**上海大学 计算机学院**

**《计算机组成原理实验》报告10**

**姓名 徐驰 学号 20124652**

**时间 周四9-11 机位 4 指导教师 刘学民**

**实验名称: 中断机制和应用（综合实验）**

**一、实验目的**

1.学习实验箱感知中断的硬件结构和工作原理

2.学习使用中断系统

3.学习使用扩展外设

4.深入理解中断系统的重要性

5.深入理解中断系统的工作原理

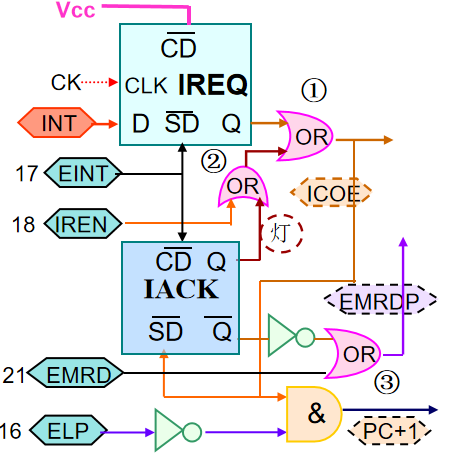
**二、实验原理**

1.程序中断

中断是程序执行过程中发生随机事件时，暂停执行原程序，转而执行另一个程序来处理该事件，然后再返回原程序继续执行的过程。与子程序调用相似，中断也需要保存断点，但不同之处在于中断发生的时间是随机的，所以必须用硬件感知中断请求，并且保存现场。中断的随机性使得系统可以在运行一个程序的同时感知并处理其他事件，这是实时监控、多用户、多任务、多线程技术的关键，也是操作系统工作的前提，是计算机系统的重要组成部分。对于计算机专业人员来说，深刻理解中断系统是使用计算机的必备知识。

2.实验箱中断感知硬件

实验箱部分硬件逻辑结构如图：



“中断返回”指令和复位操作使得EINT为低电平，导致IREQ的Q端为1，IACK的Q端为0。我们称其为状态0。

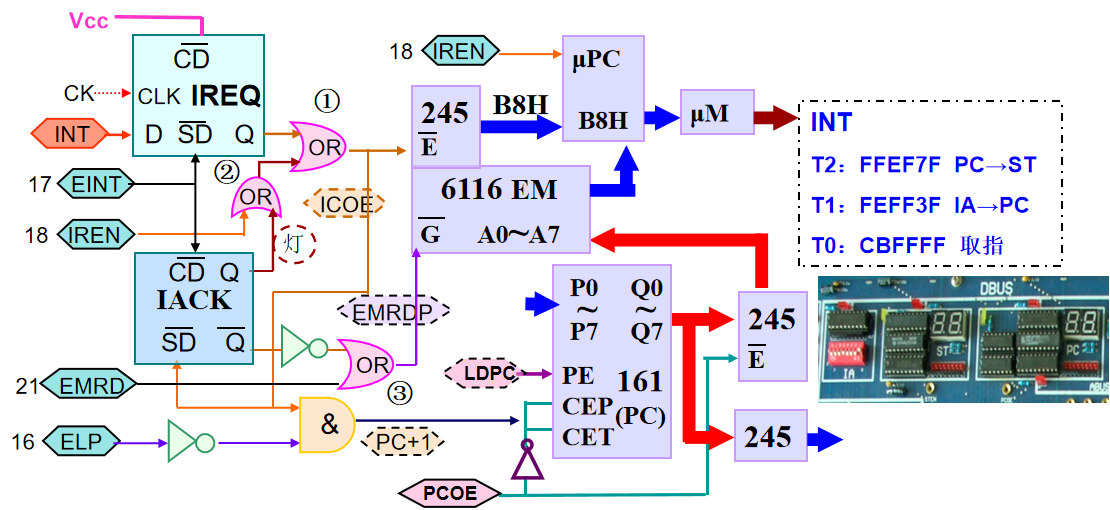
复位结束或执行其他指令时，EINT为高电平，在时钟CK驱动下，IREQ的Q端与INT端状态相同。中断请求发生，INT置为0，IREQ的Q端在1个CK后变为0。此时，若IREN为1，则中断信号被IREN拦截；只有当IREN为0，也就是取指令阶段时，中断信号才会被CPU感知。

当中断被感知后，也即①发出低电平信号。可以顺着ICOE所在电路，令IACK的SD端为低电平，从而IACK的Q端输出1，导致INT信号继续被拦截。从而，当系统进入中断服务子程序后，不会响应中断请求。直至中断服务子程序执行“中断返回”指令RETI后，EINT为0，系统进入状态0，才会在下一次IREN为0时将中断信号传给CPU。

在中断服务期间，若INT持续有效（INT=1），那么在中断返回后，将会在取指令阶段又立即进入中断。会形成“中断未返回”的现象。

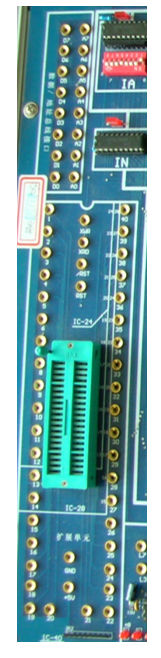
接下来，看另一部分，从而理解PC值的改变原理：

这是实验箱中断部分更完整的硬件逻辑结构：



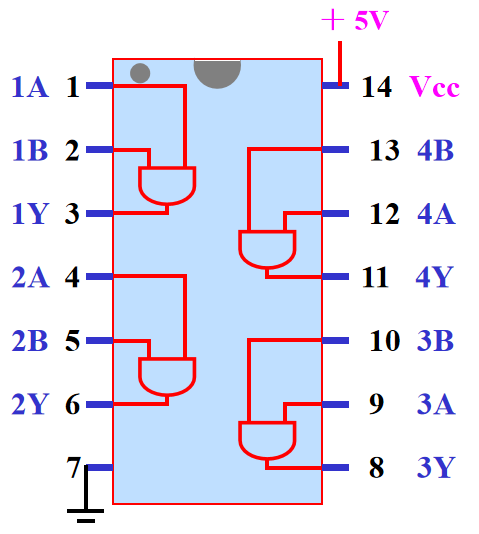
当中断被CPU感知后，ICOE会置为0，从而PC不变。当进入中断服务子程序后，ICOE为1，PC可以进行自动加1的功能。从而让中断服务子程序继续正常运行。此外，中断被感知时，ICOE为0时，会通过③或门令EM不输出；中断被感知，输出中断指令B8H到μPC，μPC再将中断微指令（INT）的控制信号输出。如图，INT的功能即将PC中的值存入ST寄存器，保存断点，IA中的值进去PC，进入中断服务子程序，其中IA的值就是中断服务子程序的进入地址。

3.实验箱外扩系统



如图，本次实验将会在上面插上74LS08芯片，完成一个与门电路的搭建。

4.74LS08逻辑结构



如图是74LS08的逻辑结构图，我们实验中用1A1B作为输入，将1Y作为输出。相当于实现了1A 与 1B 输出到 1Y的电路。

**三、实验内容**

**1．实验任务一：任选芯片74LS08中的一个与门，K1、K2接与门的输入，与门的输出作为INT信号。**

1. 实验步骤

1.打开实验箱。

2.将74LS08芯片放在外扩系统的芯片插槽上。

3.接线：将下方GND端口连接到7号端口，将+5V连接到40号端口，K1K2分别连接到1号和2号端口，3号端口接到右侧INT端口上。

4.改变K1K2的值，观察INT旁小灯的亮灭情况。

5.保留实验箱状态，以备任务二的使用。

1. 实验现象

在第4步操作中：将K1K2置11时，发现INT旁小灯亮起；将K1K2置00，01,10时，发现INT旁边小灯灭掉。

1. 实验结论

完成了既定的实验任务，实验现象符合要求。

**2．实验任务二：编制中断服务子程序使OUT交替显示AA、BB三次后返回源程序。源程序为交替显示11、55和66的程序。**

**要求：运行上述程序，在完成AA、BB交替显示三次之前恢复K1K2都为1的状态，记录OUT显示的现象、REQ灯和ACK灯的情况以及ST寄存器的值及改变情况。并且：R0显示中断时AA、BB显示的次数；R1跟踪主程序OUT显示的值；R2跟踪主程序延时的值。**

1. 实验步骤

1.打开WindowsPC机，运行编译软件CP226。

2.连接实验箱。

3.打开汇编代码.asm文件。

4.编译下载。

5.在实验箱左上角的IA寄存器处，将中断子程序入口地址设为60H（01100000）。

5.运行，观察主程序的执行情况。

6.在某一随机时间将K1或者K2拨至0，观察中断指令执行情况，以及中断子程序执行情况。

7.在中断服务子程序执行完成之前，将K1K2回拨置11，观察中断返回后的一系列现象。

8.整理器材置初始状态，实验结束。

1. 实验现象

1.第5步执行后，发现OUT寄存器交替显示11H、55H、66H，此时REQ、ACK灯都不亮，ST无变化。R0显示00H，R1内容和OUT寄存器基本同步，R2中从20H一直迅速减1直至00H，然后立即变回20H继续减1直至00H，如此往复。

2.第6步执行后，发现REQ、ACK灯亮起，ST中显示断点地址，我的实验中显示了08H，OUT寄存器中，交替显示AAH,BBH。R0显示03H，然后一段时间后减1，减到0后，中断程序就退出了。

3.第7步执行后，发现REQ、ACK灯都灭掉，回退到1第6步执行前的状态。

1. 数据记录、分析与处理

1.主程序（源程序）汇编代码

**ORG 00H**

**LOOP: MOV A,#11H**

**MOV R1,A**

**OUT**

**MOV A,#20H**

**Z1: SUB A,#01H**

**MOV R2,A**

**JZ Z2**

**JMP Z1**

**Z2: MOV A,#55H**

**MOV R1,A**

**OUT**

**MOV A,#20H**

**Z3: SUB A,#01H**

**MOV R2,A**

**JZ Z4**

**JMP Z3**

**Z4: MOV A,#66H**

**MOV R1,A**

**OUT**

**MOV A,#20H**

**Z5: SUB A,#01H**

**MOV R2,A**

**JZ Z6**

**JMP Z5**

**Z6: JMP LOOP**

2.中断服务子程序汇编代码

**ORG 60H**

**MOV R0,#03H**

**LOOP1: MOV A,#AAH**

**OUT**

**MOV A,#10H**

**T1: SUB A,#01H**

**JZ T2**

**JMP T1**

**T2: MOV A,#BBH**

**OUT**

**MOV A,#10H**

**T3: SUB A,#01H**

**JZ T4**

**JMP T3**

**T4: SUB A,#01H**

**JZ T4**

**JMP T3**

**MOV A,R0**

**SUB A,#01H**

**JZ EXIT**

**MOV R0,A**

**JMP LOOP1**

**EXIT: MOV A,R1**

**OUT**

**MOV A,R2**

**SUB A,#00H**

**RETI**

1. 实验结论

1.完成了既定的实验任务，且实验现象符合要求。

**四、建议和体会**

在这个实验中，我学习了实验箱感知中断的硬件结构和工作原理，以及如何使用中断系统和扩展外设。我也深入理解了中断系统的重要性和工作原理，这些知识对于我今后的学习和工作都非常有用。

在任务一中，我选了芯片74LS08中的一个与门，K1、K2接与门的输入，与门的输出作为INT信号。这个任务让我更加熟悉了硬件的连接方式和信号传输。

在任务二中，我编制了中断服务子程序，使OUT交替显示AA、BB三次后返回源程序。然后我运行了该程序，在完成AA、BB交替显示三次之前恢复了K1K2都为1的状态，并记录了OUT显示的现象、REQ灯和ACK灯的情况以及ST寄存器的值及改变情况。同时，我也用R0显示中断时AA、BB显示的次数，用R1跟踪主程序OUT显示的值，用R2跟踪主程序延时的值。

通过这个实验，我学到了如何编写中断服务子程序和如何处理中断请求。我也理解了中断系统是如何工作的，并且学会了如何使用扩展外设来实现更多的功能。这些知识对于我今后的学习和工作都非常有用，让我更加深入地了解了计算机系统的内部工作原理。

在今后的实验中，我将更加努力地学习和实践，以便更好地掌握这些知识。我也会积极思考如何将这些知识应用到实际的项目中，以便更好地提高我的技能水平。

**五、思考题**

**实验箱的中断服务程序中可以嵌套一般的子程序吗？**

答：不可以。如果是针对我们所用的这台实验箱，那么就不可以。我们所用的实验箱的堆栈寄存器容量只有1个地址，在实验箱发生中断进入中断服务程序中时，ST寄存器中就保存了中断发生时的断点，若中断服务程序中嵌套有子程序，那么在进入子程序时，ST寄存器就会被新的断点所覆盖，源程序的中断断点信息就会丢失，中断服务子程序进入RETI指令后，就无法返回到正确的源程序位置。

当然，如果堆栈寄存器ST的容量足够大，那么理论上中断服务程序是可以嵌套一般的子程序的，但仍然是不推荐的一种做法，在中断服务程序中嵌套子程序可能会影响中断的响应时间和中断处理的正确性。因为中断服务程序需要尽快地响应中断请求并处理中断事件，所以在中断服务程序中调用复杂的子程序可能会导致中断响应时间过长，甚至错过了其他重要的中断请求，从而影响系统的性能和可靠性，因此不建议在中断服务程序中嵌套复杂的子程序。