

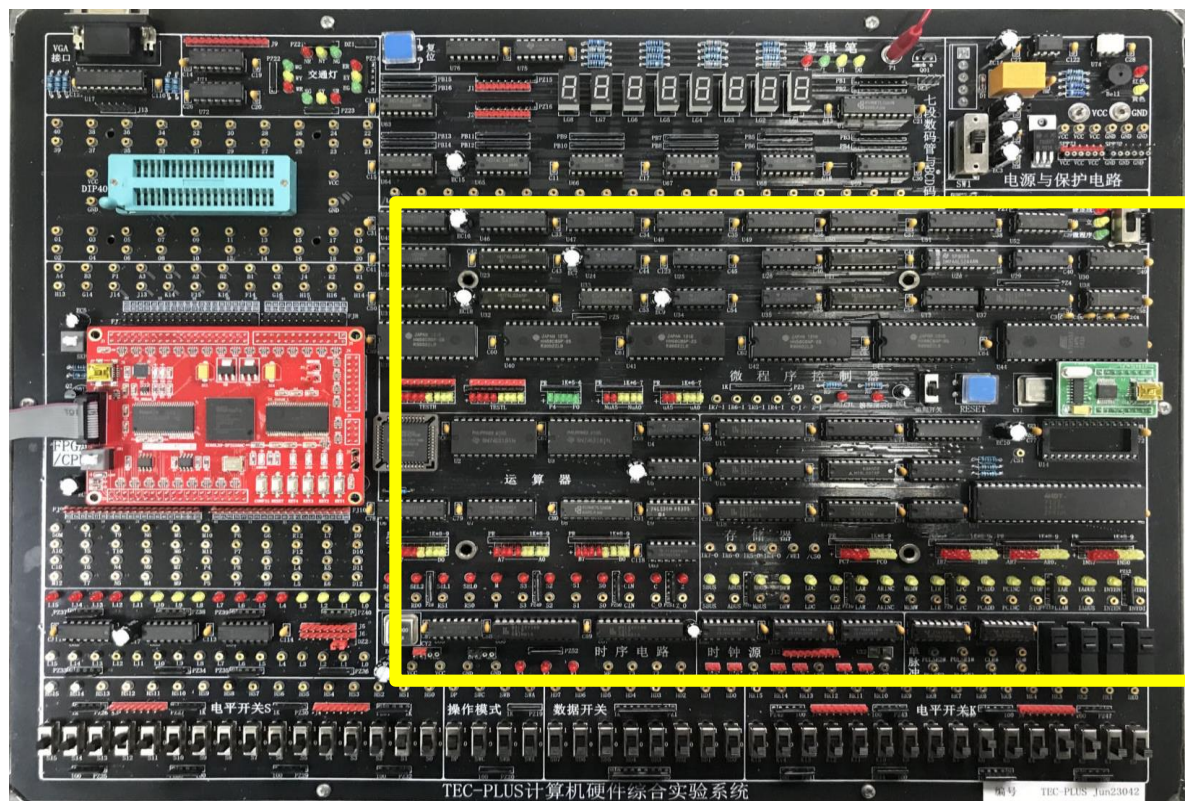
# 实验五



## CPU组成与机器指令的执行

# 实验五

- 实验目的
- 实验电路
- 实验任务
- 实验步骤
- 实验要求

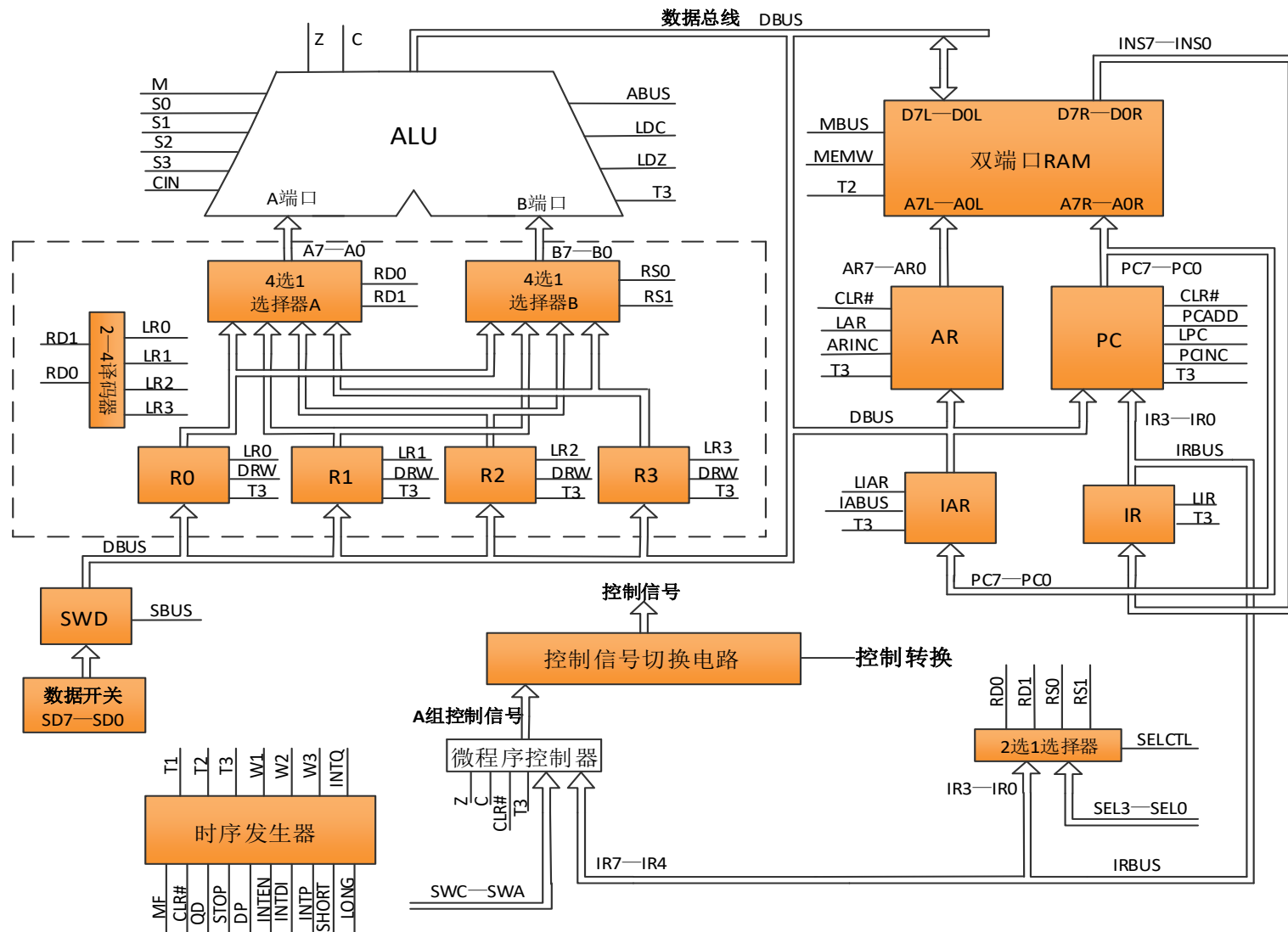


# 实验五

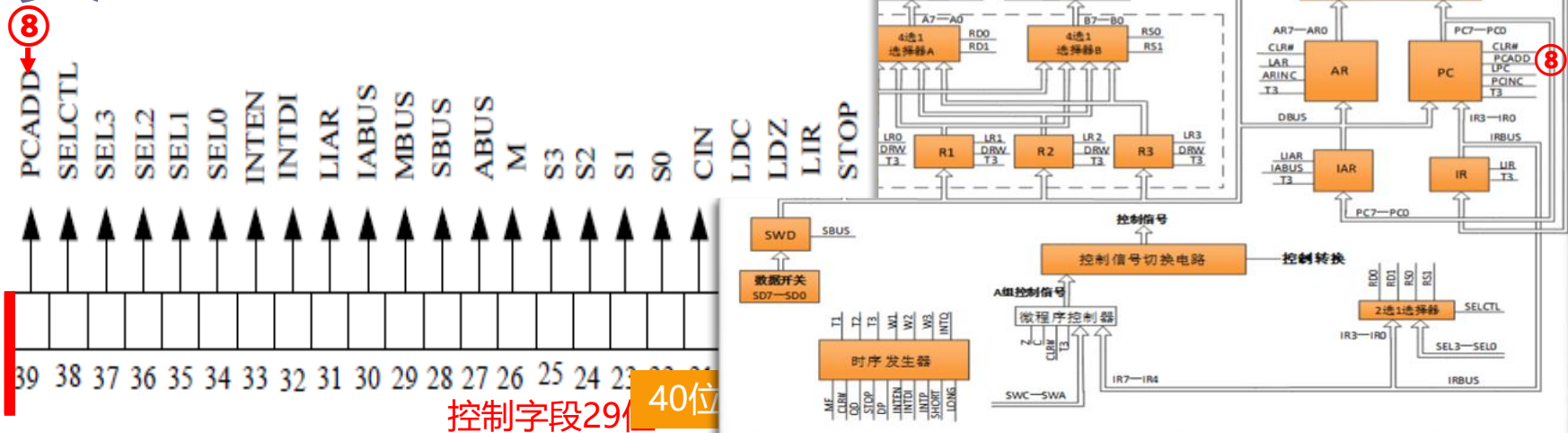
## 实验目的

- ① 用微程序控制器控制数据通路，将相应的信号线连接，构成一台能够运行测试程序的CPU；
- ② 执行一个简单的程序，掌握机器指令与微指令的关系；
- ③ 理解计算机如何取出指令、如何执行指令、如何在一条指令执行结束之后自动取出下一条指令并执行，从而牢固建立计算机整机概念。

# 实验五 | 实验电路 >> 概览



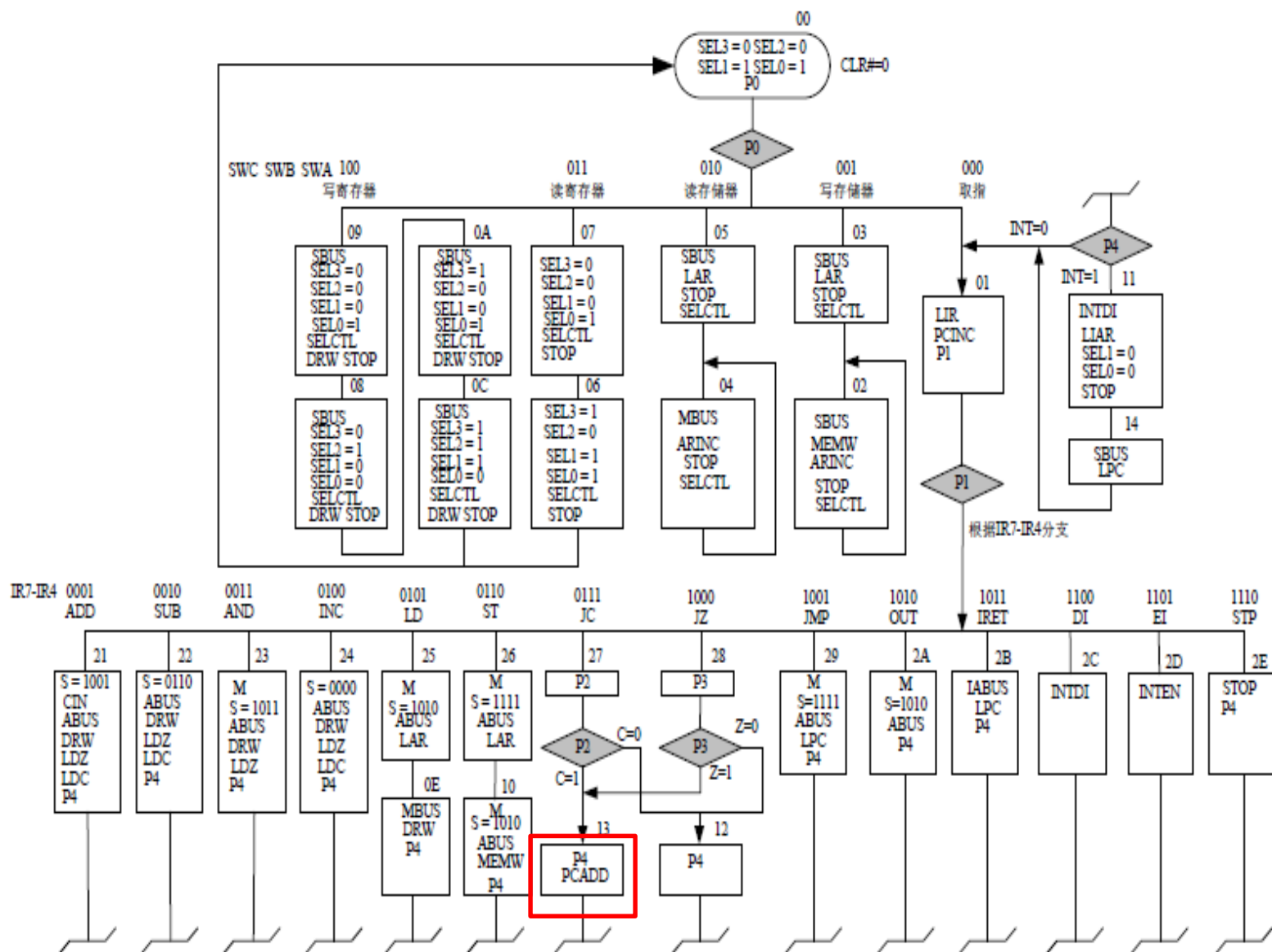
# 实验五 | 实验电路 >> 微指令



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。

## 课前复习

## 实验四 >> 微程序流程



# 实验五 | 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。

# 实验五 | 实验任务

- ① 完成对给定程序的手工汇编；
- ② 通过简单的连线构成能够运行程序的TEC-8模型计算机；
- ③ 将程序写入存储器，给寄存器R2、R3赋初值；
- ④ 跟踪执行程序，用单拍方式运行一遍，用连续方式运行一遍，详细记录实验过程及结果；
- ⑤ 用实验台操作检查程序运行结果。



# 实验五 | 实验任务

地址	指令	二进制机器代码
00H	LD R0,[R3]	0101 0011
01H	INC R3	
02H	LD R1,[R3]	
03H	SUB R0,R1	
04H	JZ 0BH	1000 0110
05H	ST R0,[R2]	0110 1000
06H	INC R3	
07H	LD R0,[R3]	
08H	ADD R0,R1	
09H	JC 0CH	

地址	指令	二进制机器代码
0AH	INC R2	
0BH	ST R2,[R2]	
0CH	AND R0,R1	
0DH	OUT R2	
0EH	STP	
0FH	85H	
10H	23H	
11H	EFH	
12H	00H	

# 实验五 | 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 参考连线：

数据通路	IR4-I	IR5-I	IR6-I	IR7-I	C-I	Z-I
电平开关	IR4-O	IR5-O	IR6-O	IR7-O	C-O	Z-O

# 实验五 | 实验步骤

## □ 单拍方式

- ① 打开电源；
- ② 通过写存储器操作将手工汇编的程序写入存储器；
- ③ 通过读存储器操作，将指令逐条读出，检查写入是否正确；
- ④ 通过写寄存器操作，将数12H写入寄存器R2，0FH写入寄存器R3；
- ⑤ 通过读寄存器，检查写入寄存器的数据是否正确；
- ⑥ 将单拍开关DP设置为1，按下复位按钮CLR，在单微指令下运行程序并记录过程数据；
- ⑦ 读取四个寄存器的值并记录；
- ⑧ 读取存储器12H和13H的值并记录。

注意：实验过程中根据下一页的表格观察指示灯状态并记录。

# 实验五 | 实验步骤

## □ 连续方式

- ① 通过写存储器操作将手工汇编的程序写入存储器；（单拍已做且无修改无需重复）
- ② 通过写寄存器操作，将数12H写入寄存器R2，0FH写入寄存器R3；
- ③ 将存储器12H号存储单元的值改为00H；
- ④ 将单拍开关DP设置为0，按复位按钮CLR，在连续方式下运行程序；  
✓ （按一次QD，程序自动运行到STP指令）
- ⑤ 读取四个寄存器的值并记录；
- ⑥ 读取存储器12H和13H的值并记录。

# 实验五 | 实验要求

## □ 做好预习

- 务必在实验课前，完成对程序的手工汇编！

## □ 填写实验记录表

- 只记录单拍方式下程序执行过程，每按下QD，记录一条数据；

## □ 认真完成实验报告，要求包括：

- 程序的手工汇编结果；
- 实验结果记录表；
- 关键步骤的截图。

# 实验六



## 中断原理实验

# 实验六 | 基本概念

## □ 中断

- 在计算机执行程序的过程中，出现某些急需处理的异常情况或特殊请求，CPU暂时停止现行程序，而转去对这些异常情况或特殊请求进行处理，处理完毕后CPU又自动返回到现行程序的断点处，继续执行源程序。

## □ 中断向量

- 中断服务程序的入口地址

## □ 中断屏蔽

- 关闭中断



# 实验六 | 实验原理 >> 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

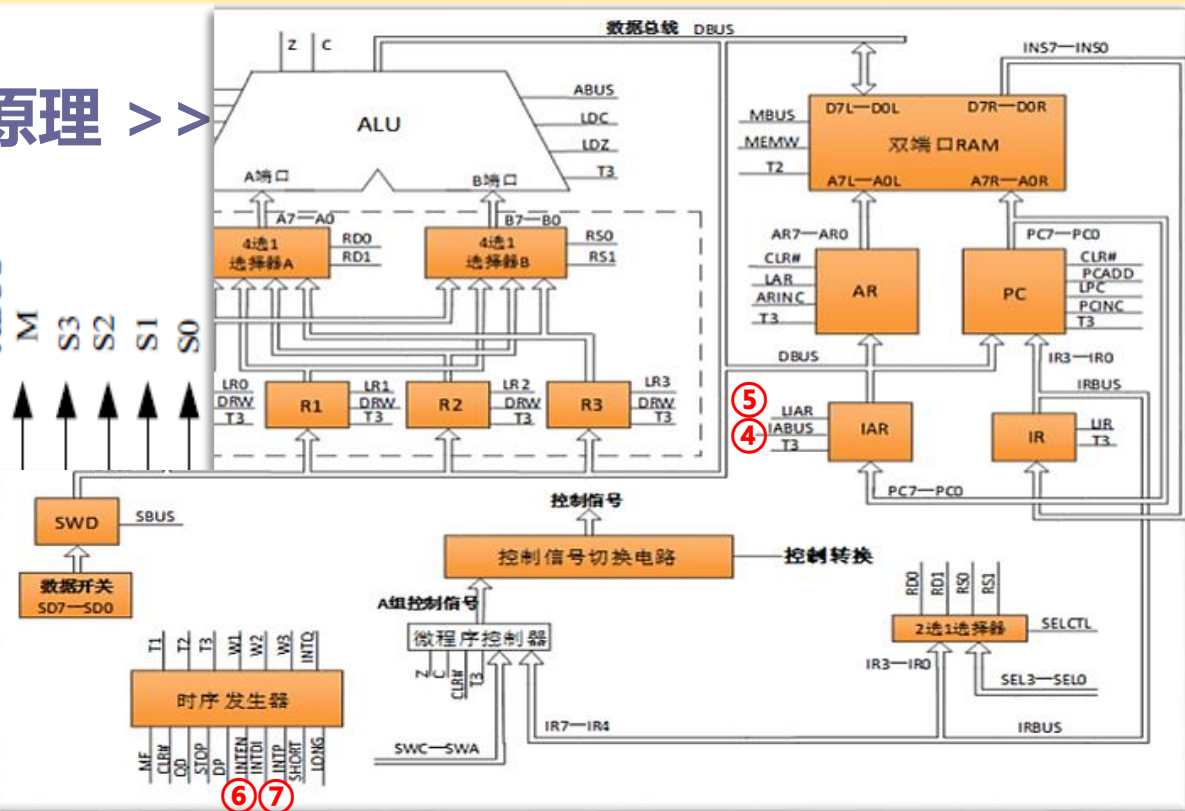
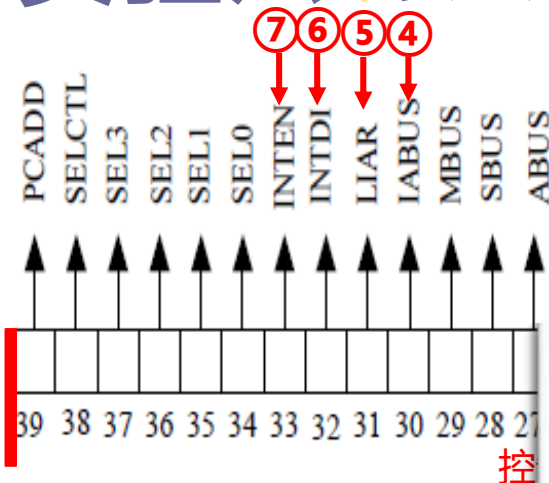
注:

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。



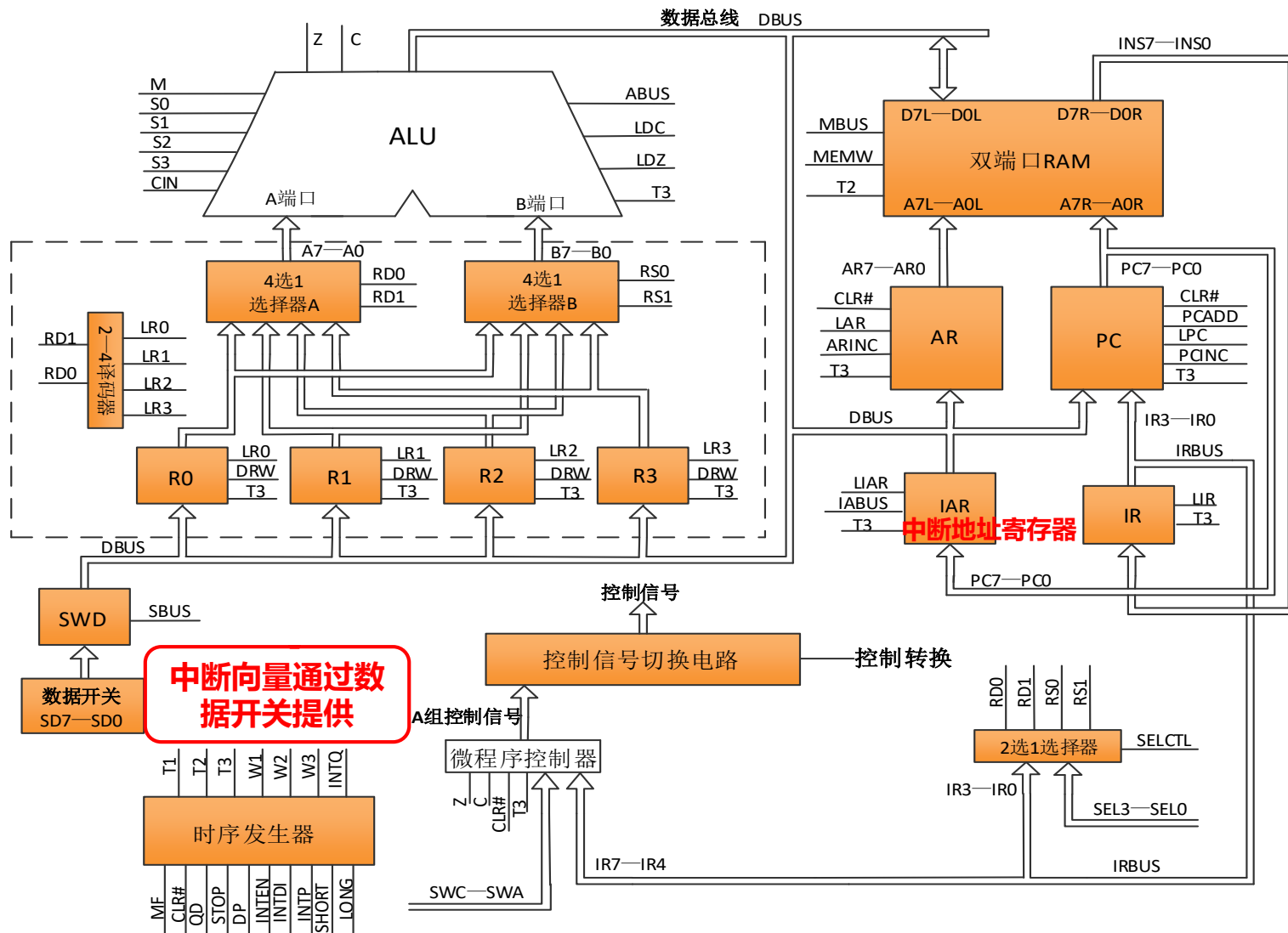


# 实验六 | 实验原理 >>



序号	字段	
3	STOP	=1时, 在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。

# 实验六 | 实验电路

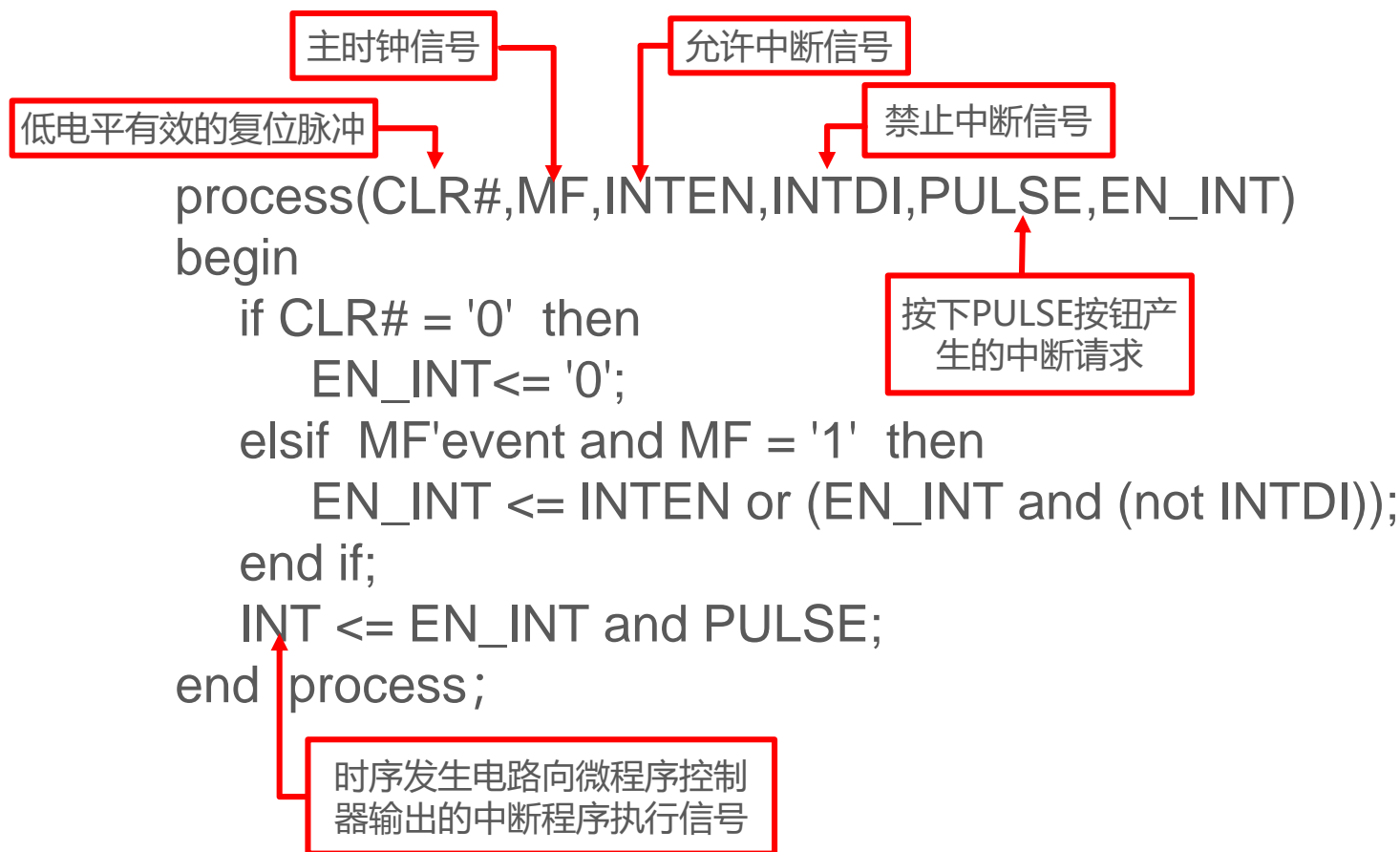


# 实验六 | 实验原理

- 中断在时序发生器中，设置了一个允许中断触发器EN\_INT
  - 当它为1时，允许中断
  - 当它为0时，禁止中断
  - 复位脉冲CLR#使EN\_INT复位为0
- EN\_INT为1，允许中断时，按下PULSE按钮产生的中断请求，INT=1

# 实验六 | 实验原理

## □ 使用VHDL 语言描述的TEC-8中的中断触发器如下：



# 实验六 | 实验任务

## 主程序机器代码

地址	指令	二进制机器代码
00H	EI/DI	
01H	INC R0	
02H	INC R0	
03H	INC R0	
04H	INC R0	
05H	INC R0	
06H	INC R0	
07H	INC R0	
08H	INC R0	
09H	JMP [R1]	

## 中断服务程序机器代码

地址	指令	二进制机器代码
45H	ADD R0,R0	
46H	EI	
47H	IRET	

# 实验六 | 实验任务

- 理解中断相关指令，以及每个信号的意义和变化条件；
- 将主程序和中断服务程序手工汇编成二进制机器代码；
- 通过简单的连线构成能够运行程序的TEC-8模型计算机；
- 将主程序和中断服务程序装入存储器，给寄存器R1赋初值01H，R0赋初值0；
- 执行三遍主程序和中断服务程序，详细记录中断有关信号变化情况，特别记录好断点和R0的值；
- 将主程序中地址为00H的EI指令改为DI，重新运行程序，记录现象。

# 实验六 | 实验任务

主程序机器代码

①

地址	指令
00H	EI
01H	INC R0
02H	INC R0
03H	INC R0
04H	INC R0
05H	INC R0
06H	INC R0
07H	INC R0
08H	INC R0
09H	JMP [R1]

R0=00H  
R1=01H

开中断

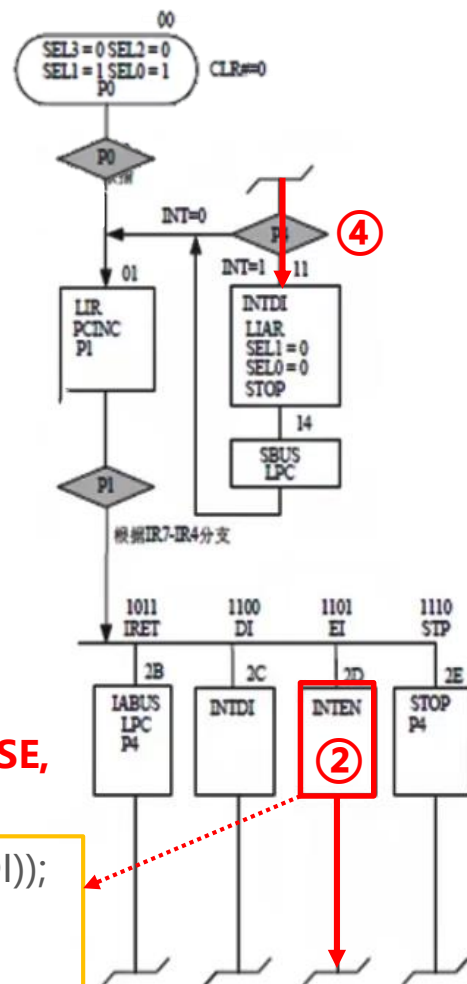
中断服务程序机器代码

地址	指令
45H	ADD R0,R0
46H	EI
47H	IRET

③ 主程序运行过程中，只要出现了中断信号PULSE，那么INT信号有效。

$EN\_INT \leq INTEN \text{ or } (EN\_INT \text{ and } (\text{not } INTDI));$

$INT \leq EN\_INT \text{ and } PULSE;$



# 实验六 | 实验任务

主程序机器代码

地址	指令
00H	EI
01H	INC R0
02H	INC R0
03H	INC R0
04H	INC R0
05H	INC R0
06H	INC R0
07H	INC R0
08H	INC R0
09H	JMP [R1]

R0=00H  
R1=01H

中断服务程序机器代码

地址	指令
45H	ADD R0,R0
46H	EI
47H	IRET

开中断

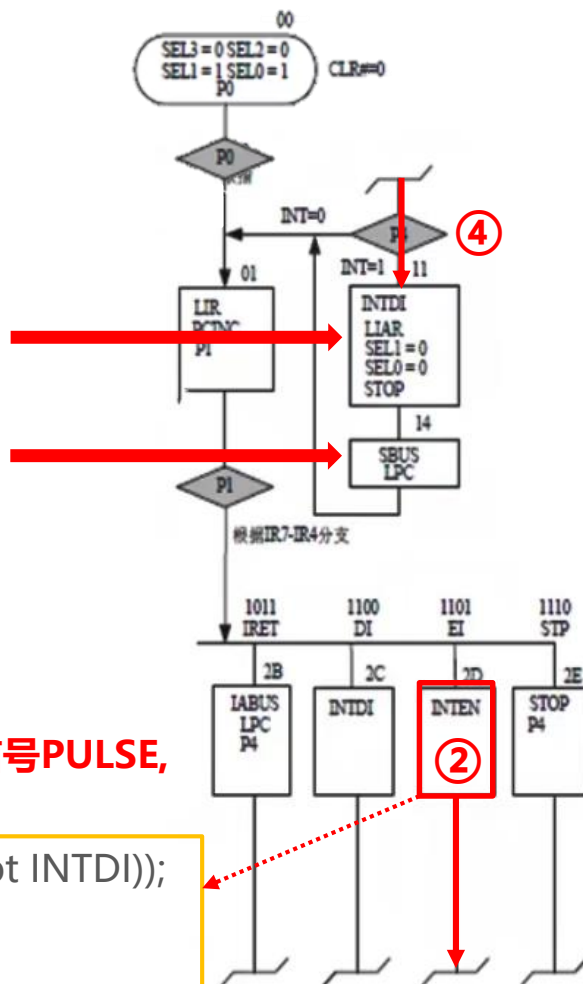
⑤ 置INTDI信号有效，拒绝中断；  
保存中断地址至中断地址寄存器。

⑥ 接受中断向量45H（来自数据开关）；  
将中断向量送给PC。

③ 主程序运行过程中，只要出现了中断信号PULSE，  
那么INT信号有效。

$EN\_INT \leq INTEN \text{ or } (EN\_INT \text{ and } (\text{not } INTDI));$

$INT \leq EN\_INT \text{ and } PULSE;$





# 实验六 | 实验任务

主程序机器代码

地址	指令
00H	EI
01H	INC R0
02H	INC R0
03H	INC R0
04H	INC R0
05H	INC R0
06H	INC R0
07H	INC R0
08H	INC R0
09H	JMP [R1]

R0=00H

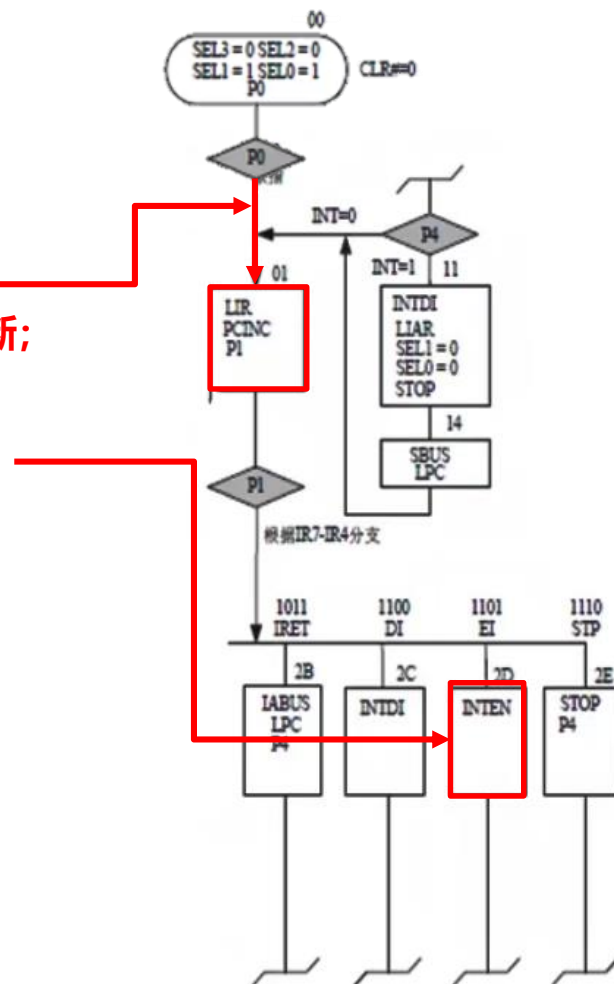
R1=01H

中断服务程序机器代码

地址	指令
45H	ADD R0,R0
46H	EI
47H	IRET

⑦ 之后，进入中断服务程序执行；  
在开中断之前，不允许再次中断；

⑧ 开中断，置INTEN有效；  
此后可以继续响应中断；



# 实验六 | 实验任务

主程序机器代码

地址	指令
00H	EI
01H	INC R0
02H	INC R0
03H	INC R0
04H	INC R0
05H	INC R0
06H	INC R0
07H	INC R0
08H	INC R0
09H	JMP [R1]

R0=00H

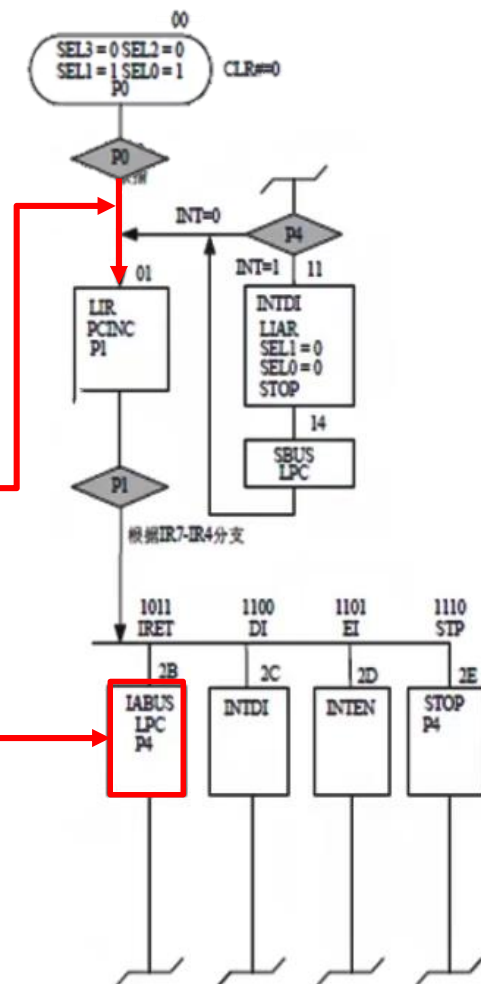
R1=01H

中断服务程序机器代码

地址	指令
45H	ADD R0,R0
46H	EI
47H	IRET

⑨ 中断返回。  
从中断寄存器中取出主程序中断地址；  
将该地址送给PC；

⑩ 返回主程序断点处继续执行；



# 实验六 | 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 参考连线：

数据通路	IR4-I	IR5-I	IR6-I	IR7-I	C-I	Z-I
电平开关	IR4-O	IR5-O	IR6-O	IR7-O	C-O	Z-O

# 实验六 | 实验步骤

- 打开电源;
- 通过控制台写存储器操作, 将主程序和中断服务程序写入存储器。
- 执行3遍主程序和中断子程序
  - ① 通过控制台写寄存器操作将R0设置为00H, 将R1设置为01H。
  - ② 单拍开关DP=0, 按复位按钮CLR。按QD按钮, 启动程序从00H开始执行。
  - ③ 按PULSE按钮, 产生一个中断请求信号PULSE, 中断主程序的运行。记录下这时的断点PC、R0(指示灯A7~A0上显示)的值。
  - ④ 单拍开关DP=1, 在数据开关上设置中断服务程序的入口地址45H。按QD按钮, 一步步执行中断服务程序, 直到返回到断点为止。
  - ⑤ 按照步骤(1)~(4), 再重复做2遍。
- 将存储器00H的指令改为DI, 按照步骤3, 重做一遍, 记录发生的现象。

# 实验六 | 实验要求

## □ 做好预习

- 务必在实验课前，完成对程序的手工汇编！

## □ 填写实验记录表

# 实验六 | 拓展要求

- 对实验五的程序进行改造，在**不改变其逻辑功能、指令执行顺序的前提下**，使之成为**主程序**完成中断功能，结合实验六的中断服务程序，将两个实验综合起来一并完成。
  - 可以根据需要改变寄存器的操作数，但不能改变原程序的执行顺序。
  - 可以根据需要在主程序中加入一些中断相关指令，但不能改变原程序的逻辑。
  - 可以根据需要改变原程序最后几个地址单元的数据。
  - 通过 JMP 指令，使主程序处于循环状态。
  - .....

# 实验六 | 拓展要求

- 只需完成一份实验报告，包括：
  - 新设计的主程序的指令序列以及手工汇编结果。
  - 中断服务程序的手工汇编结果。
  - 单拍方式下的过程记录表：每按下QD，记录一条数据。
  - 连续方式下的中断情况：PC断点值、寄存器数据等。
  - 关键步骤的截图。