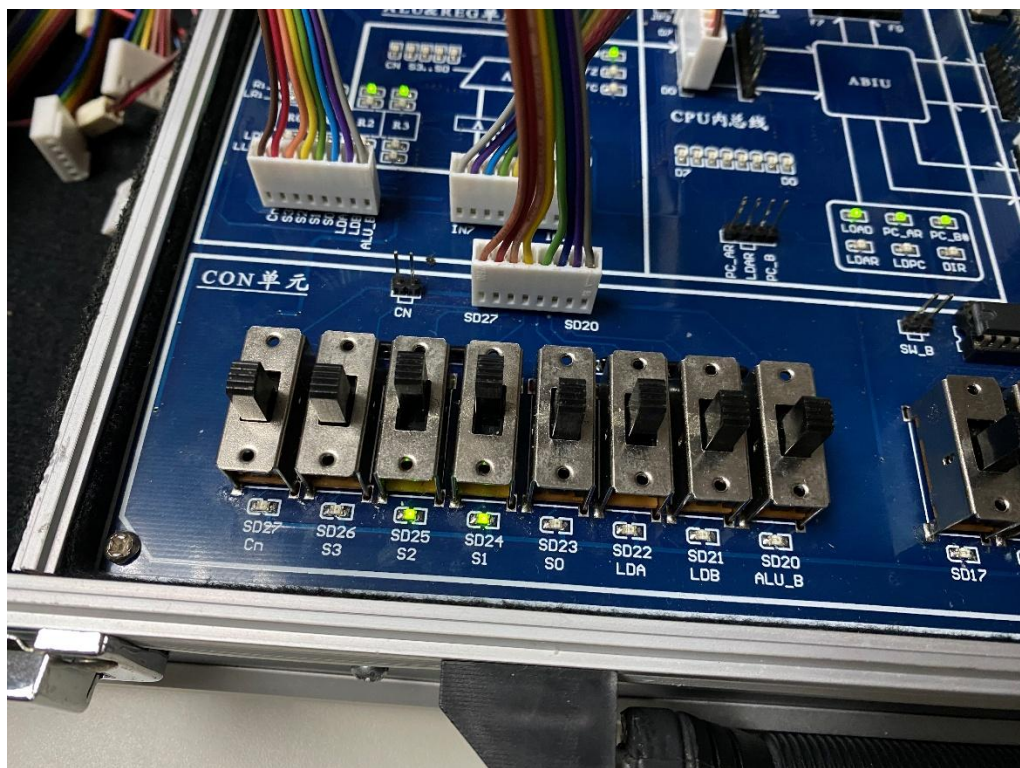


图 3.2.5

在 MC 单元后续微地址显示为 000001 时，观察 OUT 单元的显示值是否为 01000110。结果如图3.2.6， 3.2.7所示：

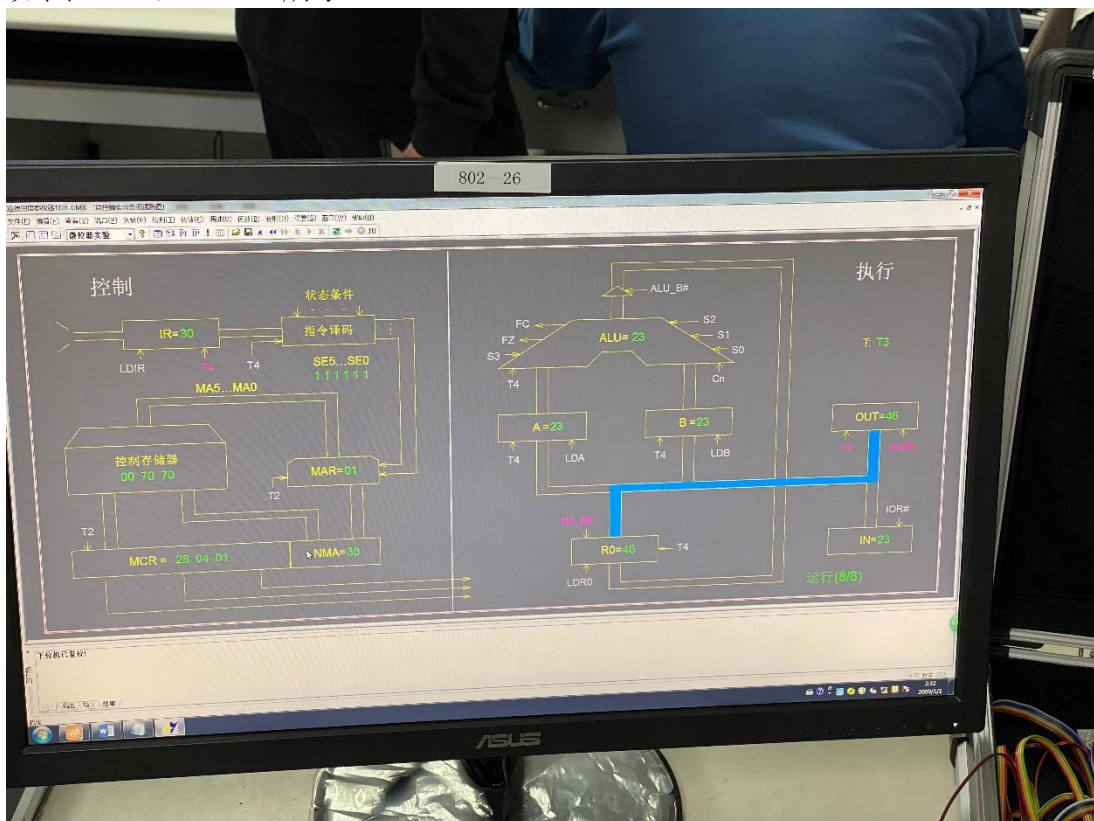


图 3.2.6



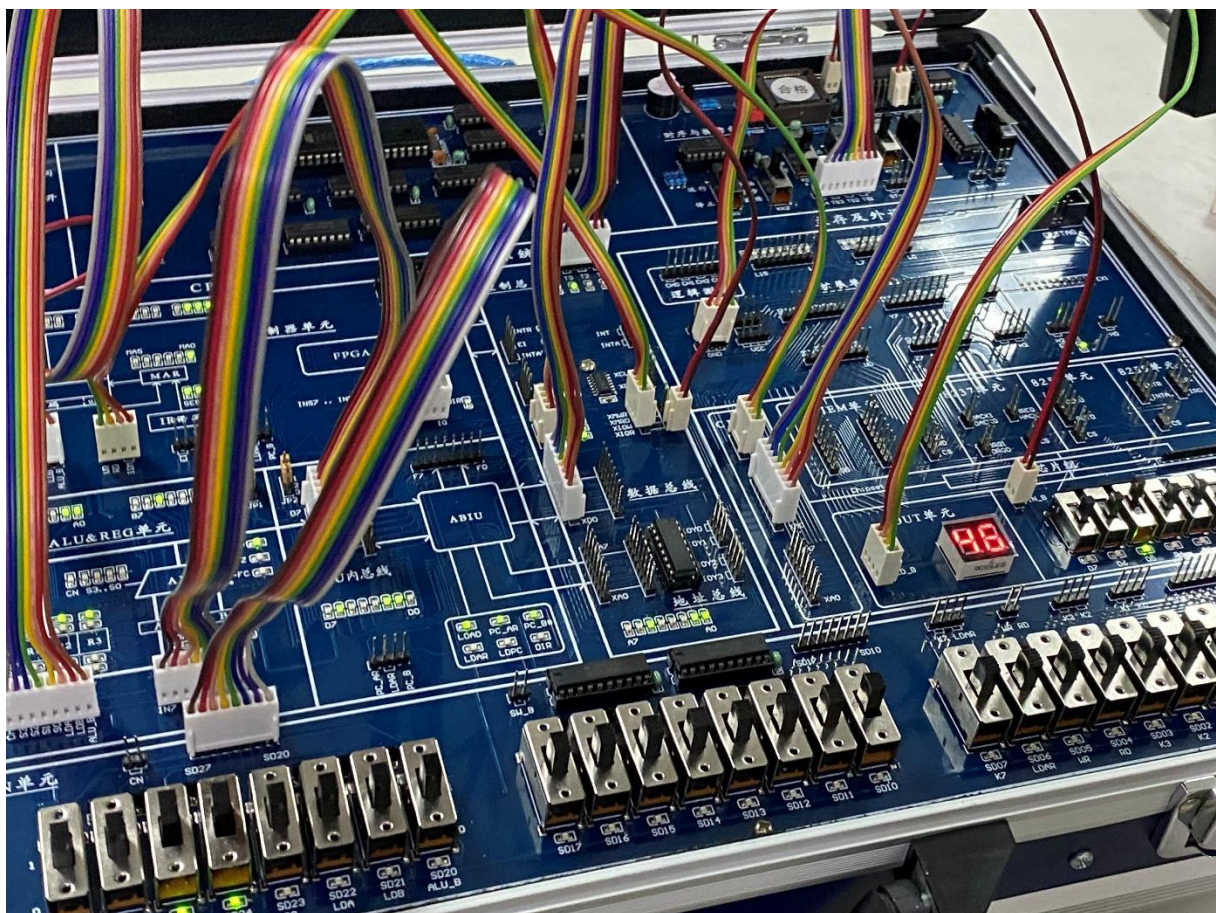


图 3.2.7

## 2) 联机运行

### (1) 观测数据通路图

打开 TDX-CMX 软件, 在菜单上选择【实验】—【微控器实验】, 打开本实验的数据通路图, 也可以通过工具栏上的下拉框打开数据通路图, 数据通路图如图 3-2-8 所示。


操作方法同本机运行, 仔细观察每条机器指令的执行过程, 体会后续微地址被强置转换的过程, 这是计算机识别和执行指令的根基。

按本机运行的顺序给出数据和指令, 观察最后的运算结果是否正确。

### (2) 观测微程序流图

打开数据通路图后, 点击“【调试】—【微程序流图】”, 打开微程序流程图, 如图 3-2-9 所示, 操作方法同本机运行, 跟踪显示每条机器指令的执行过程。

### (3) 观测信号时序图

点击  打开选择观察信号窗口, 或者选择联机软件的【调试】—【时序观测窗】选





择想要观察的信号，如图 3-2-11，点击确定。

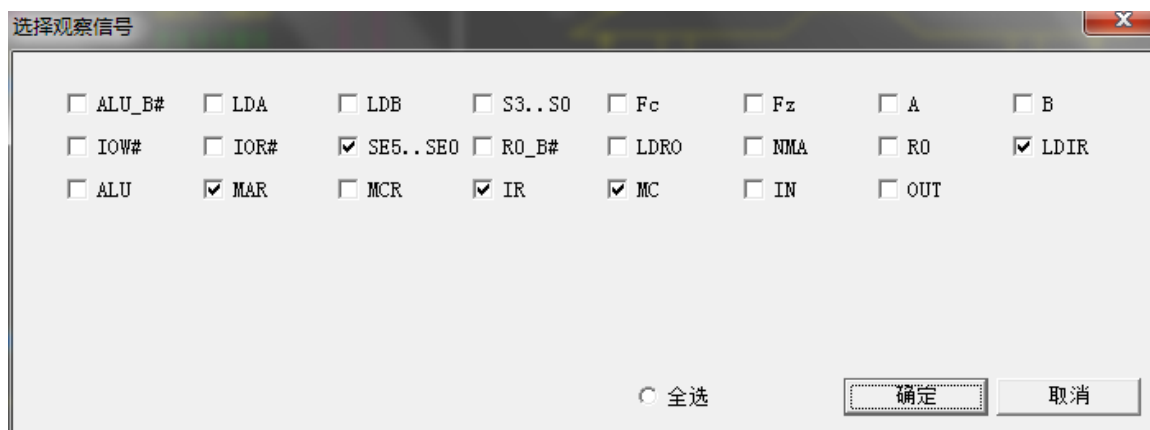


图 3-2-11 选择观察信号

弹出时序观测窗，如图 3-2-12 所示。

按动CON 单元的 CLR 按钮，将时序与操作台单元的开关 KK2 置为‘单拍’档，然后按动4 次ST 按钮，此时序观测窗口上方 T 代表节拍，从左到右依次为 T1、T2、T3、T4，观察 T2时刻上升沿时，NMA 中地址 01H 送入 MAR，MC 显示 MAR 地址对应的值。

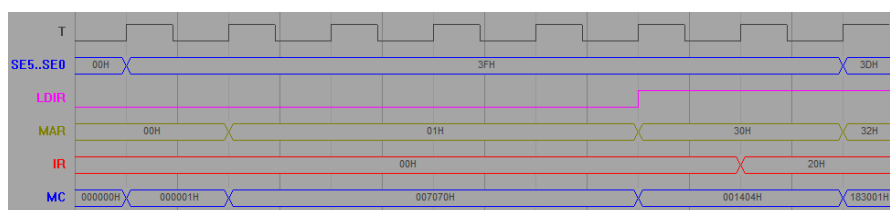


图 3-2-12 时序观测图

再按动 4 次ST 按钮，观察第二个机器周期 T2 时刻上升沿时，NMA 中地址 30H 送入 MAR，MC 显示MAR 地址对应的值，MCR 为上一机器周期执行的微指令 007070H，该微指令使 LDIR 有效；T3 时刻上升沿 CON 单元SD27~SD20 输入的机器指令 20H 被打入指令寄存器 IR；T4 时刻上升沿，机器指令译码 SE5...SE0 变更为 3DH，所以 MAR 变为地址 32H，将要执行的微指令MC 变为183001H。由此可知第二个机器周期执行了微指令 007070H ，完成了机器指令 20H 的译码。再按动 4 次ST 按钮，观察 IN 单元数据被打入寄存器 R0 中。

结果如图3.2.8所示：

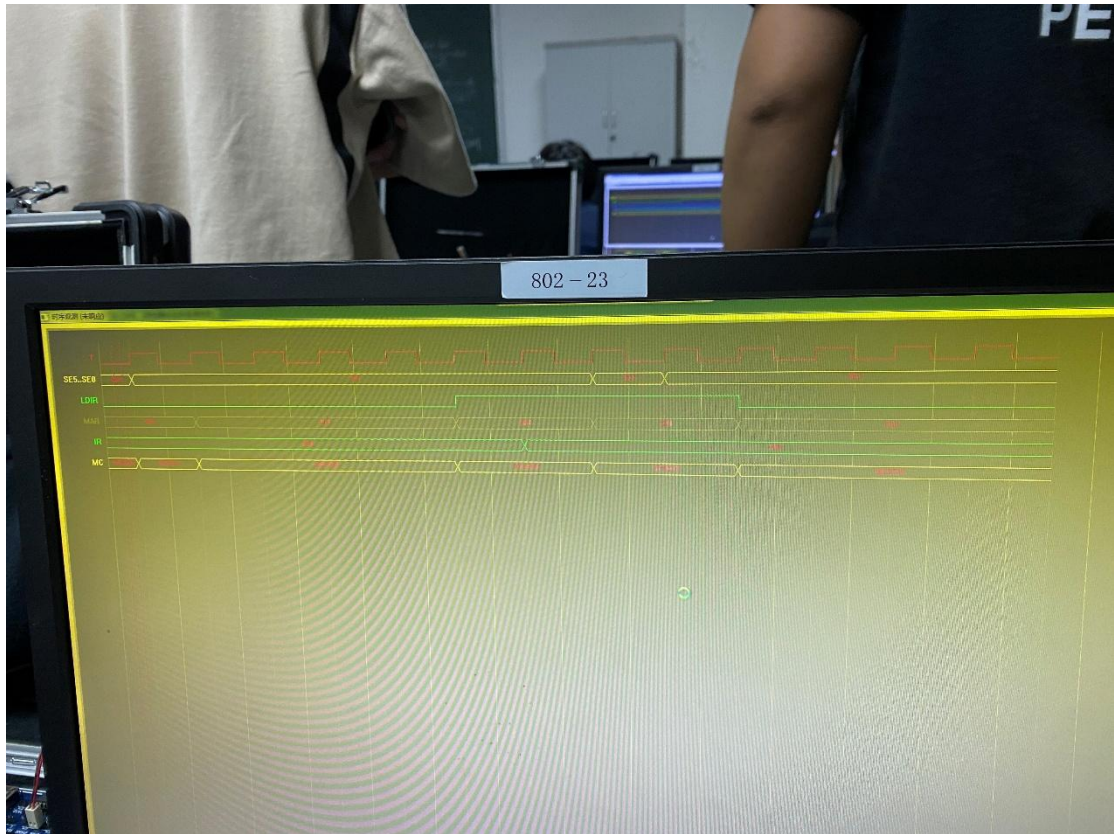


图 3.2.8

### 3 实验小结

在本次实验中，我们深入探讨了时序发生器的原理与设计方法，进一步了解了 FPGA 的应用设计及 EDA 软件的使用过程。通过实验，我掌握了微程序控制器的基本组成结构，包括如何设计和配置时序控制单元，使得微程序控制器能够有效地执行指定的操作。

实验的关键在于熟练掌握微程序的编制与写入过程。在编写微程序后，我们将其写入控制器，并通过观察微程序的运行过程和数据流动加深了对其工作的理解。此外，基于数据通路图的观察，我们进一步理解了微程序控制器如何通过控制信号来协调各个单元的操作步骤，确保指令的顺利执行。这一过程让我理解到微程序控制器的实际工作原理，即如何通过信号传输和时序控制实现复杂的数据处理。

通过分析微程序流程图，我学习了微程序控制器的操作原理，了解了其信号传递的逻辑和作用。在实验过程中，我们还通过信号时序图观察了控制器的工作状态，对微程序控制器如何根据时序控制信号完成数据处理有了更直观的理解。这种基于信号的观察方式让我更加清晰地看到微程序控制器各模块之间的信号交互和控制逻辑，使我对控制器的时序机制有了深入的认识。

---

总体而言，通过这次实验，我进一步掌握了微程序控制器的设计、编程和运行原理，并熟悉了 FPGA 在硬件设计中的应用。实验加深了我对微程序控制器各项原理的理解，特别是在数据通路和信号控制方面的操作，使我对其在电子系统中的实际应用有了更为全面的认识。