

## 课前复习 | 第11、12周开放时间

- 05月07日 周二晚 18:30 – 20:50
- 05月09日 周四晚 18:30 – 20:50
- 05月14日 周二晚 18:30 – 20:50
- 05月16日 周四晚 18:30 – 20:50

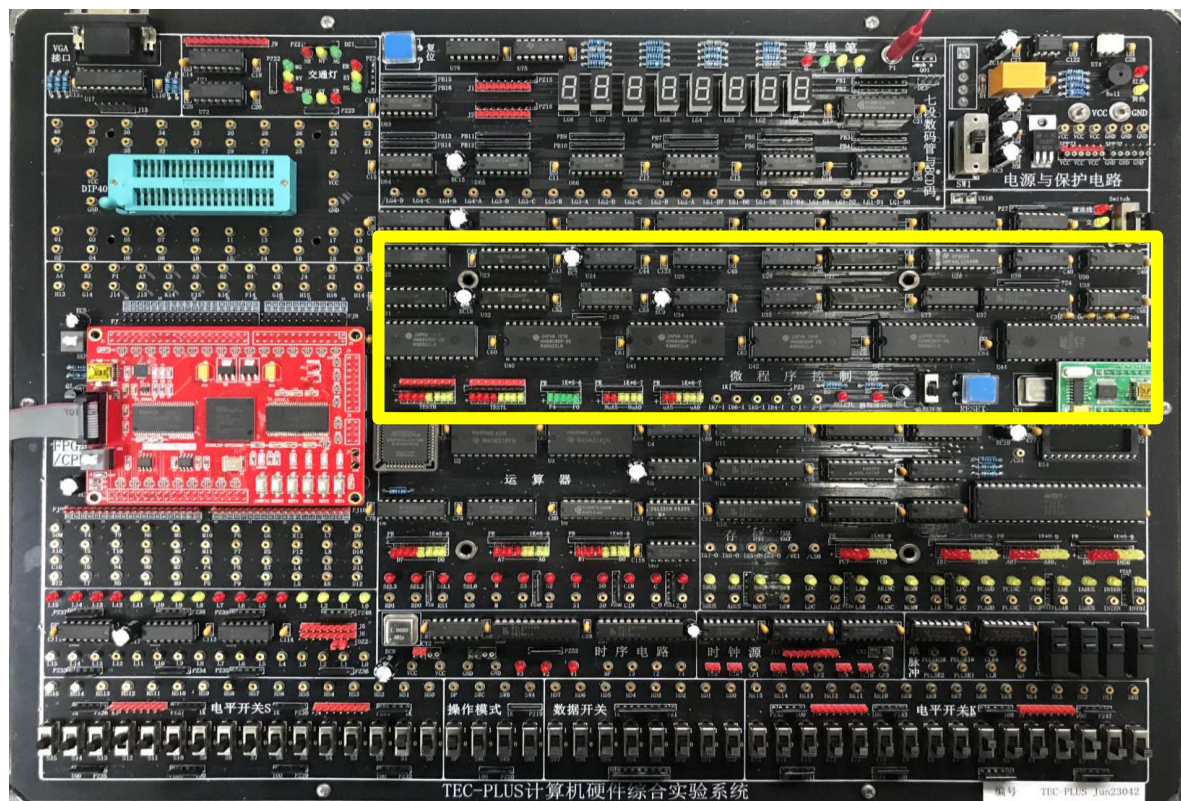
# 实验四



## 微程序控制器组成

# 实验四

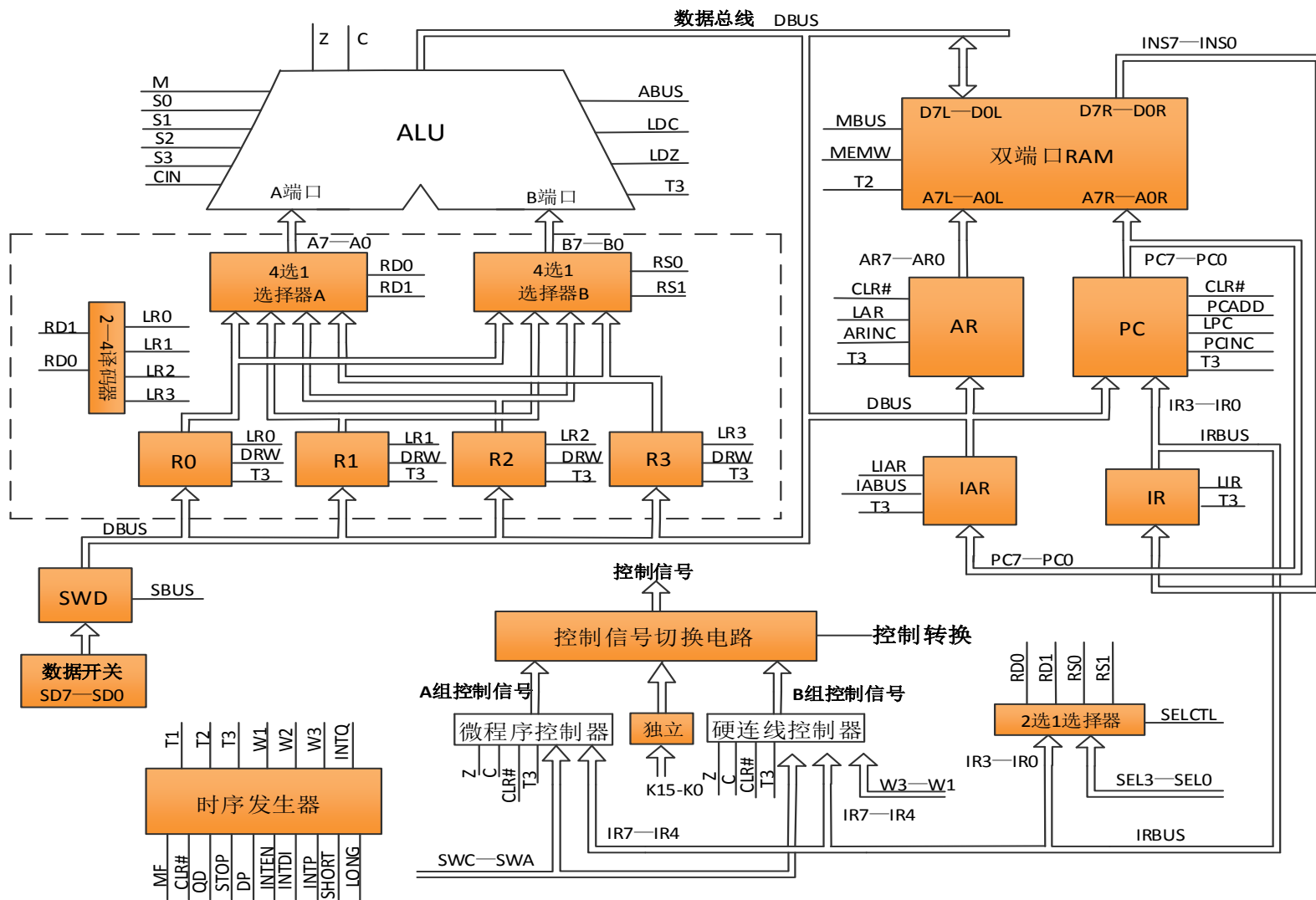
- 实验目的
- 实验电路
- 实验任务
- 实验步骤
- 实验要求



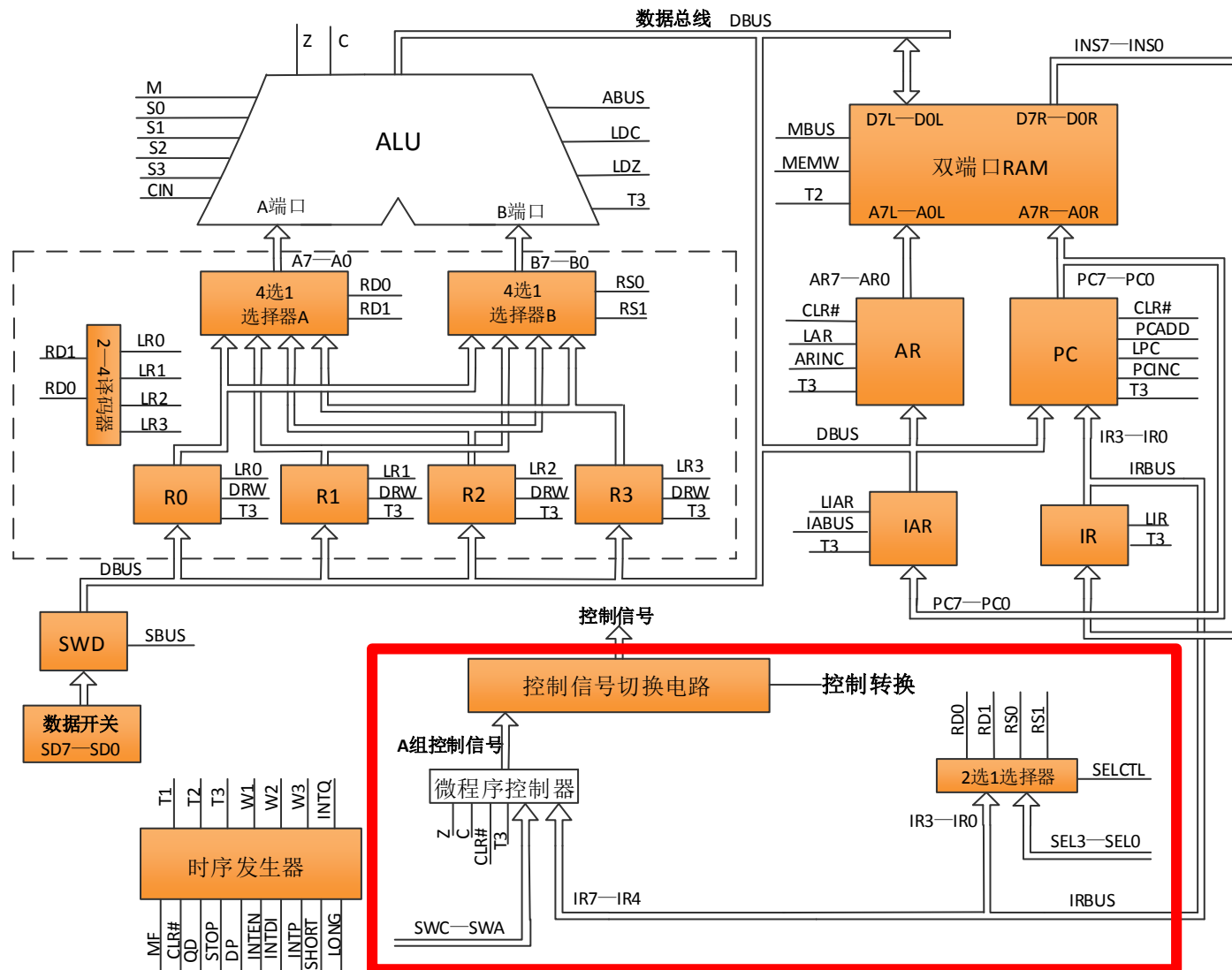
## 实验四 | 实验目的

- ① 掌握微程序控制器的原理；
- ② 掌握TEC-8模型计算机中微程序控制器的实现方法，尤其是微地址转移逻辑的实现方法；
- ③ 理解条件转移对计算机的重要性；

# 实验四 | 实验电路>>概览

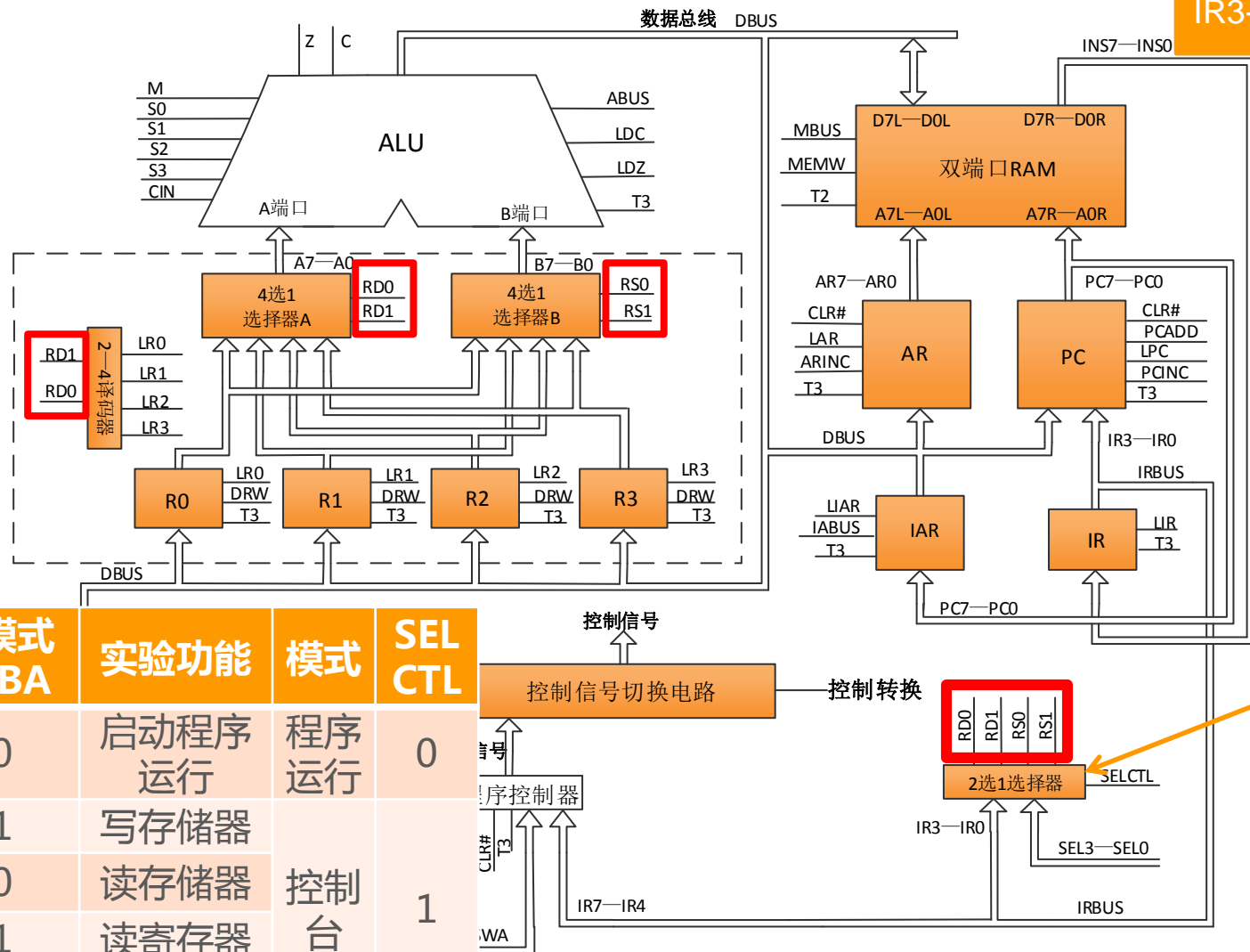


# 实验四 | 实验电路>>概览



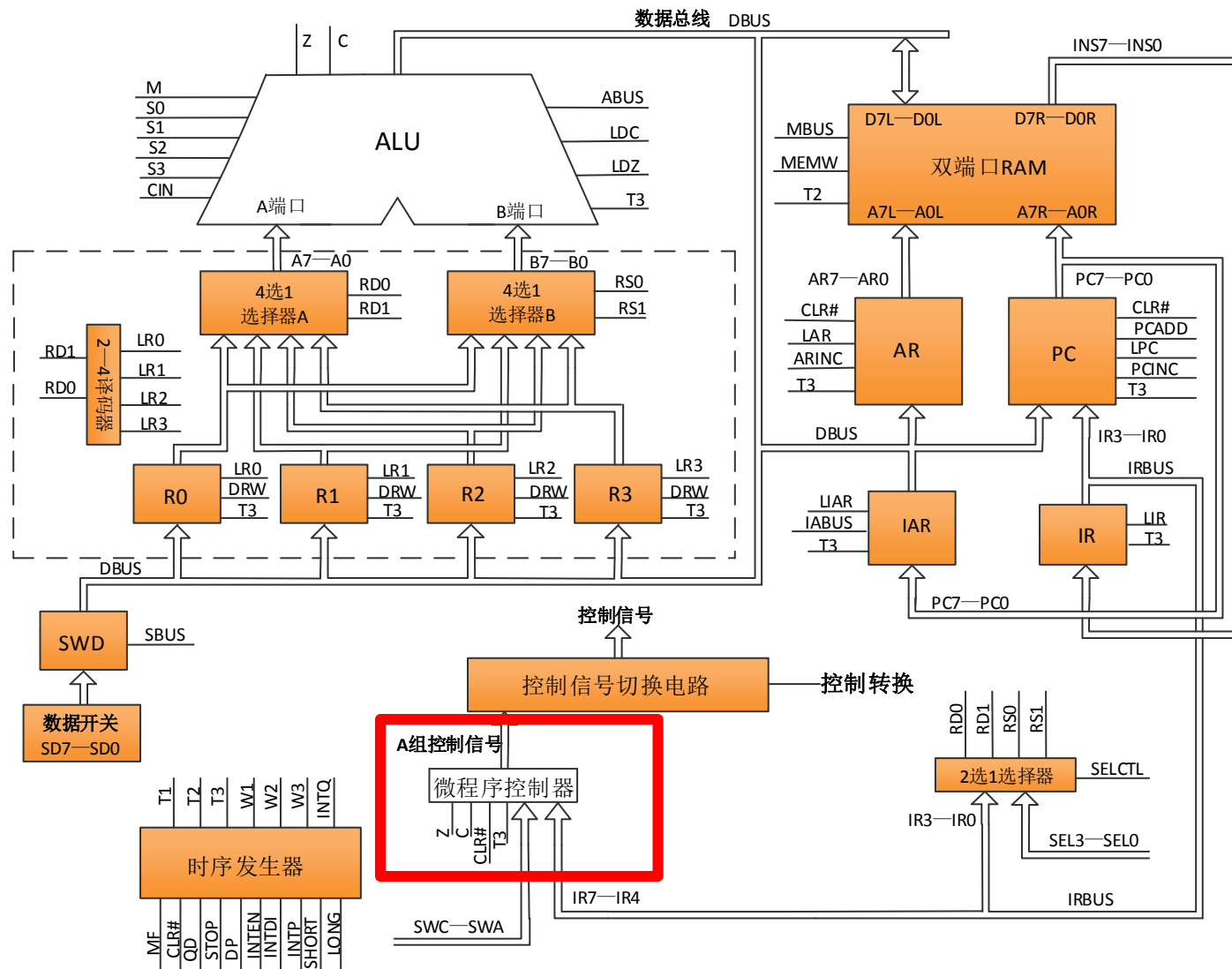
# 实验四 | 实验电路 >> 2选1选择器

**2选1选择器:**  
当SELCTL=1时选中  
SEL3-SEL0;当  
SELCTL=0时, 选中  
IR3-IR0;



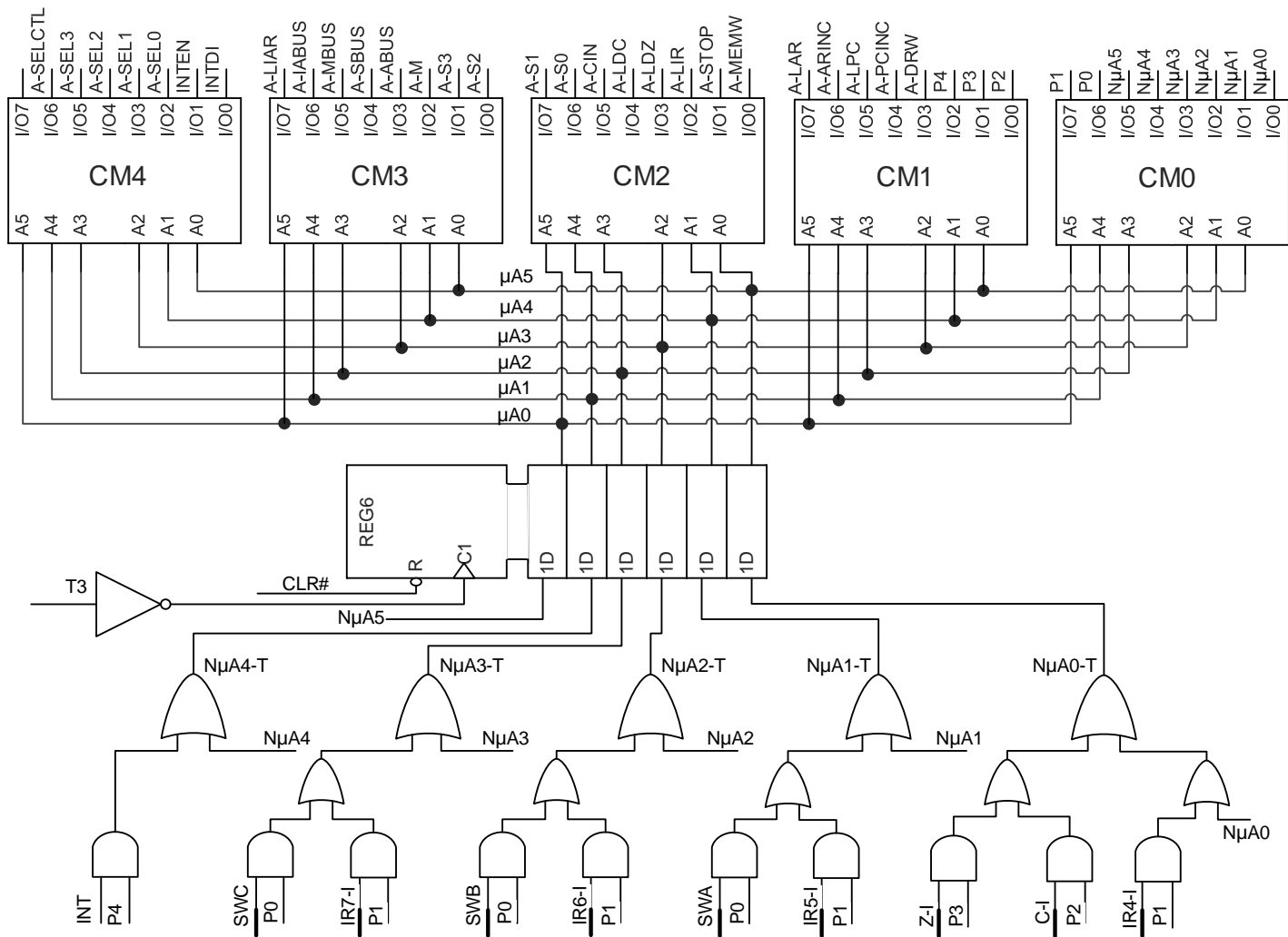
操作模式 SWCBA	实验功能	模式	SEL CTL
000	启动程序 运行	程序 运行	0
001	写存储器	控制 台	1
010	读存储器		
011	读寄存器		
100	写寄存器		

# 实验四 | 实验电路 >> 微程序控制器

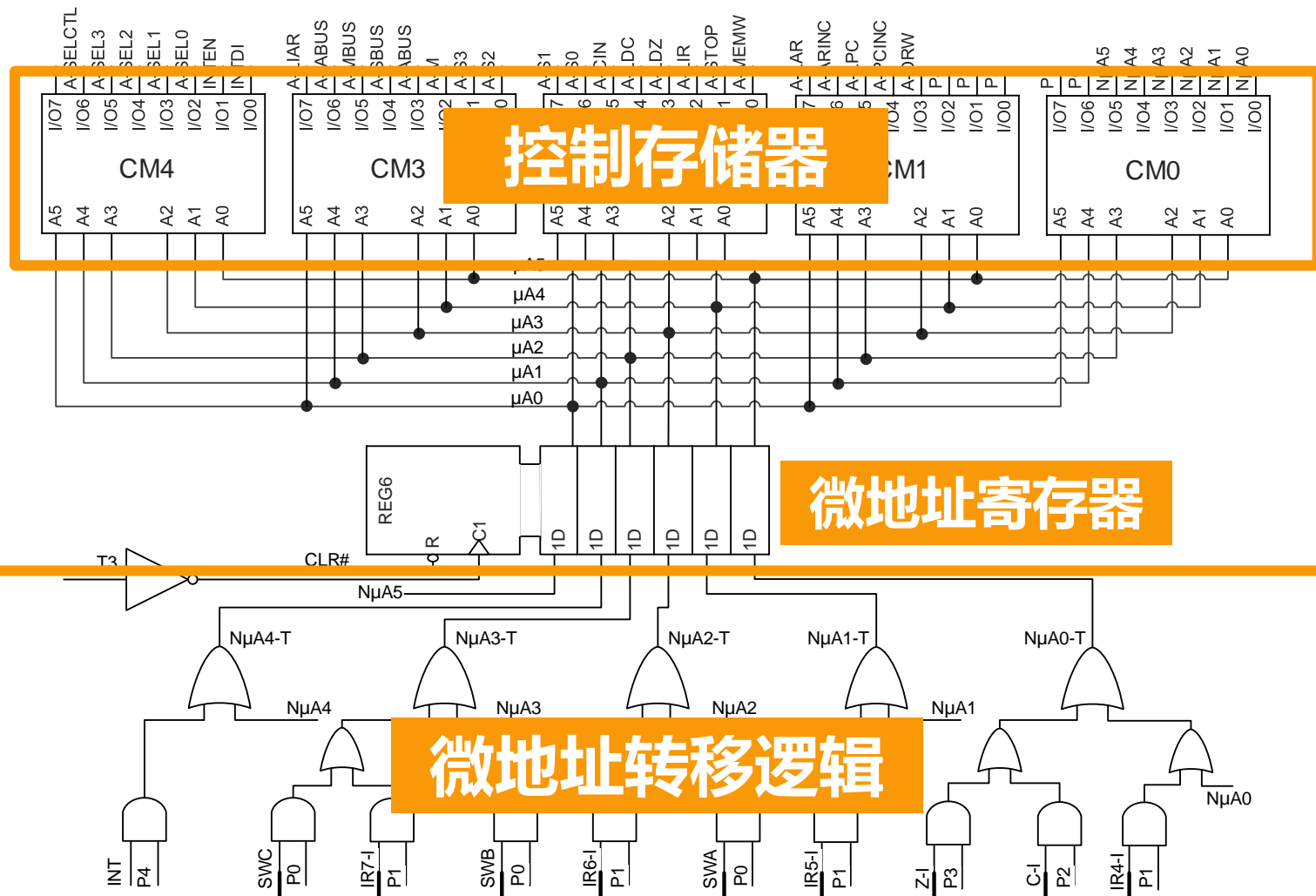




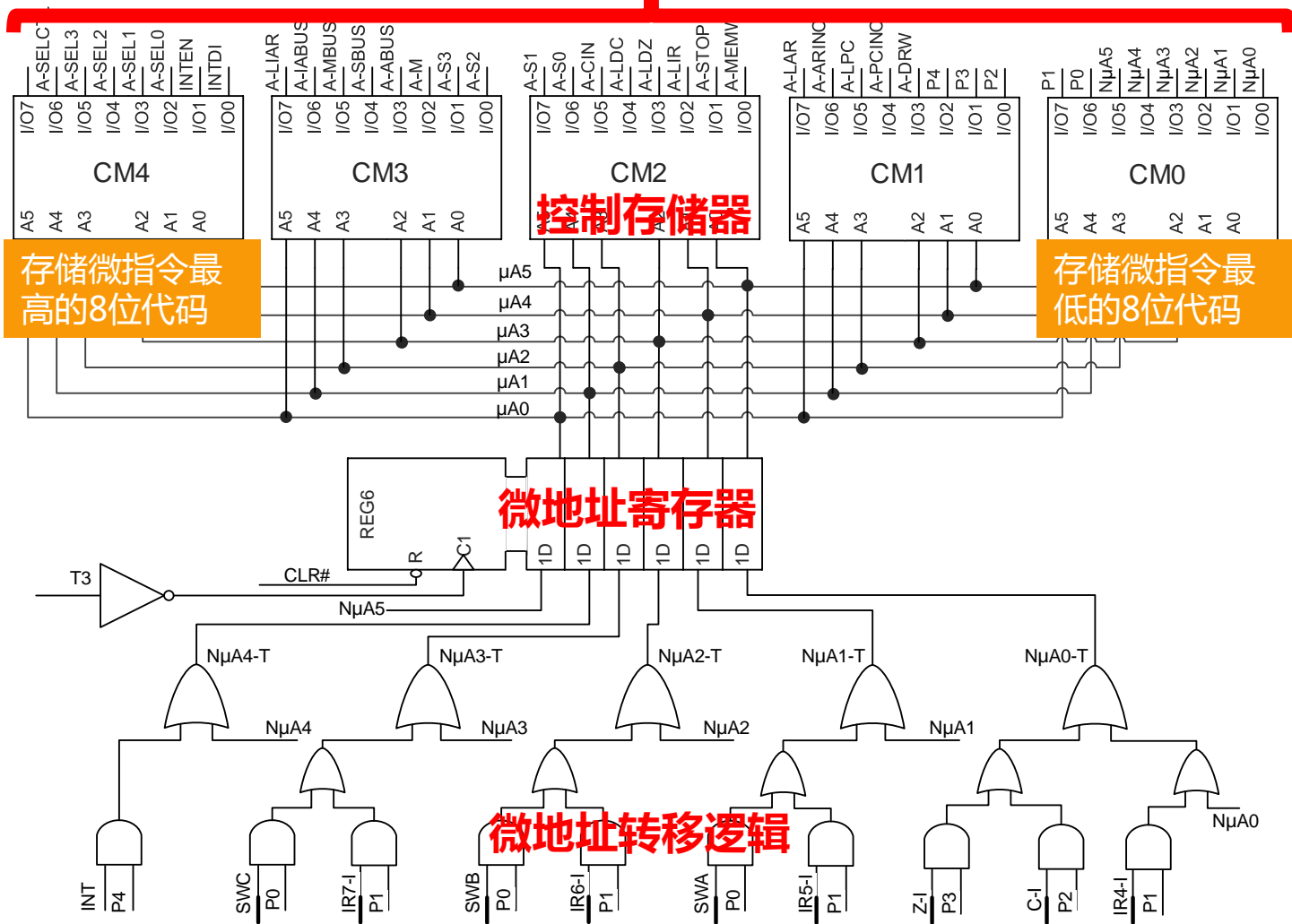
# 实验四 | 实验电路 >> 微程序控制器电路



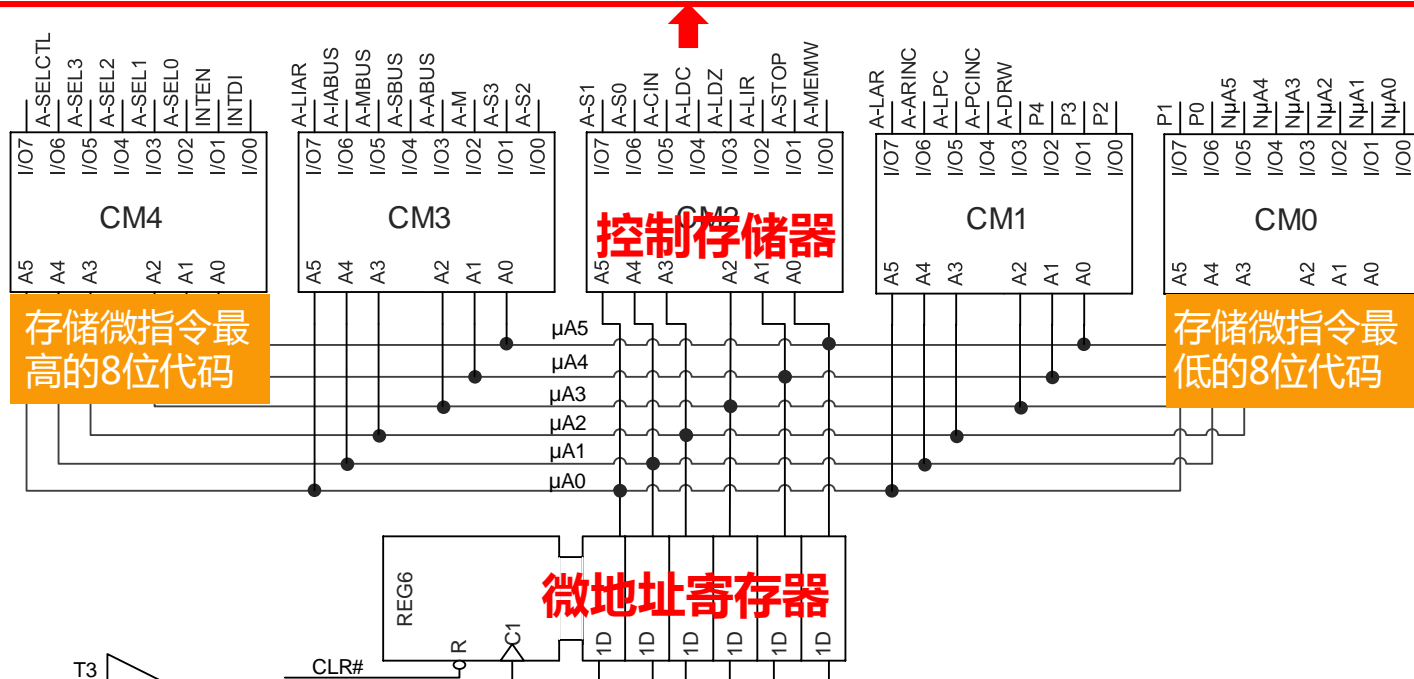
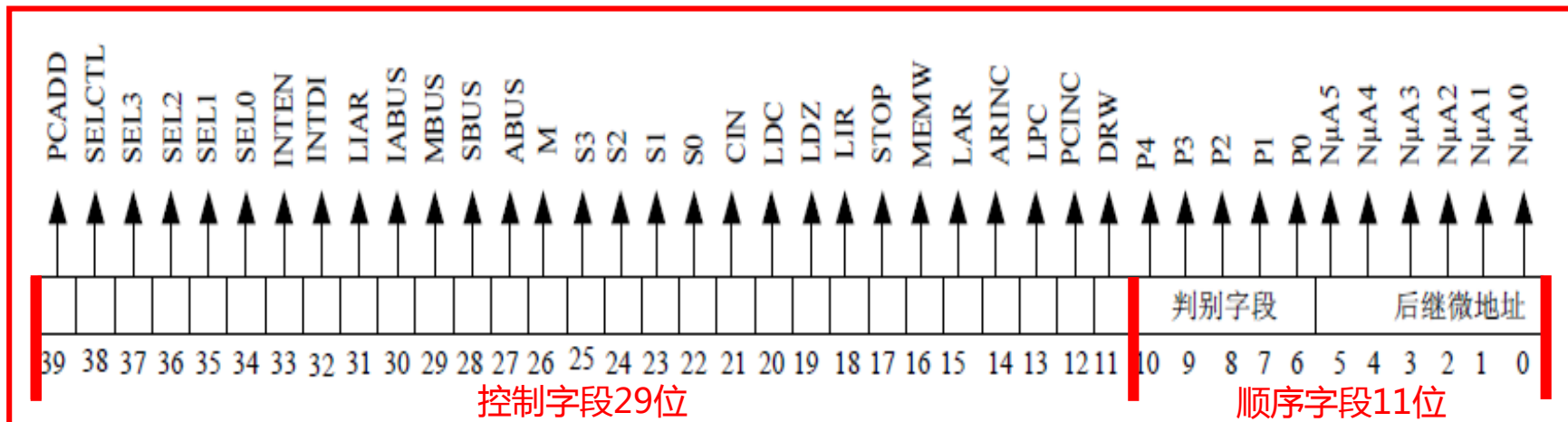
# 实验四 | 实验电路 >> 微程序控制器电路



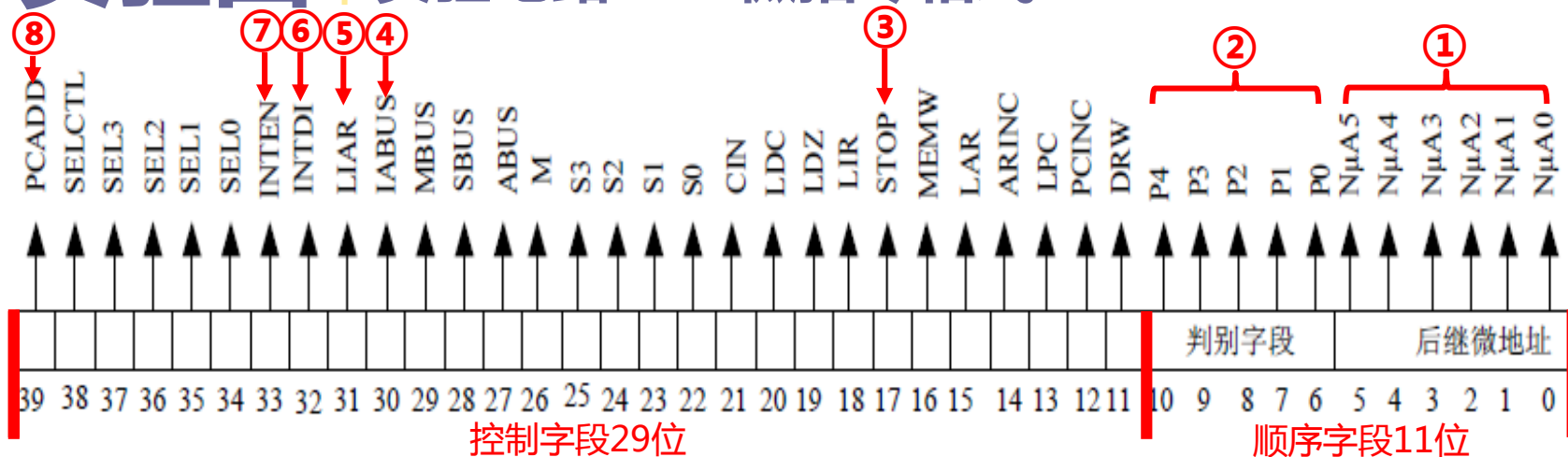
# 实验四 | 实验电路 > 40位微指令控制器电路



# 实验四 | 实验电路 >> 微指令格式

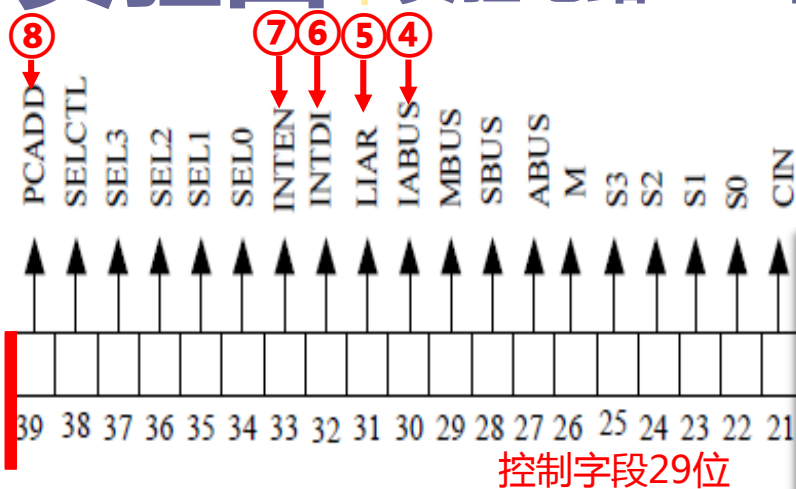


# 实验四 | 实验电路 >> 微指令格式



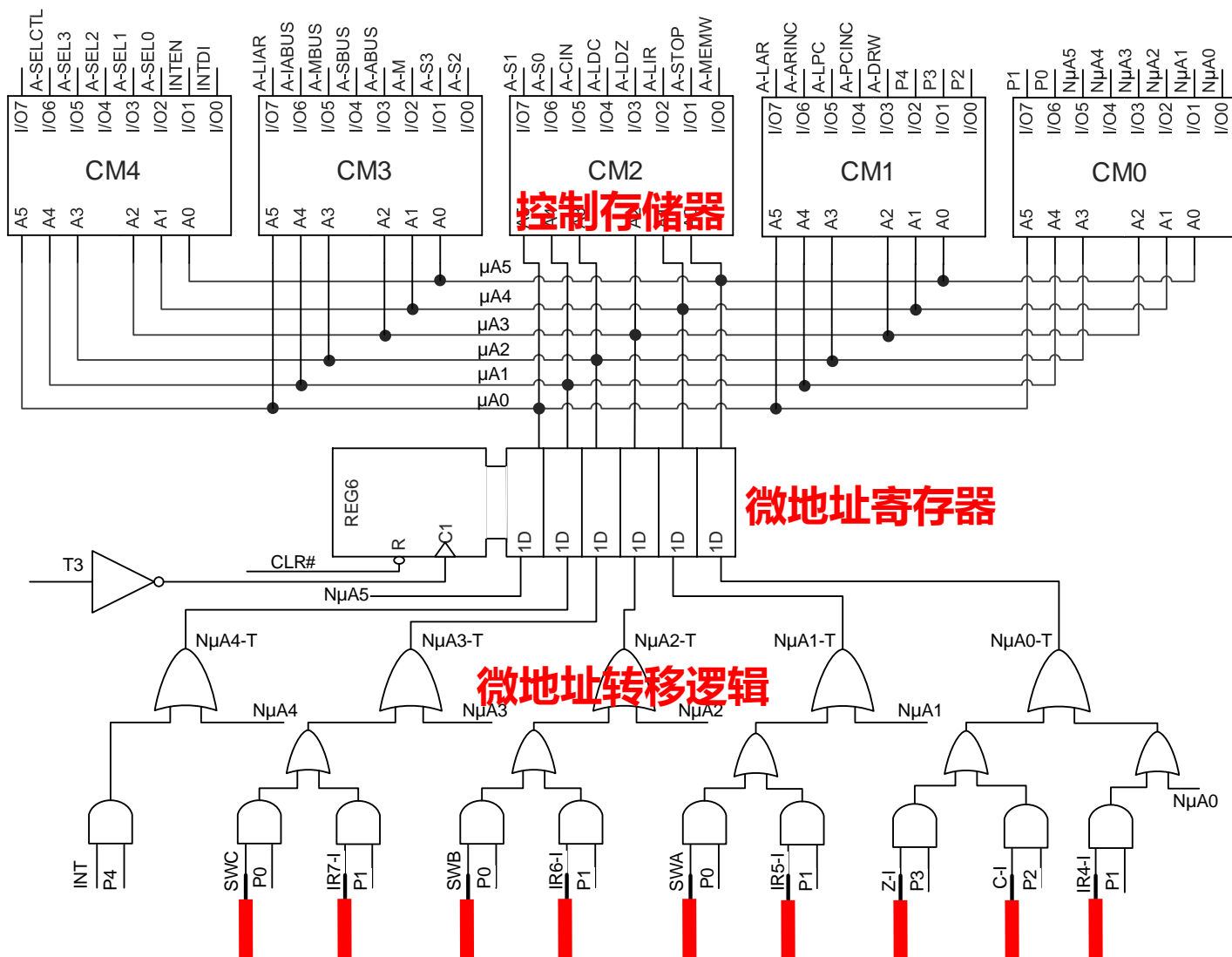
序号	字段	解释
1	NμA5~NμA0	下址，在微指令顺序执行的情况下，它是下一条微指令的地址
2	P0	=1时，根据后继微地址NμA5~NμA0和模式开关SWC、SWB、SWA确定下一条微指令的地址。
	P1	=1时，根据后继微地址NμA5~NμA0和指令操作码IR7~IR4确定下一条微指令的地址。
	P2	=1时，根据后继微地址NμA5~NμA0和进位C确定下一条微指令的地址。
	P3	=1时，根据后继微地址NμA5~NμA0和结果为0标志Z确定下一条微指令的地址。
	P4	=1时，根据后继微地址NμA5~NμA0和中断信号INT确定下一条微指令的地址。模型计算机中，中断信号INT由时序发生器在接到中断请求信号后产生。

# 实验四 | 实验电路 >> 微指令



序号	字段	解释
3	STOP	=1时，在T3结束后时序发生器停止输出节拍脉冲T1、T2、T3。
4	IABUS	=1时，将中断地址寄存器中的地址送数据总线DBUS。
5	LIAR	=1时，在T3的上升沿，将PC7~PC0写入中断地址寄存器IAR。
6	INTDI	=1时，置允许中断标志(在时序发生器中)为0，禁止TEC-8模型计算机响应中断请求。
7	INTEN	=1时，置允许中断标志(在时序发生器中)为1，允许TEC-8模型计算机响应中断请求。
8	PCADD	=1时，将当前的PC值加上相对转移量，生成新的PC。

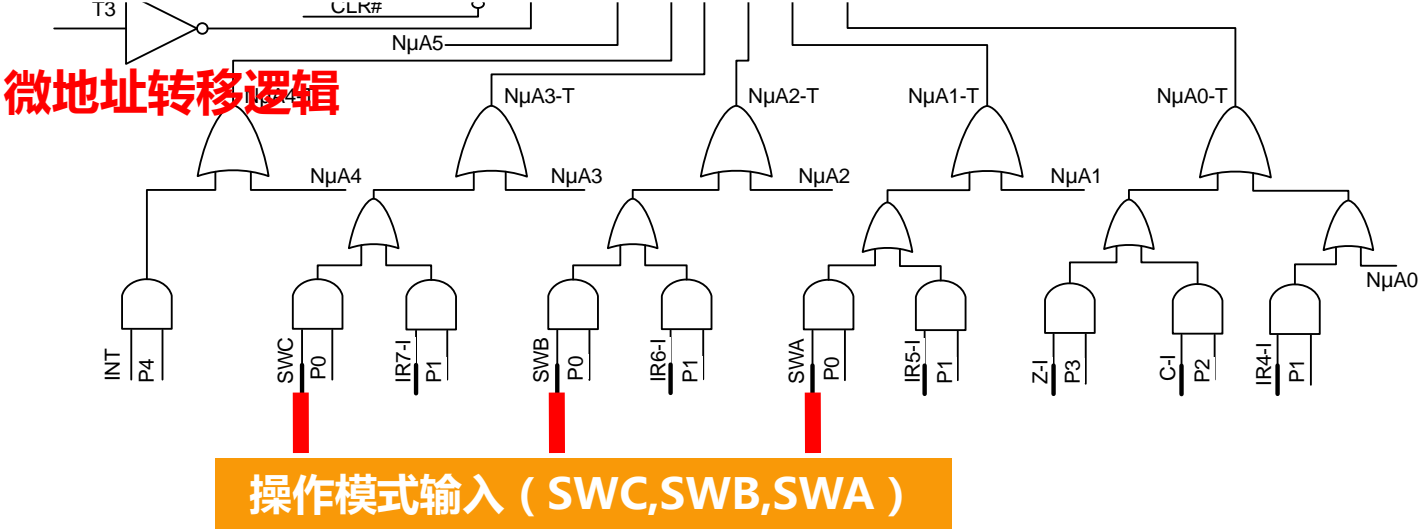
# 实验四 | 实验电路 >> 微程序控制器电路



短粗线标志的信号均有接线孔或开关控制 (SWC/SWB/SWA, IR7~IR4, Z, C)

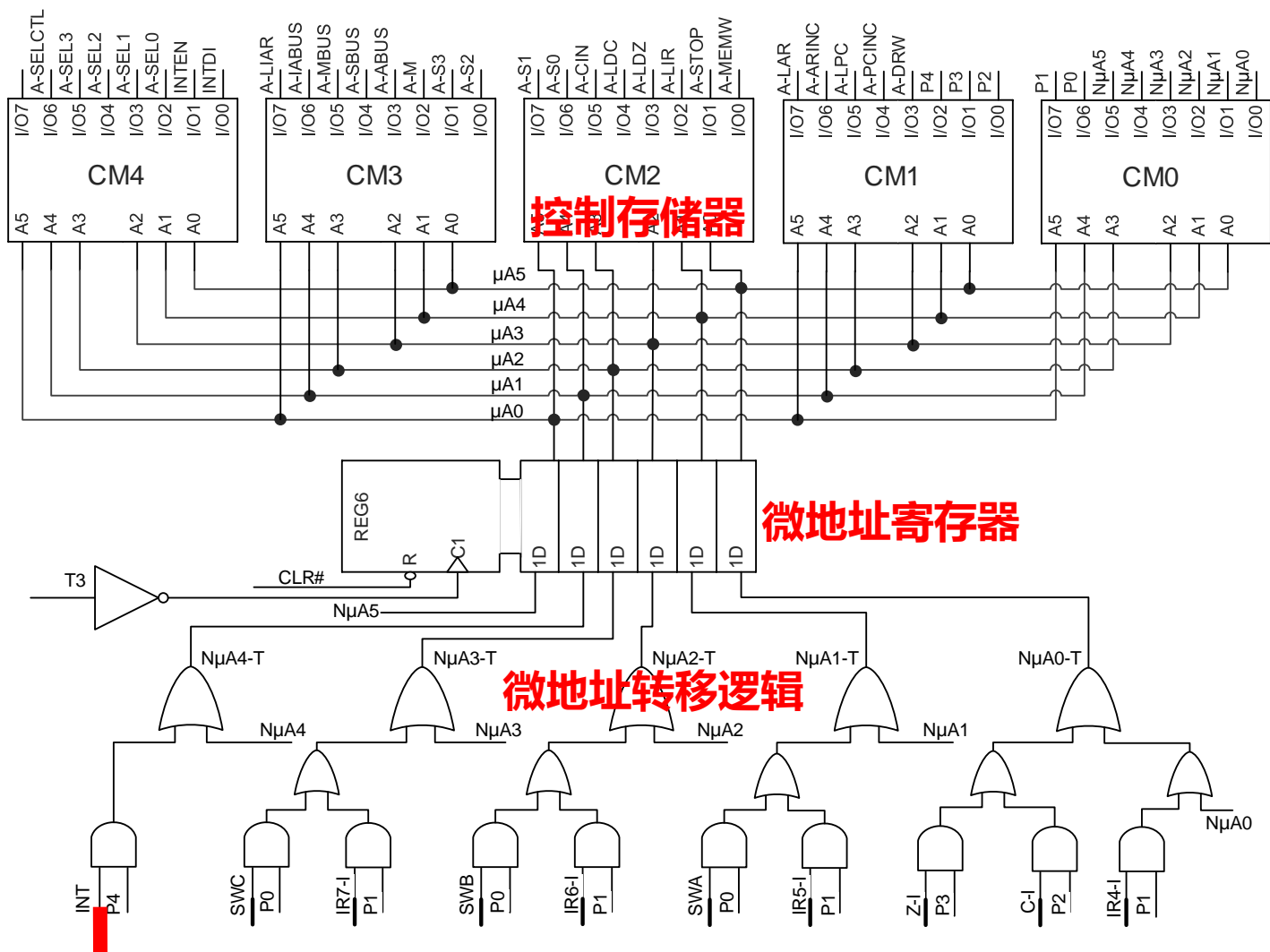
# 实验四 | 实验电路 >> 微程序控制器电路

操作模式 ( SWC,SWB,SWA )	实验功能
000	启动程序运行
001	写存储器
010	读存储器
011	读寄存器
100	写寄存器



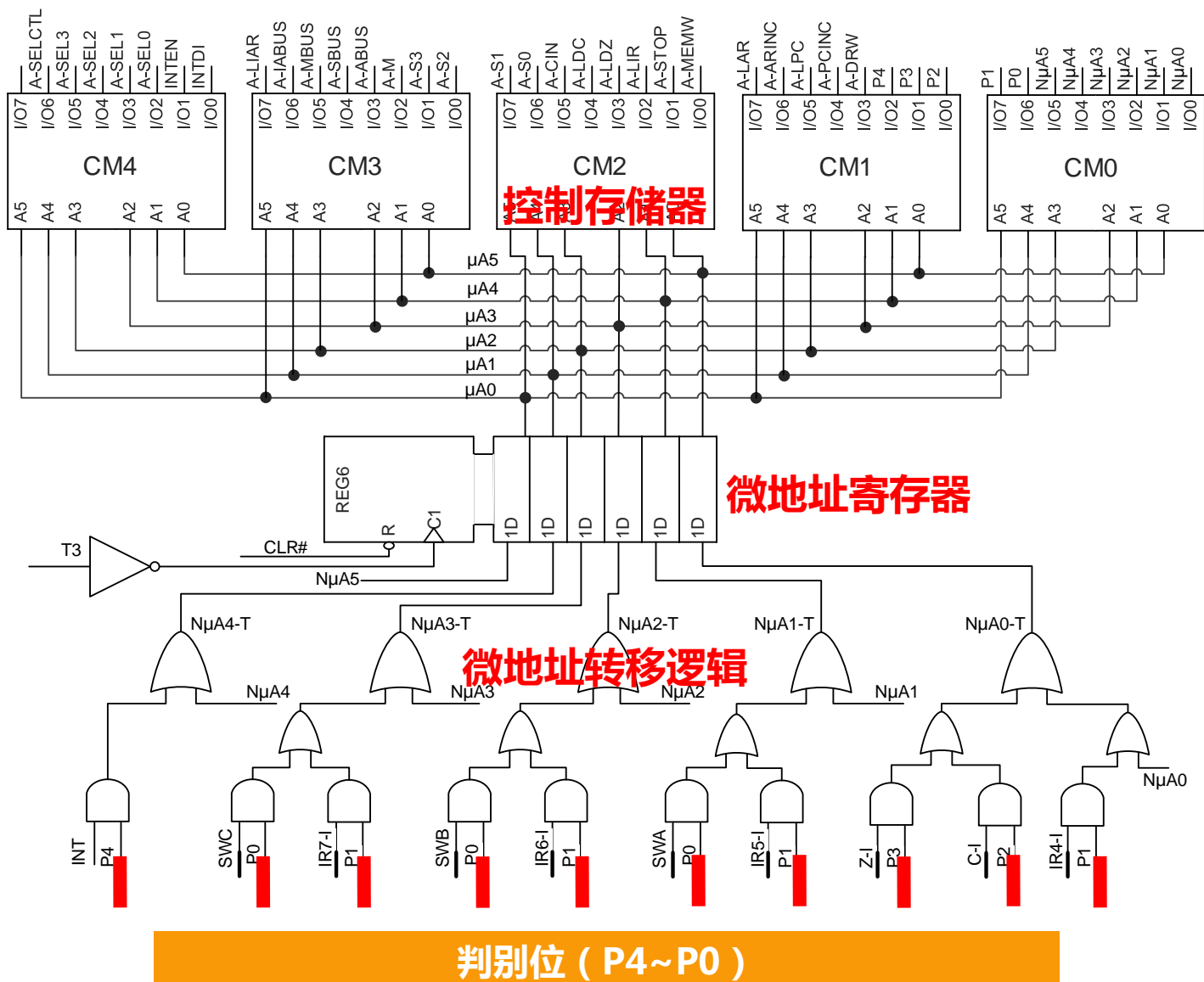


# 实验四 | 实验电路 >> 微程序控制器电路

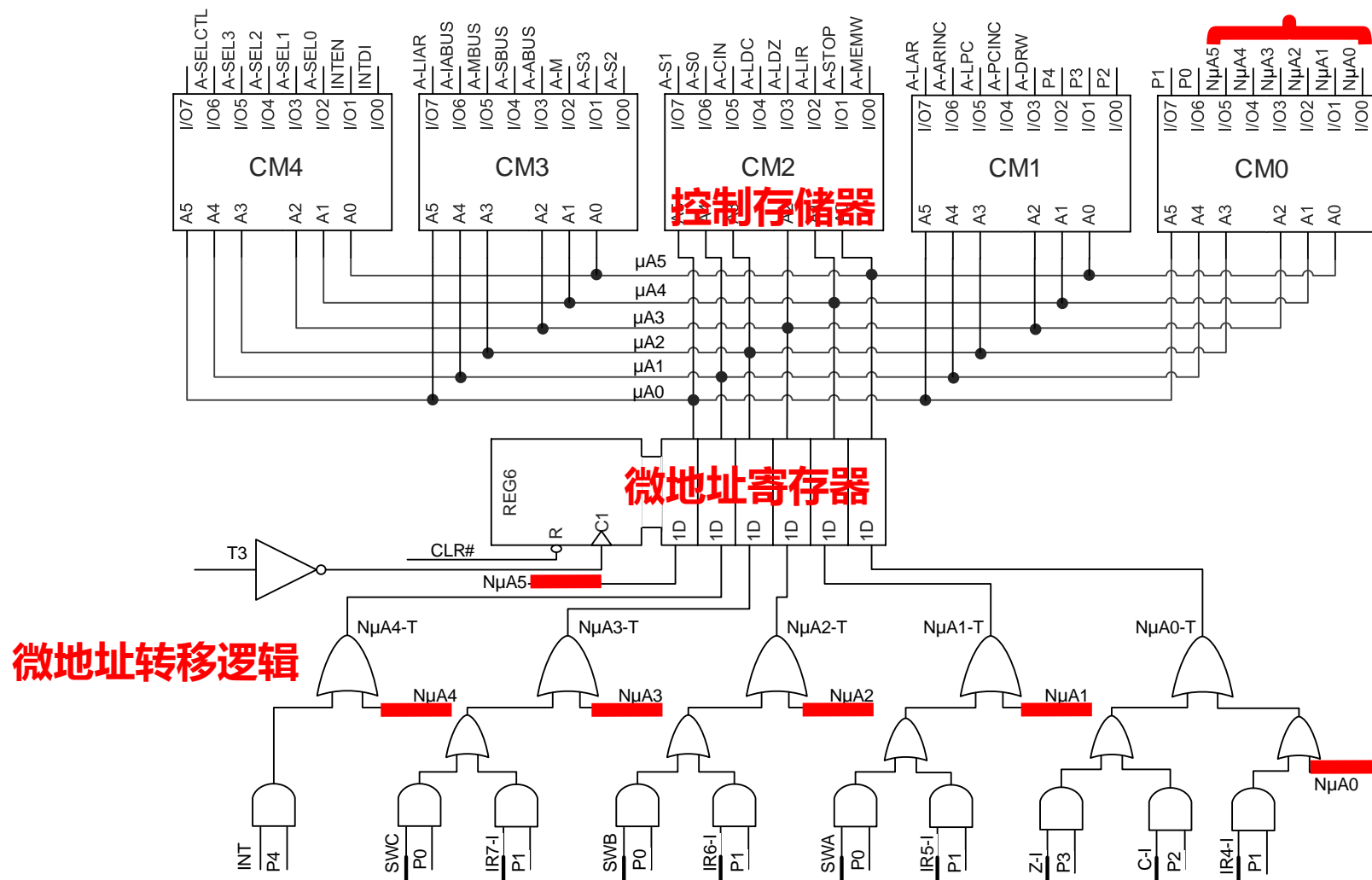


INT信号是时序发生器接收到中断请求脉冲PULSE后产生的中断信号

# 实验四 | 实验电路 >> 微程序控制器电路

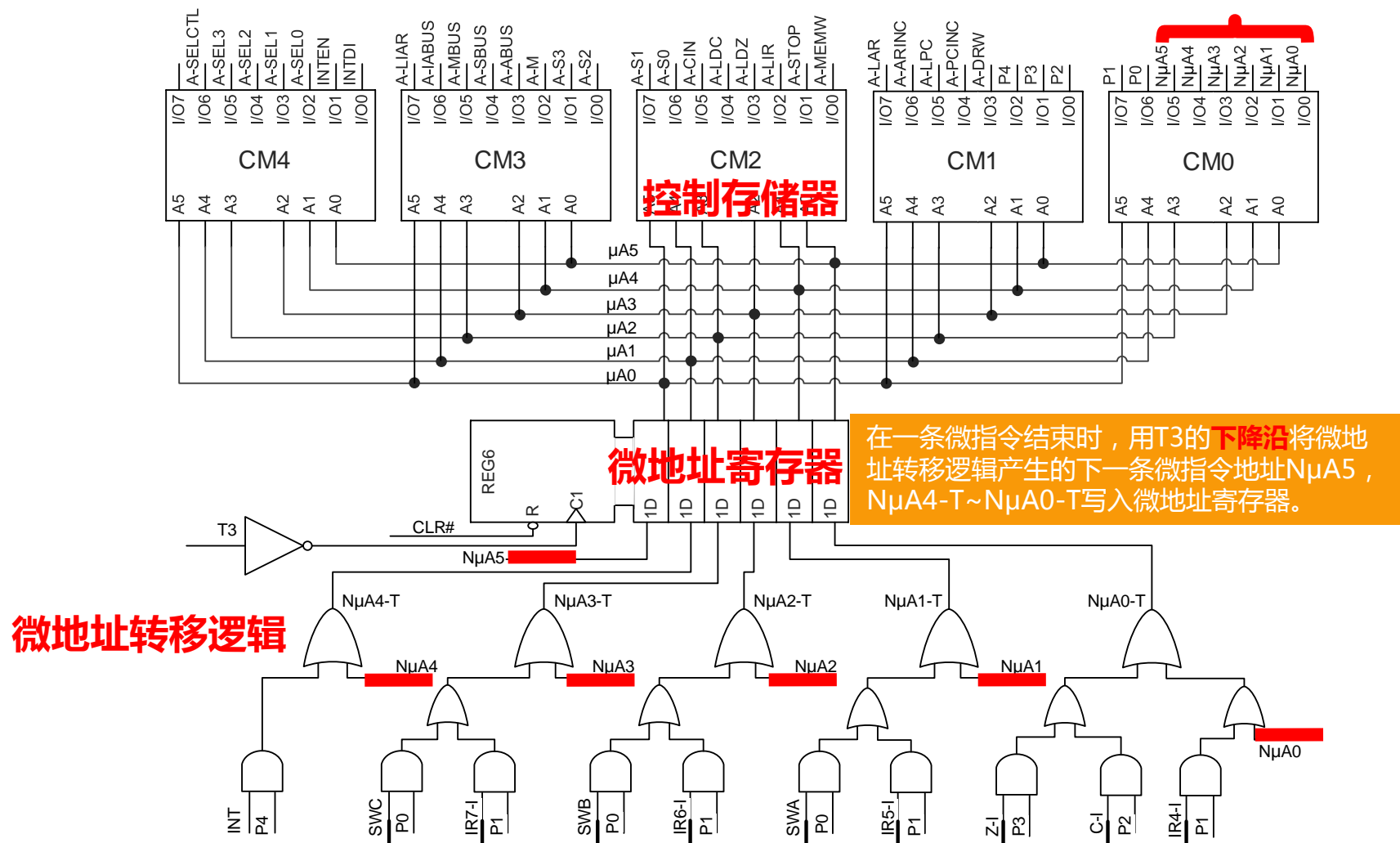


# 实验四 | 实验电路 >> 微程序控制器电路

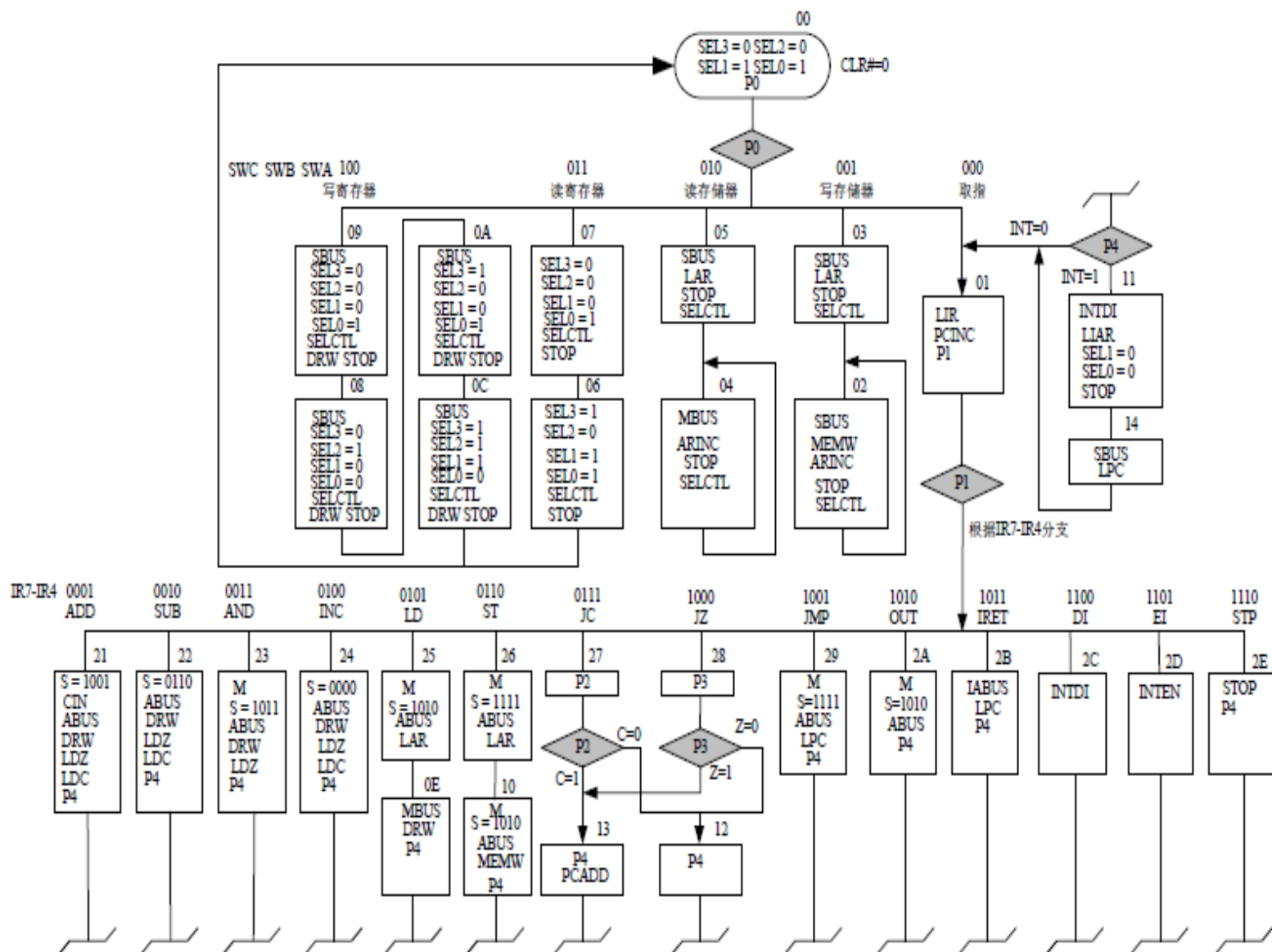


从控存中读取到的微指令的最后六位

# 实验四 | 实验电路 >> 微程序控制器电路



# 实验四 | 实验电路 >> 微程序写寄存器流程



# 实验四

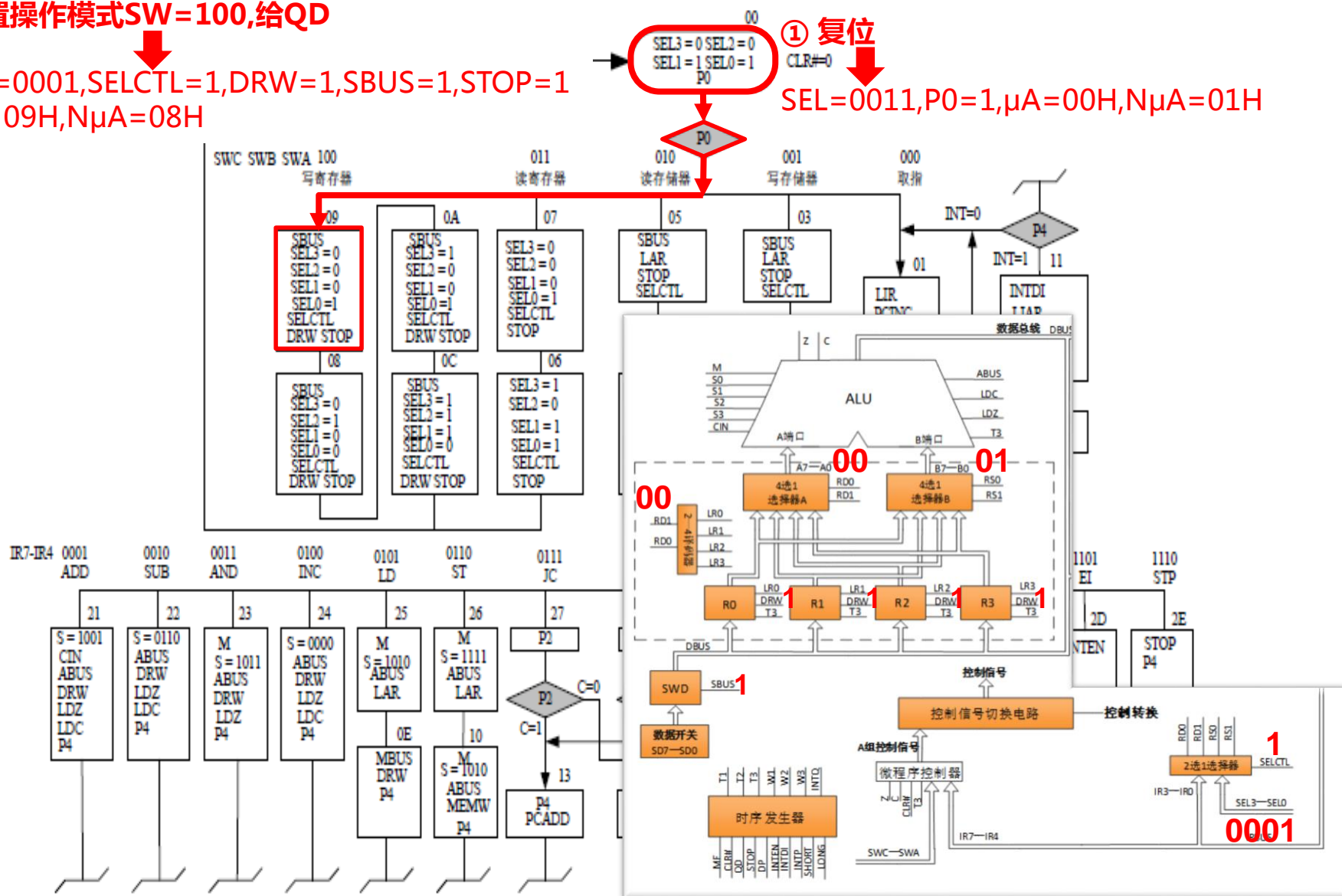
## 实验电路 >> 微程序写寄存器流程

② 置操作模式SW=100,给QD

SEL=0001,SELCTL=1,DRW=1,SBUS=1,STOP=1  
μA=09H,NμA=08H

① 复位

SEL=0011,P0=1,μA=00H,NμA=01H



# 实验四

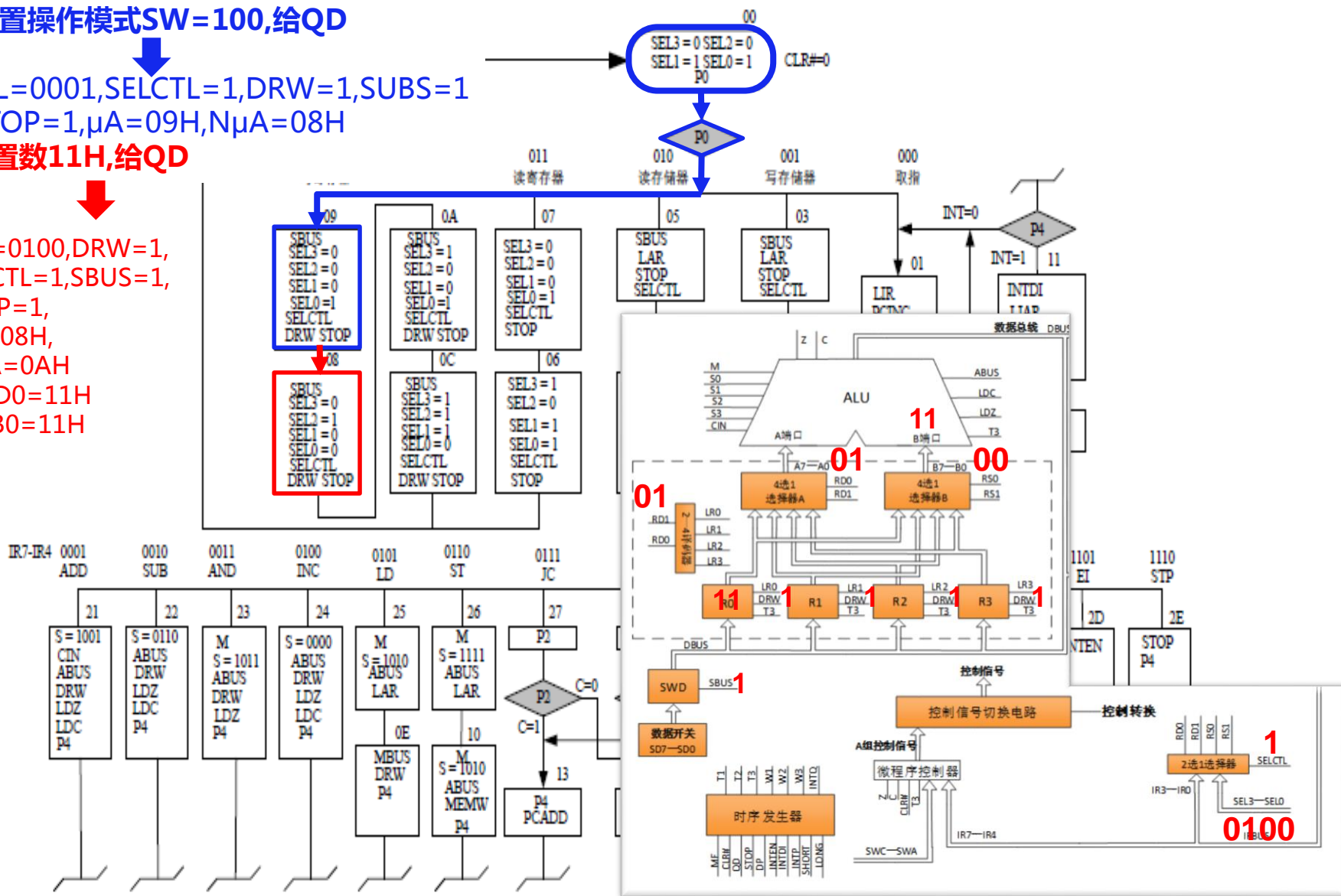
## 实验电路 >> 微程序写寄存器流程

② 置操作模式SW=100,给QD

```
SEL=0001,SELCTL=1,DRW=1,SUBS=1
,STOP=1,μA=09H,NμA=08H
```

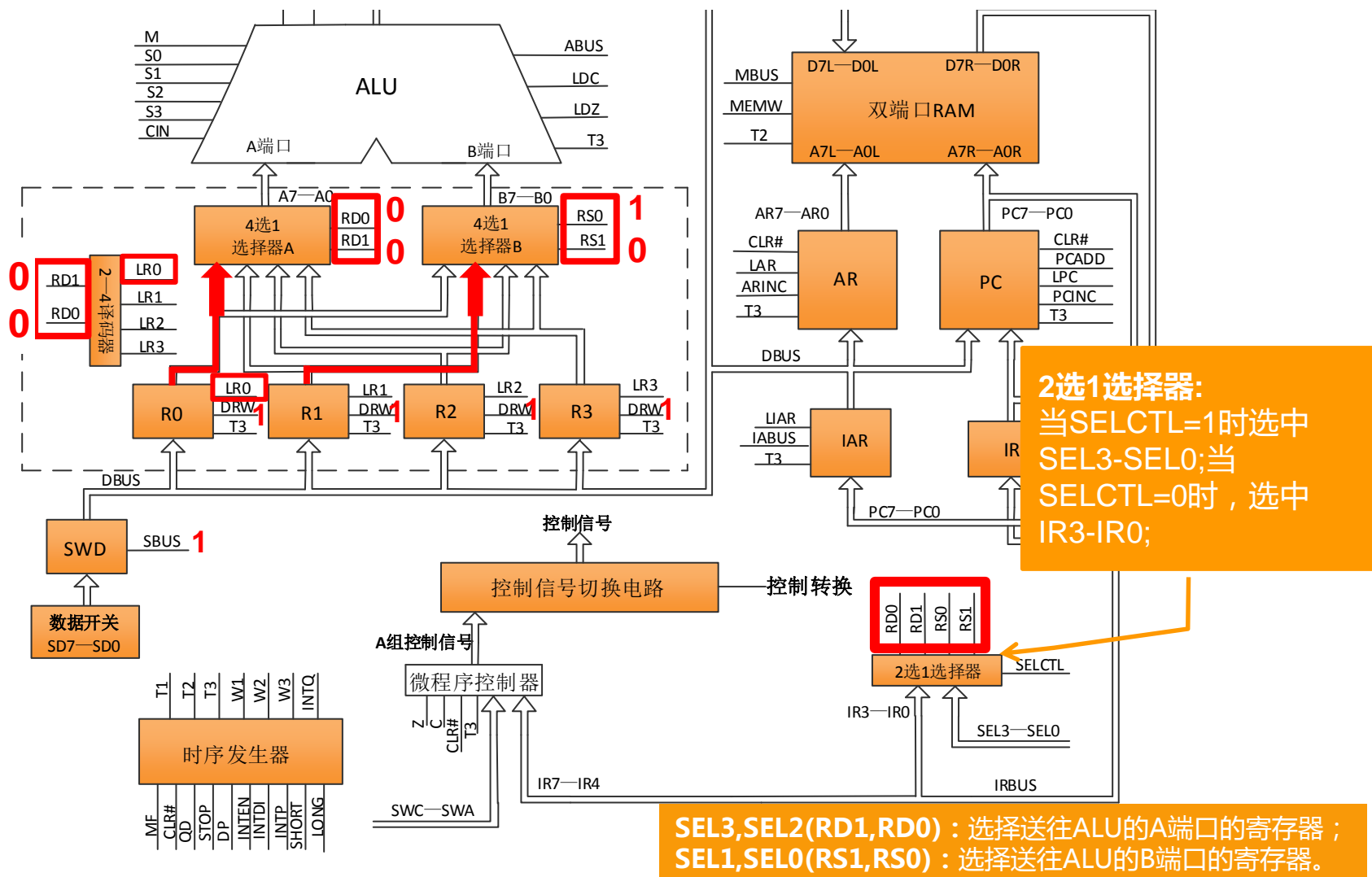
### ③置数11H,给QD

```
SEL=0100,DRW=1,  
SELCTL=1,SBUS=1,  
STOP=1,  
μA=08H,  
NμA=0AH  
D7-D0=11H  
B7-B0=11H
```



# 实验四 | 实验电路 >> 微程序写寄存器流程

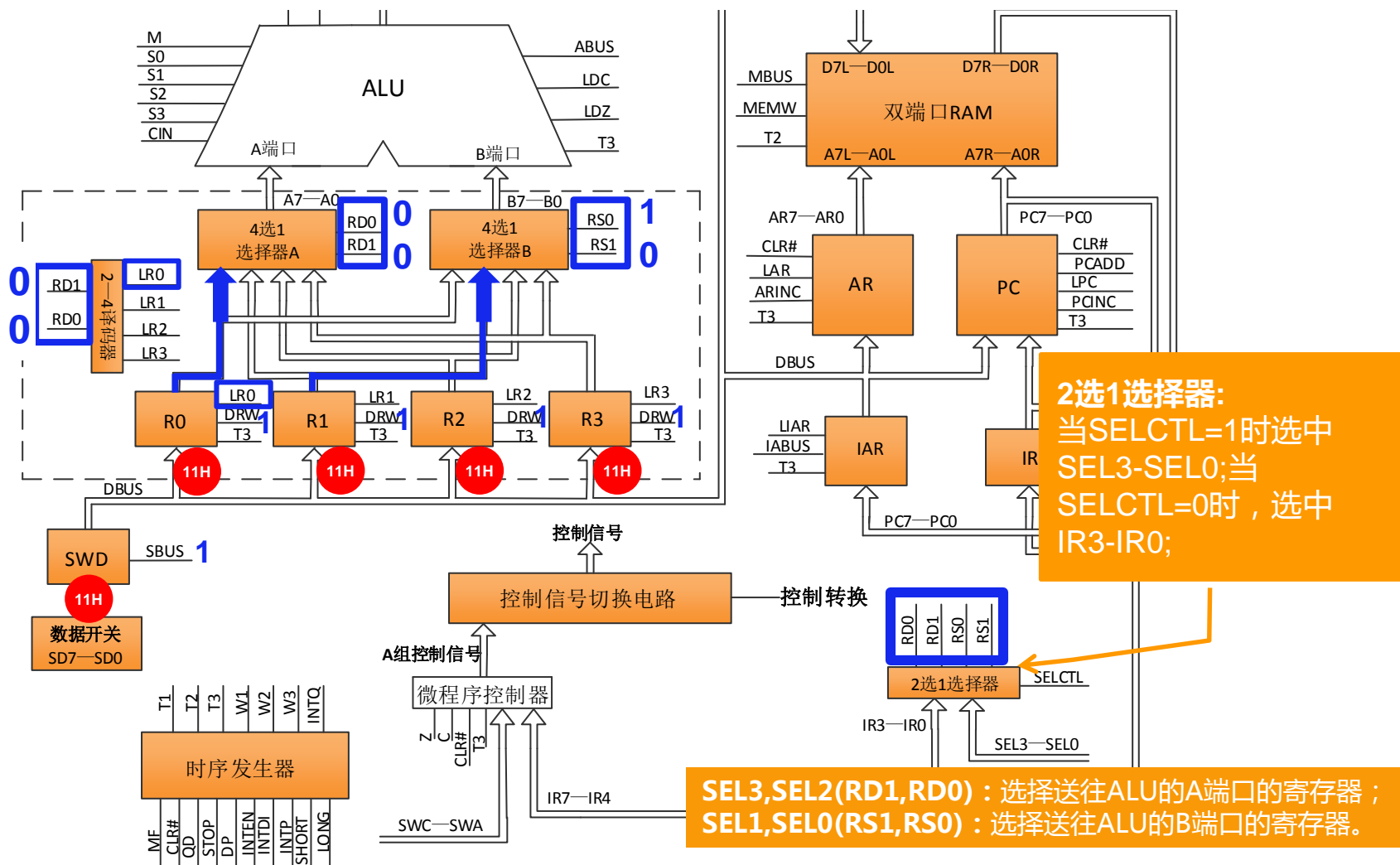
③ 条件：SEL=0001,SELCTL=1,DRW=1,SBUS=1,STOP=1, $\mu A=09H$ , $N\mu A=08H$





# 实验四 | 实验电路>>微程序写寄存器流程

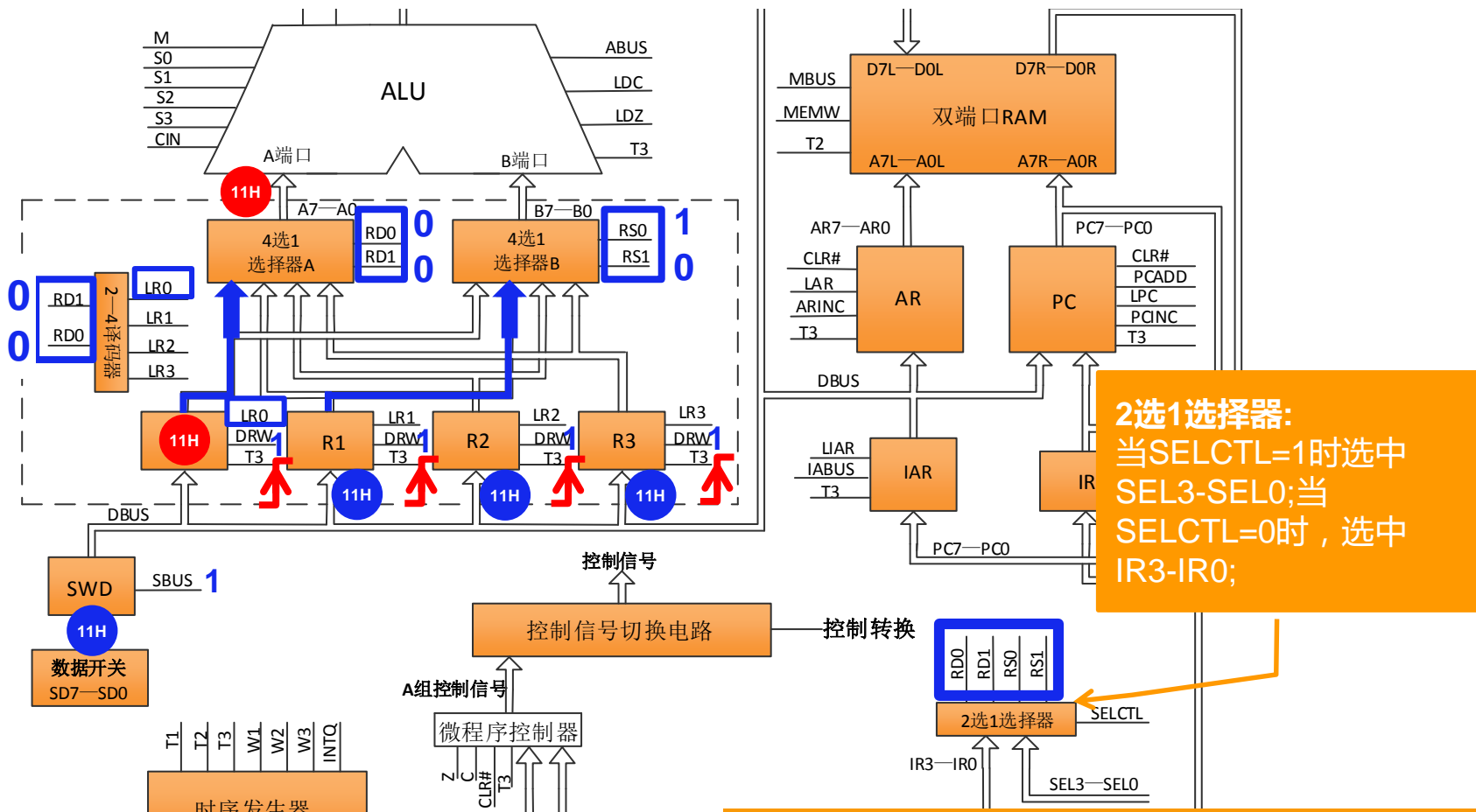
③ 条件：SEL=0001,SELCTL=1,DRW=1,SBUS=1,STOP=1, $\mu A=09H$ , $N\mu A=08H$   
置数 11H



# 实验四

## 实验电路>>微程序写寄存器流程

③ 条件: SEL=0001, SELCTL=1, DRW=1, SBUS=1, STOP=1,  $\mu A=09H$ ,  $N\mu A=08H$   
置数 11H, 给 QD

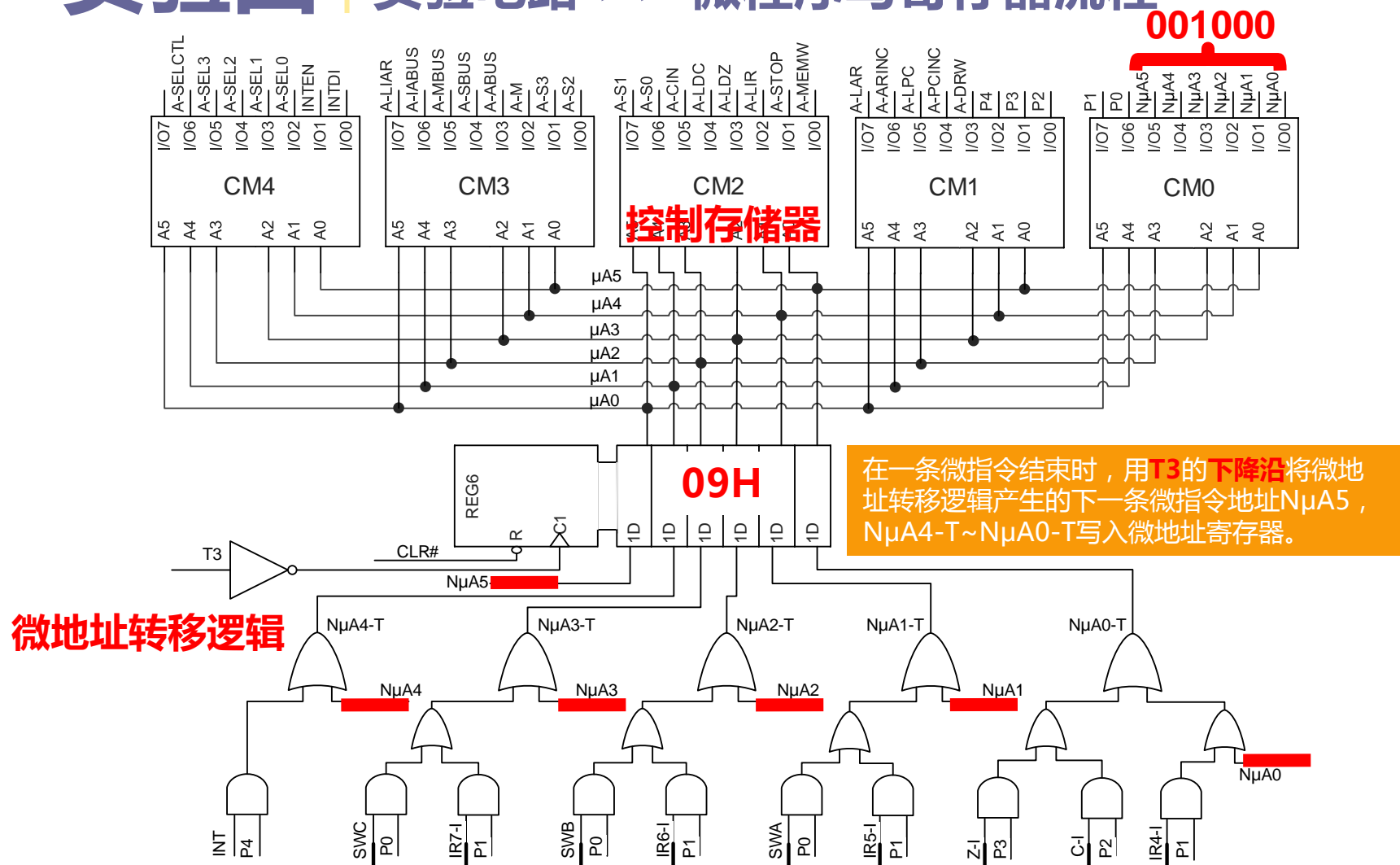


**思考1：**当前ALU的A/B两个端口的值对吗？

**SEL3,SEL2(RD1,RD0) : 选择送往ALU的A端口的寄存器 ;**  
**SEL1,SEL0(RS1,RS0) : 选择送往ALU的B端口的寄存器。**

# 实验四

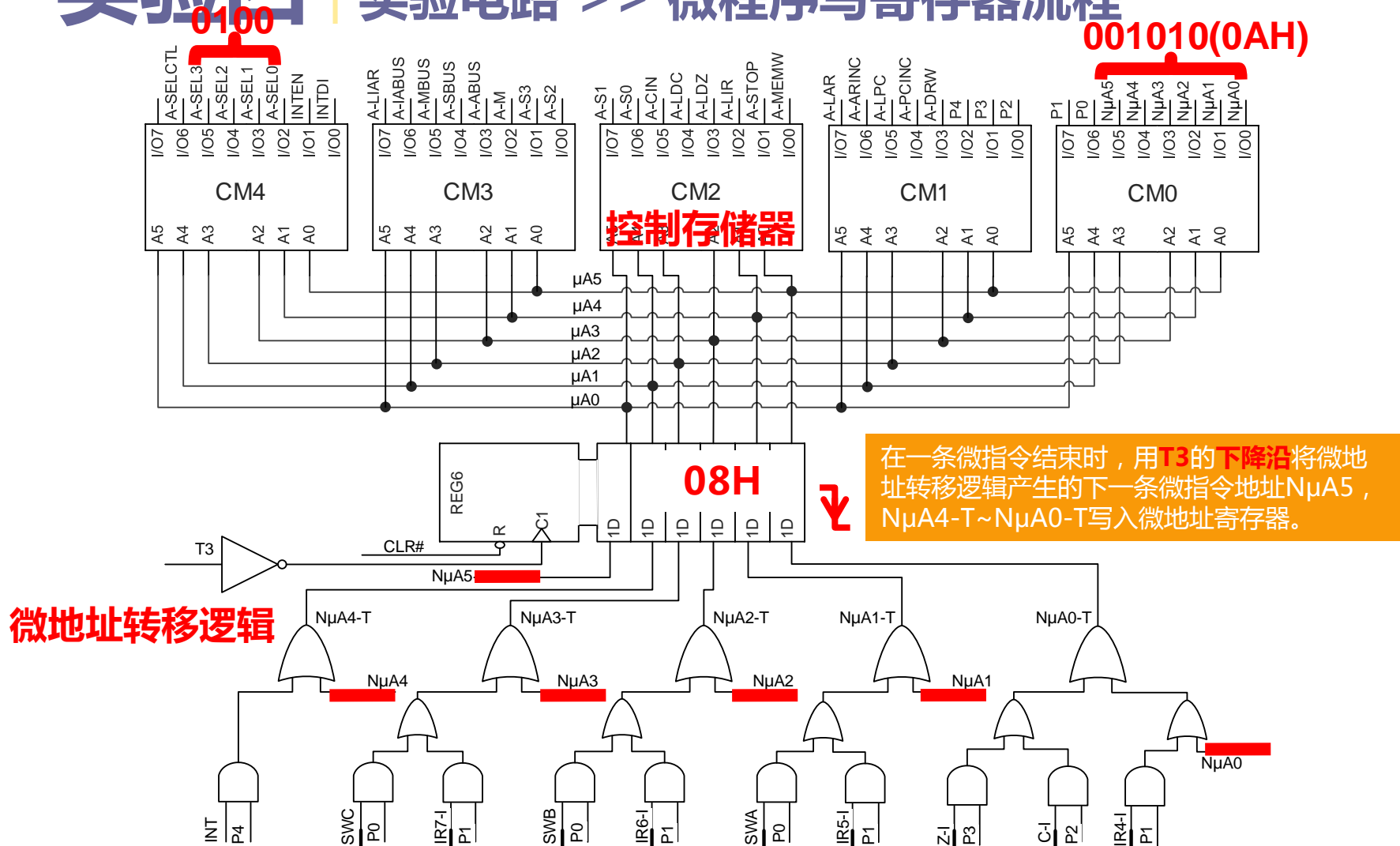
## 实验电路 >> 微程序写寄存器流程



③ 条件：SEL=0001,SELCTL=1,DRW=1,SBUS=1,STOP=1,μA=09H,NμA=08H  
置数 11H

# 实验四

## 实验电路 >> 微程序写寄存器流程

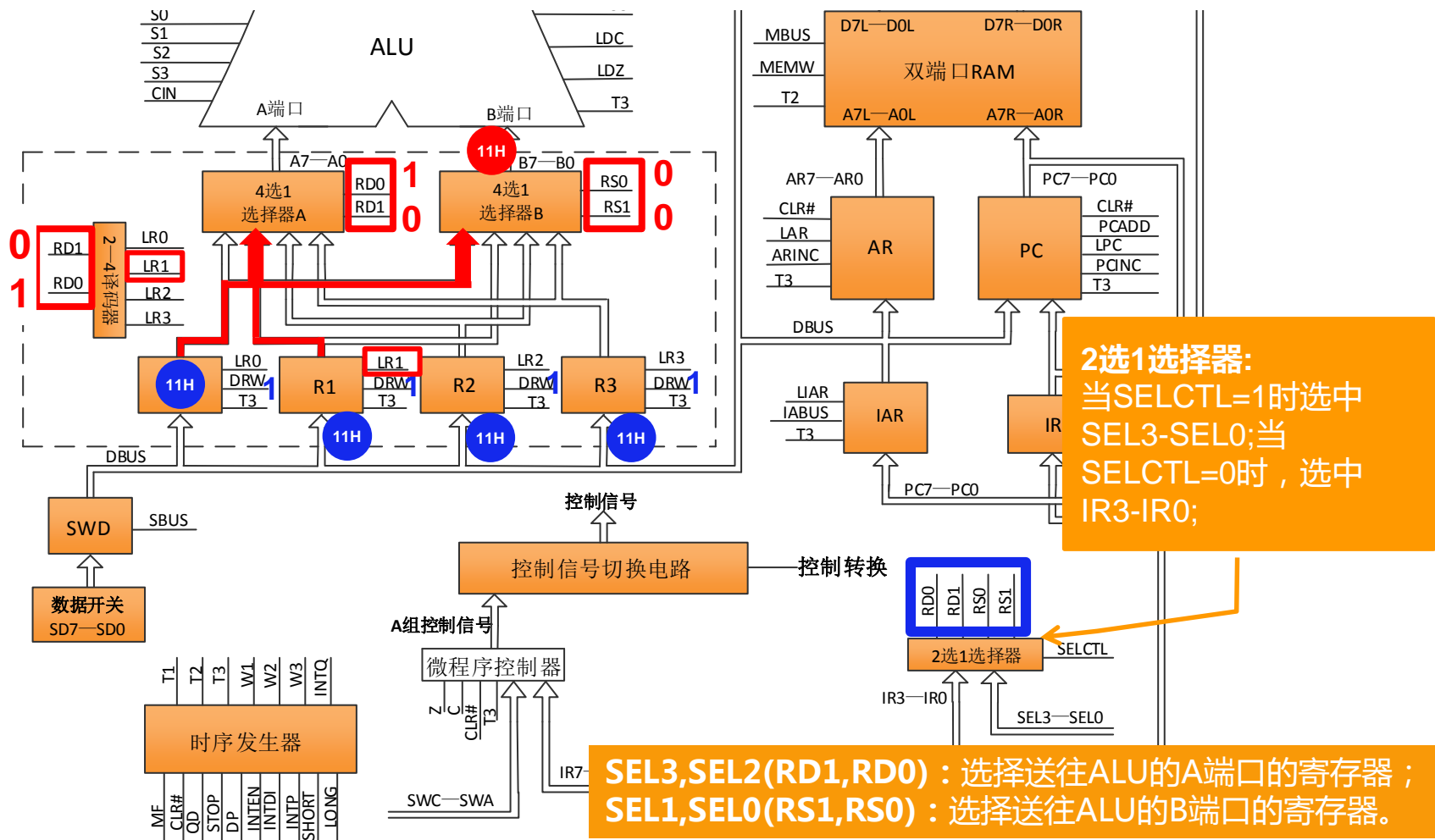


- ③ 条件：SEL=0001,SELCTL=1,DRW=1,SBUS=1,STOP=1,μA=09H,NμA=08H  
置数 11H,给QD  
结果：SEL=0100, SELCTL=1,DRW=1,SBUS=1,STOP=1,μA=08H,NμA=0AH

# 实验四 | 实验电路>>微程序写寄存器流程

③ 条件：SEL=0001,SELCTL=1,DRW=1,SBUS=1,STOP=1, $\mu A=09H$ , $N\mu A=08H$   
置数 11H,给QD

结果：SEL=0100, SELCTL=1,DRW=1,SBUS=1,STOP=1, $\mu A=08H$ , $N\mu A=0AH$



# 实验四

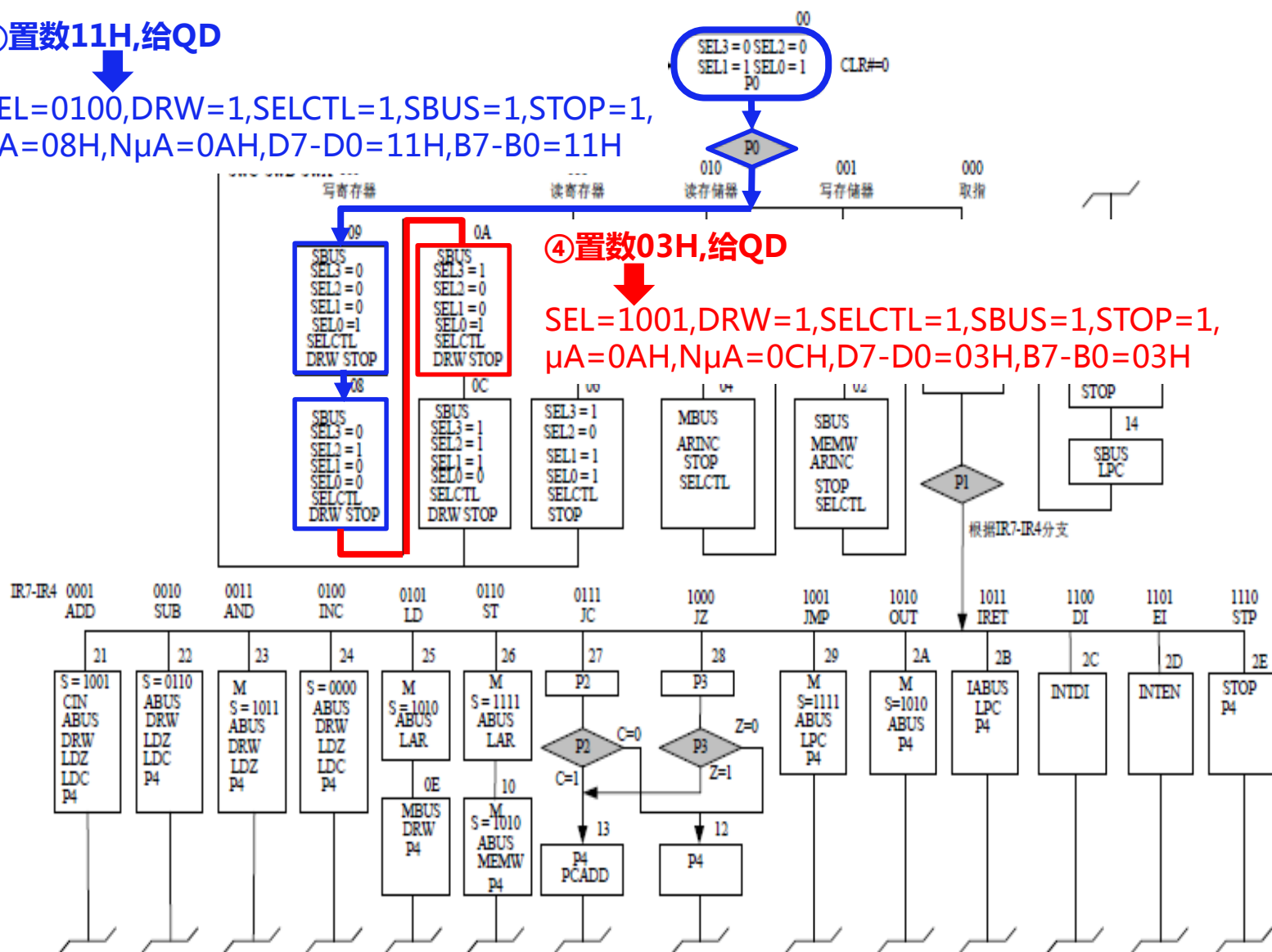
## 实验电路 >> 微程序写寄存器流程

### ③置数11H,给QD

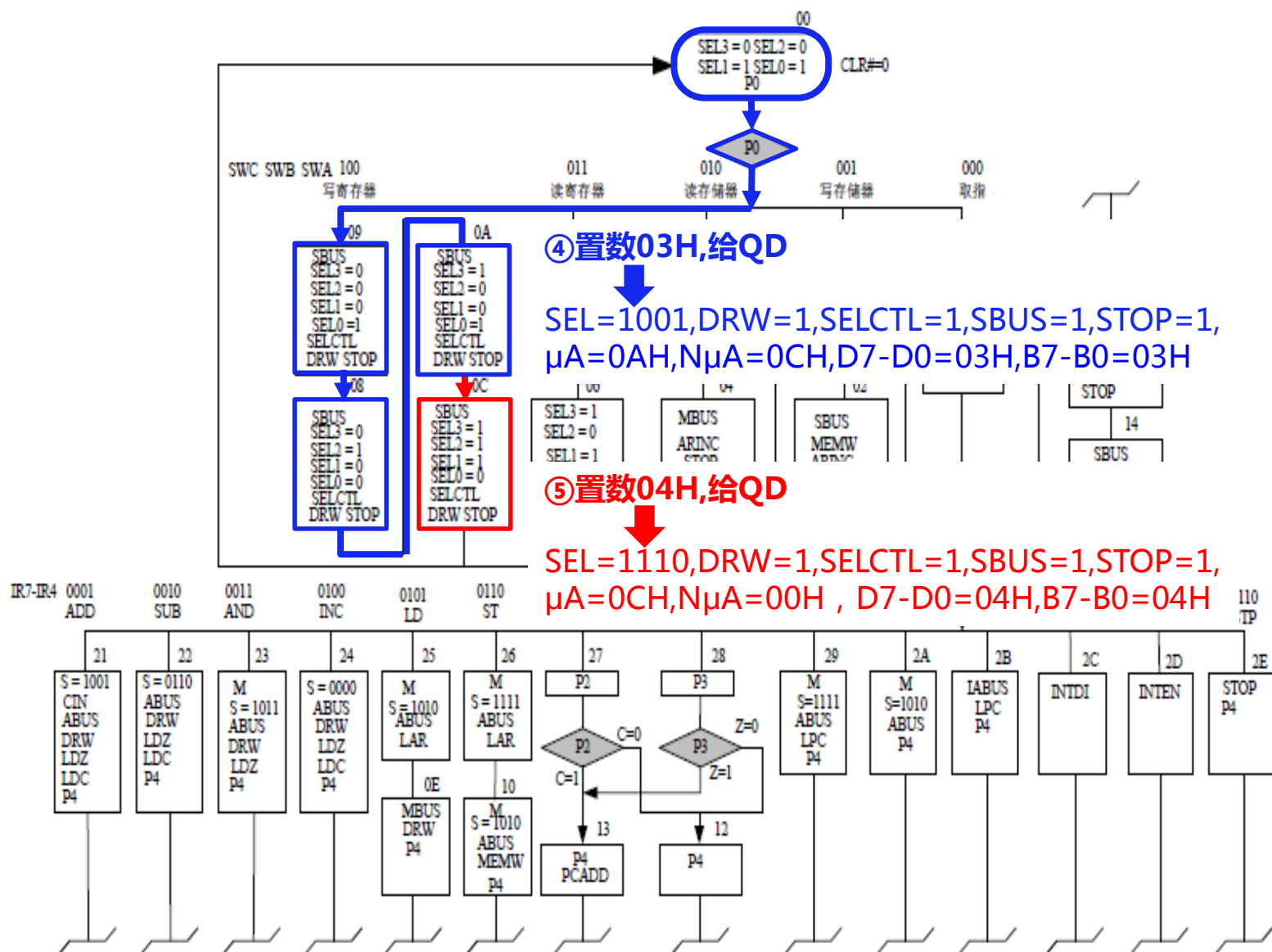
SEL=0100,DRW=1,SELCTL=1,SBUS=1,STOP=1,  
μA=08H,NμA=0AH,D7-D0=11H,B7-B0=11H

### ④置数03H,给QD

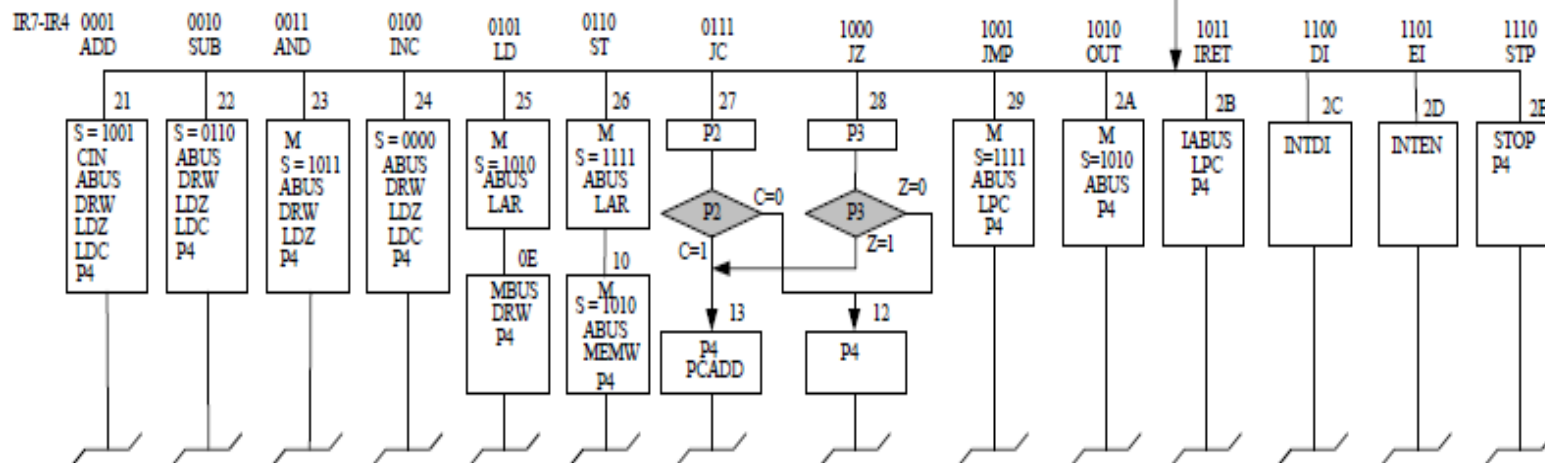
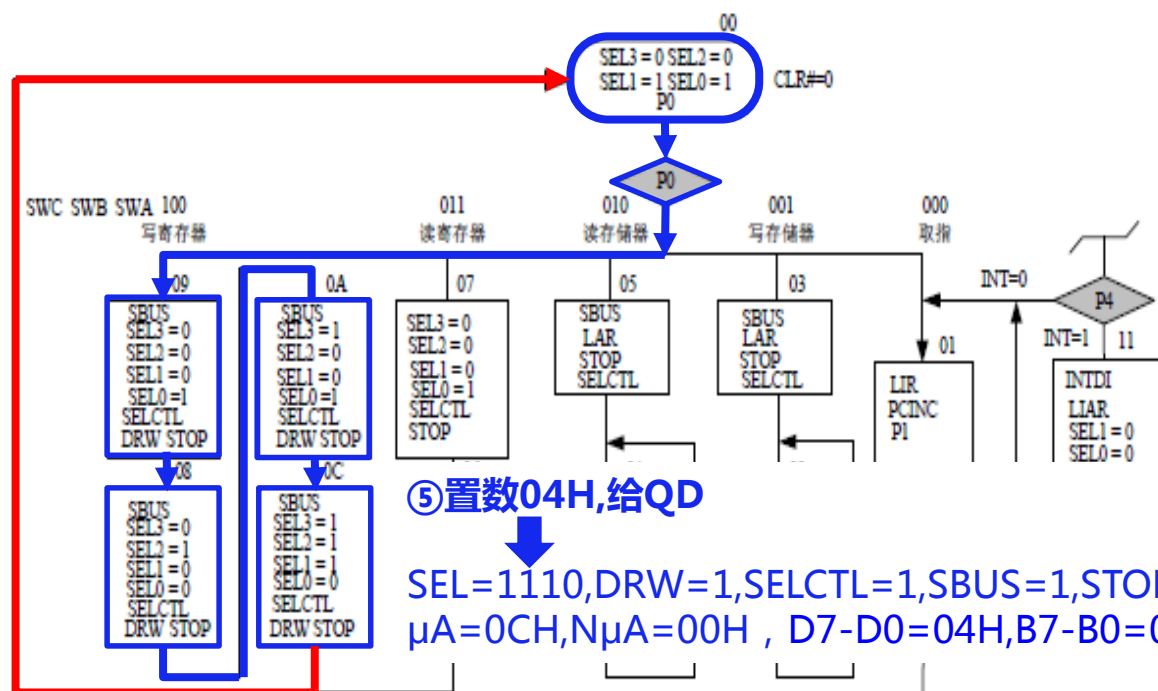
SEL=1001,DRW=1,SELCTL=1,SBUS=1,STOP=1,  
μA=0AH,NμA=0CH,D7-D0=03H,B7-B0=03H



# 实验四 | 实验电路 >> 微程序写寄存器流程

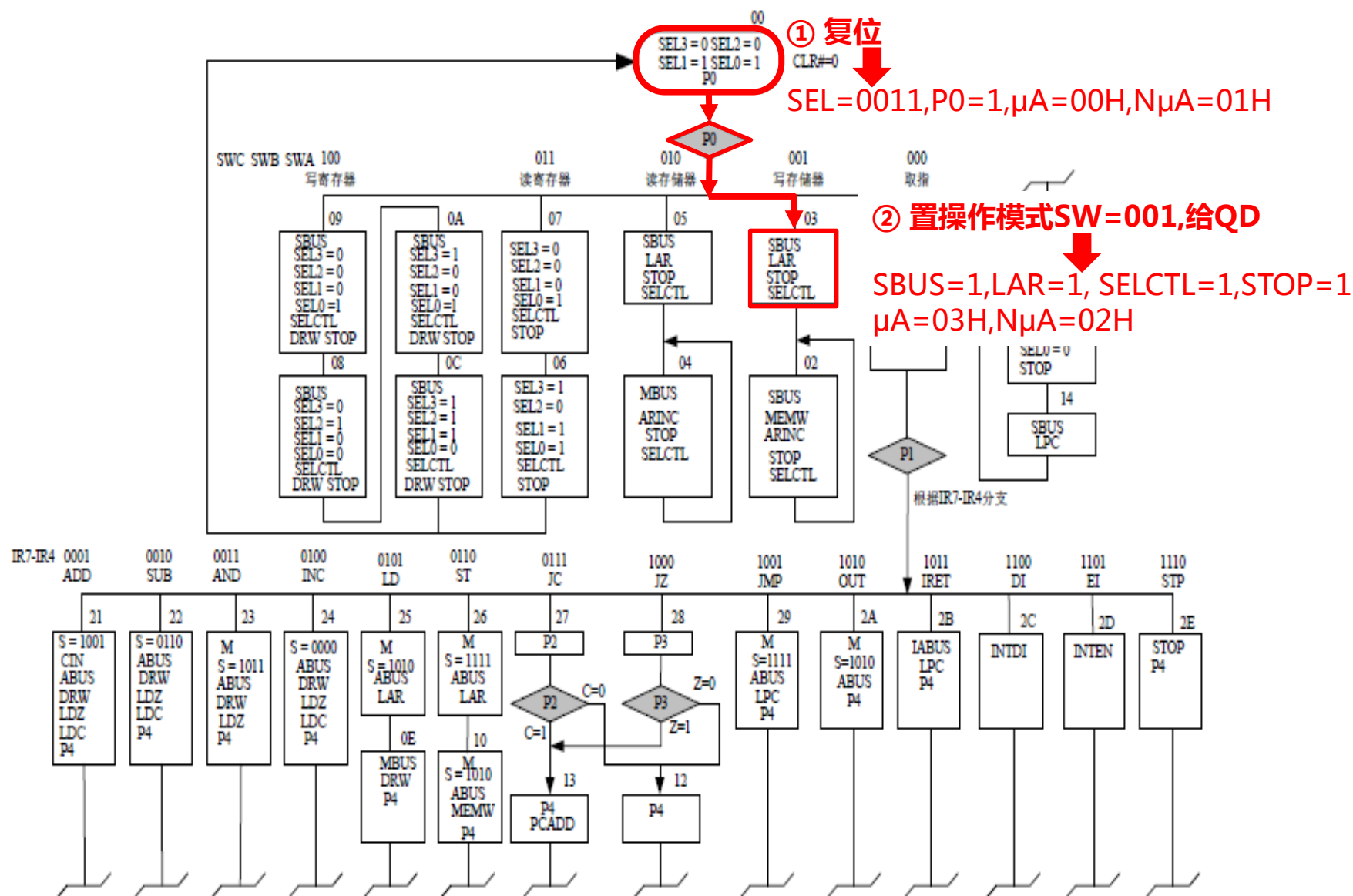


# 实验四 | 实验电路 >> 微程序写寄存器流程

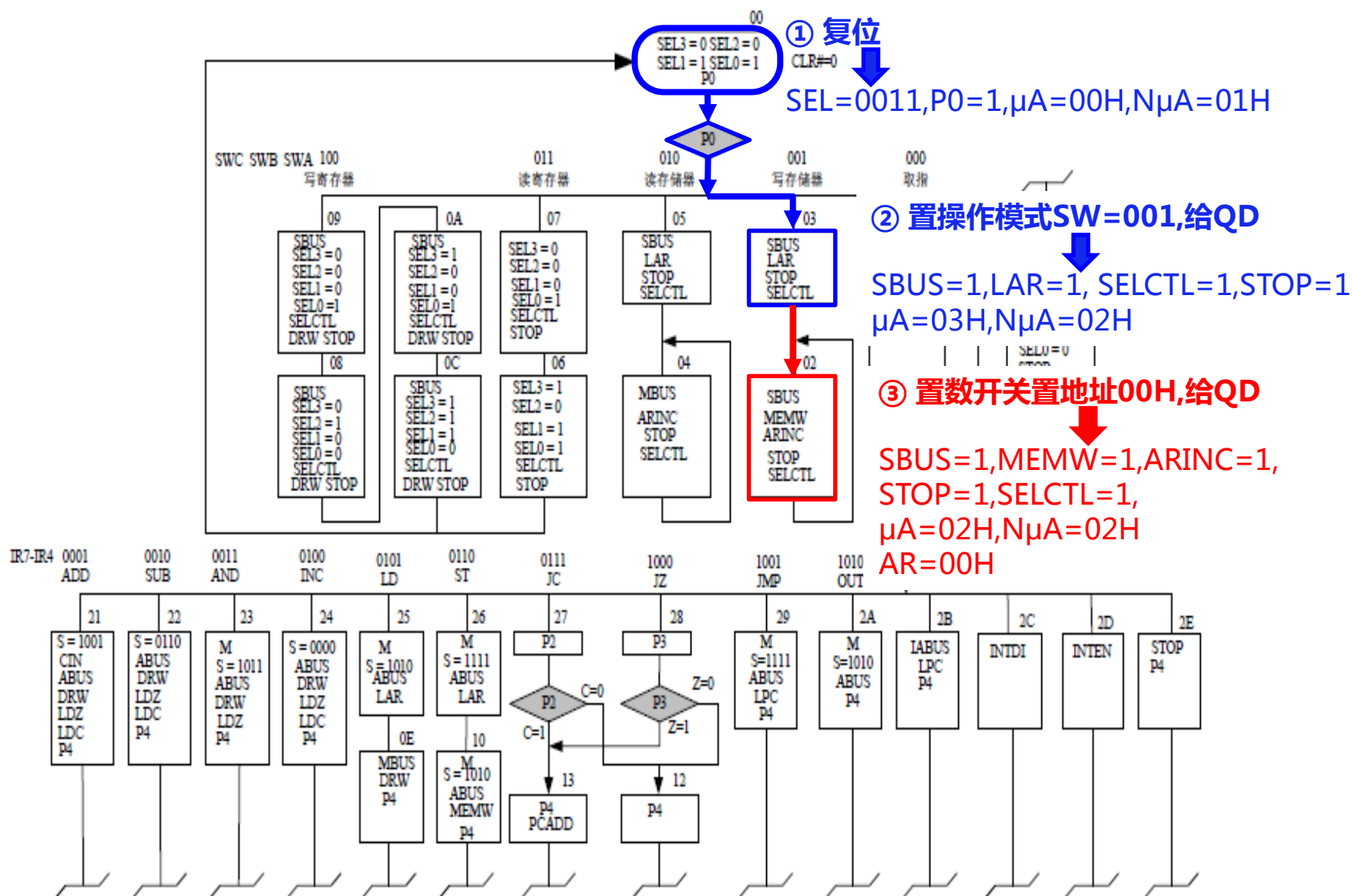




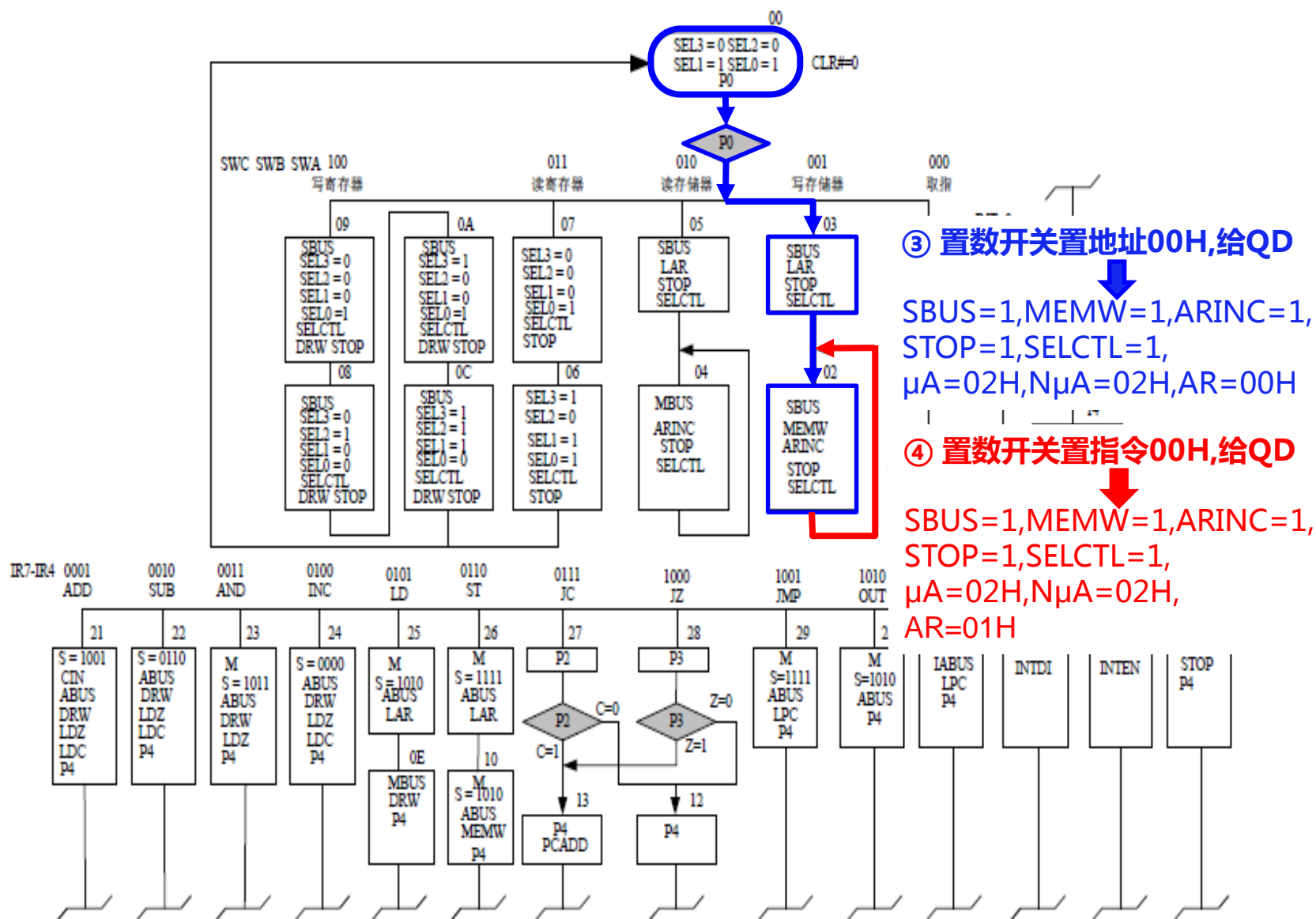
# 实验四 | 实验电路 >> 微程序写存储器流程



# 实验四 | 实验电路 >> 微程序写存储器流程



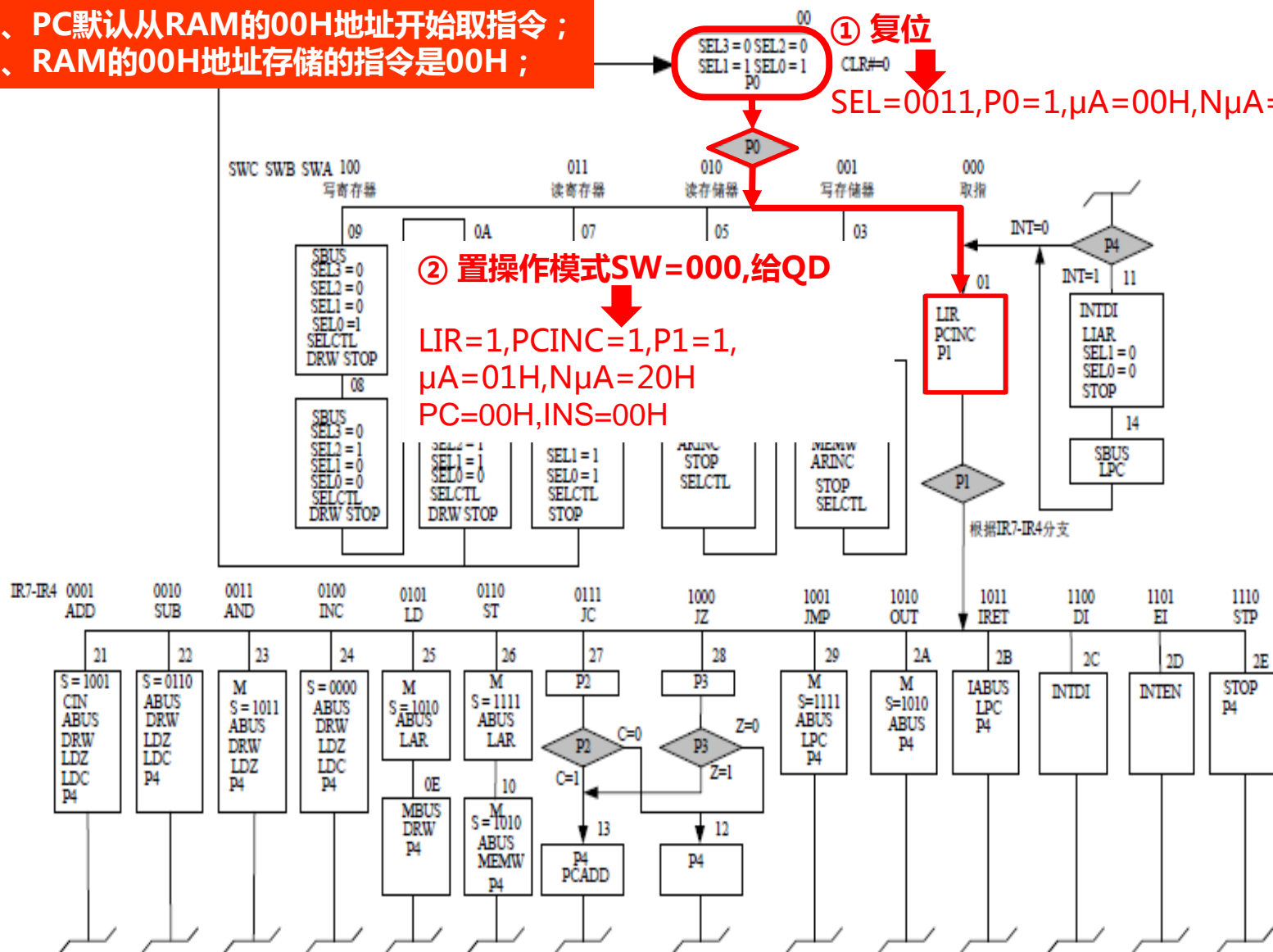
# 实验四 | 实验电路 >> 微程序写存储器流程



# 实验四 | 实验电路 >> 微程序取指流程

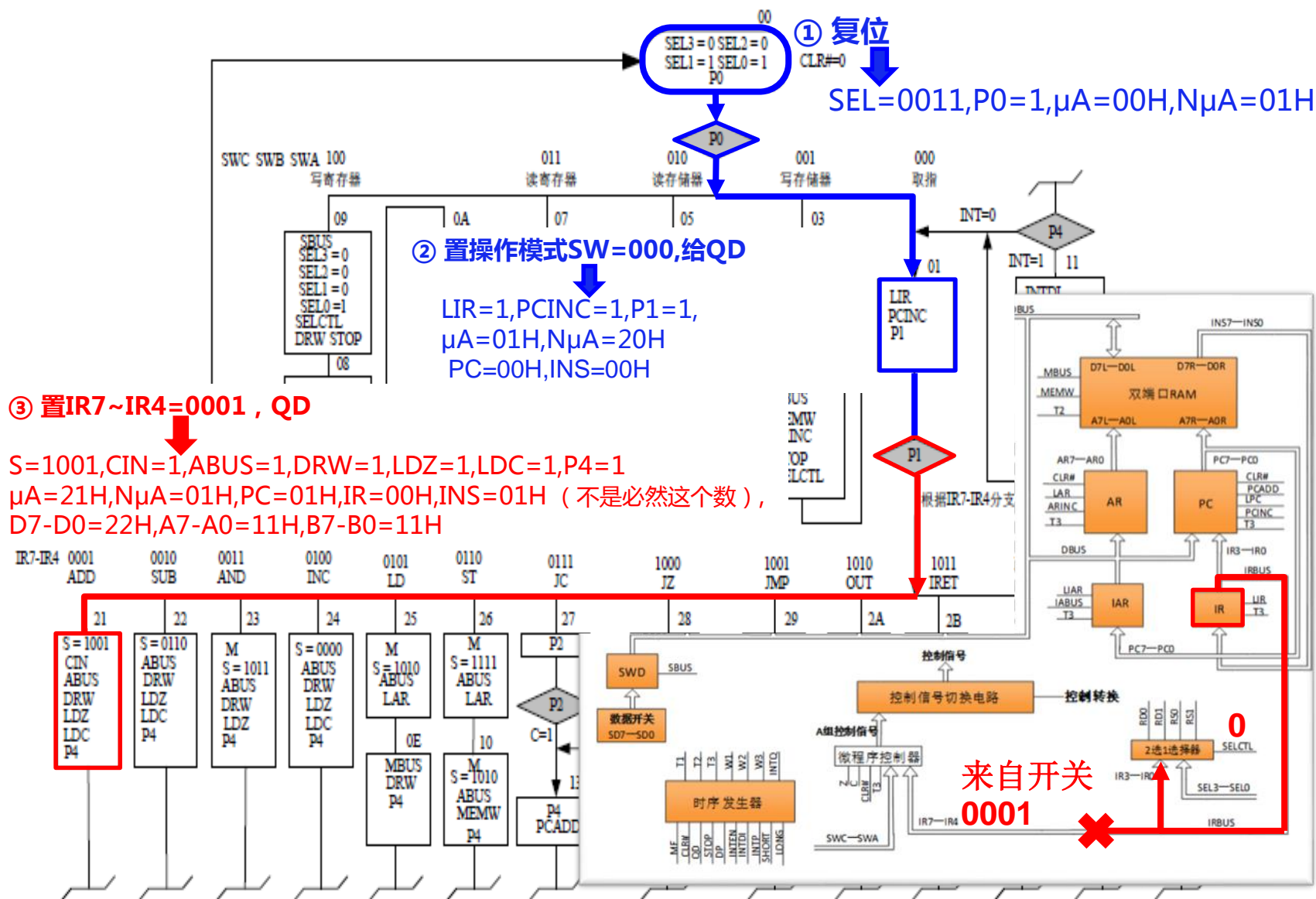
- 1、PC默认从RAM的00H地址开始取指令；
- 2、RAM的00H地址存储的指令是00H；

① 复位  
CLR#0  
SEL=0011, P0=1,  $\mu A=00H$ ,  $N\mu A=01H$

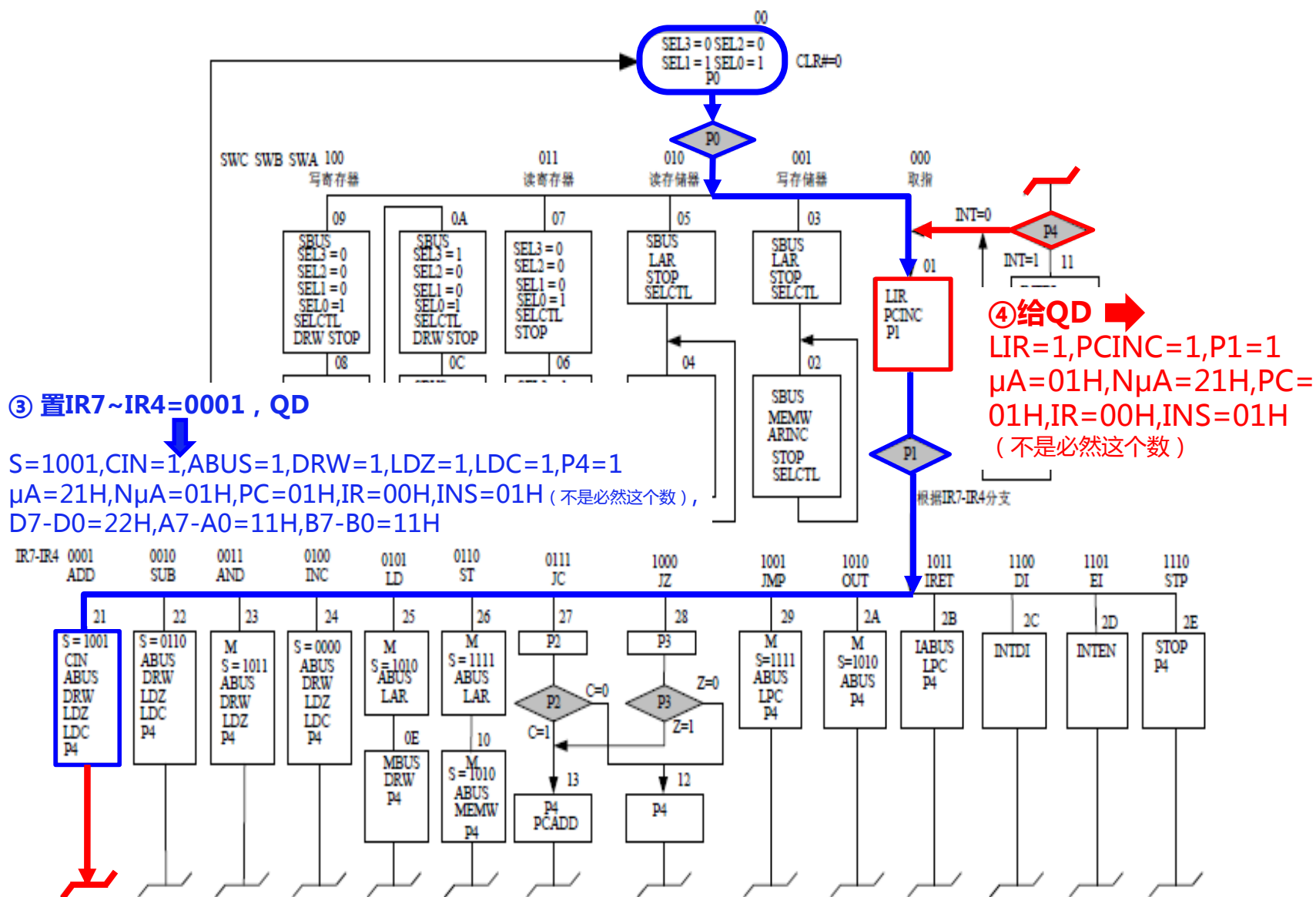


# 实验四

## 实验电路 >> 微程序取指流程



# 实验四 | 实验电路 >> 微程序取指流程



# 实验四 | 实验电路 >> TEC-8 模型计算机指令系统

名称	助记符	功能	指令格式		
			IR(7~4)	IR(3~2)	IR(1~0)
加法	ADD Rd, Rs	$Rd \leftarrow Rd + Rs$	0001	Rd	Rs
减法	SUB Rd, Rs	$Rd \leftarrow Rd - Rs$	0010	Rd	Rs
逻辑与	AND Rd, Rs	$Rd \leftarrow Rd \text{ and } Rs$	0011	Rd	Rs
加1	INC Rd	$Rd \leftarrow Rd + 1$	0100	Rd	XX
取数	LD Rd, [Rs]	$Rd \leftarrow [Rs]$	0101	Rd	Rs
存数	ST Rs, [Rd]	$Rs \rightarrow [Rd]$	0110	Rd	Rs
C条件转移	JC addr	$C=1$ , 则 $PC \leftarrow @ + \text{offset}$	0111	offset	
Z条件转移	JZ addr	$Z=1$ , 则 $PC \leftarrow @ + \text{offset}$	1000	offset	
无条件转移	JMP [Rd]	$PC \leftarrow Rd$	1001	Rd	XX
输出	OUT Rs	$DBUS \leftarrow Rs$	1010	XX	Rs
中断返回	IRET	返回断点	1011	XX	XX
关中断	DI	禁止中断	1100	XX	XX
开中断	EI	允许中断	1101	XX	XX
停机	STP	暂停运行	1110	XX	XX

注：

1. XX代表随意值，Rs代表源寄存器号，Rd代表目的寄存器号；
2. 在条件转移指令中，@代表当前PC的值（@不是当前指令的PC值，是当前指令的PC值+1），offset是一个4位的有符号数，第3位是符号位，0代表正数，1代表负数。

# 实验四 | 实验任务

## ① 熟悉微程序流程图和微程序指令系统

- ✓ 跟踪控制台操作写寄存器、写存储器、读存储器、读寄存器的执行过程；
- ✓ 跟踪指令的执行过程
  - ✓ 执行ADD、LD、ST指令



## 实验四 | 实验步骤

### □ 实验准备（**不要打开电源** ）

- ① 控制器转换开关：微程序；
- ② 编程开关：正常；
- ③ 操作模式：DP = 1；
- ④ 参考连线：

控制器	IR4-I	IR5-I	IR6-I	IR7-I	/CS0
电平开关	K0	K1	K2	K3	GND

# 实验四 | 实验步骤

## 1. 跟踪控制台操作**写寄存器**的执行

- ① **打开电源**→按复位按钮CLR；
- ② 拨动操作模式开关SW=100，按QD，进入目标控制台操作模式；
- ③ 向R0-R3寄存器中分别存入数据。

## 2. 跟踪控制台操作**写存储器**的执行

- ① 按复位按钮CLR；
- ② 拨动操作模式开关SW=001，按QD，进入目标控制台操作模式；
- ③ 向存储器的00H,01H,02H,03H地址中分别存入指令或数据。

**注意：**实验过程中根据实验过程记录表观察指示灯状态并记录。

# 实验四 | 实验步骤

## 3. 跟踪指令的执行

- ① 按复位按钮CLR ；
- ② 设置SWC=0,SWB=0,SWA=0,按QD , 进入启动程序运行模式 ；
- ③ 设置电平开关K3~K0为0001 ( 对应IR7~IR4 ) , 按QD ；
- ④ 跟踪ADD指令的执行 ；
- ⑤ 设置操作码 , 跟踪LD指令的执行 ；
- ⑥ 设置操作码 , 跟踪ST指令的执行。

注意：实验过程中根据实验过程记录表观察指示灯状态并记录。

# 实验四 | 实验步骤

## 4. 跟踪控制台操作**读寄存器**的执行

- ① 按复位按钮CLR；
- ② 拨动操作模式开关SW=011，按QD，进入目标控制台操作模式；
- ③ 从R0-R3寄存器中分别读出数据并跟踪流程。

## 5. 跟踪控制台操作**读存储器**的执行

- ① 按复位按钮CLR；
- ② 拨动操作模式开关SW=010，按QD，进入目标控制台操作模式；
- ③ 从存储器的00H,01H,02H,03H地址中分别读出指令或数据。

**注意：**实验过程中根据实验过程记录表观察指示灯状态并记录。

# 实验四 | 实验要求

- 掌握TEC-8模型计算机微程序控制器的工作原理；
- 完成实验报告，内容包括：
  - 实验目的；
  - 实验过程表格
    - 需要记录的过程：
      - 读寄存器、写寄存器、写存储器、读存储器的执行
      - ADD、LD、ST三个指令的执行
      - 按照实验步骤依次操作