

上海大学 计算机学院

《计算机组成原理实验》报告十

姓名 严昕宇

学号 20121802

时间 2022. 05. 10

机位 19

指导教师 刘跃军

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

二、实验原理

1. 程序中断

因“随机性”原因，使一个程序暂停执行，转而执行另一个程序，以处理随机事件，然后再返回原程序继续执行的过程成为“中断”。

中断发生的“随机性”决定了“必须用硬件感知中断请求”、“不仅要保存断点，还必须保存现场”。中断发生时间与正在运行的程序的无关性，使得整个系统在运行一个程序的同时，还能感知其它事件的发生。

2. 实验箱系统的程序转移硬件机制

只有“中断返回”指令和复位操作使 EINT 为低电平，作用到 IREQ 的 SD 端，使上面 D 触发器的 Q 端为 1，使下面 D 触发器的 Q 端输出 0。

系统复位结束或执行其他指令时，EINT 为无效的高电平，在时钟 CK 驱动下，IREQ 的 Q 端输出 D 端的 INT 状态。IREQ 的 Q 端无论输出 0 或 1，①号“或门”总输出 1，即不允许中断请求通过。同时 1 又送入 IACK 的 SD 端；于是下触发器的 SD 和 CD 端的输入都是无效状态，触发器保持稳定。

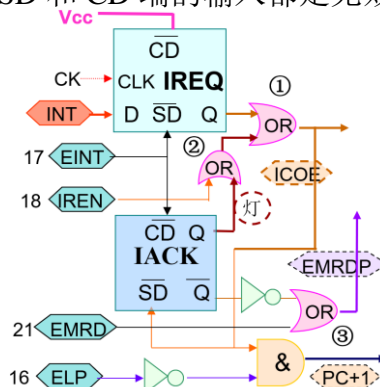


图 1. 实验箱中断感知硬件的逻辑结构

当执行取指微指令时，IREN=0，于是②号或门输出 0，这时①号“或门”对 IREQ 的 Q 端开放，若有中断请求就会在这时被 CPU 感知。即无论中断请求在何时提出，都只能在取指阶段被感知。

系统进入中断服务子程序后，ACK 灯保持亮，且不响应新的中断请求（仅一级中断）。直到中断服务程序执行“中断返回”指令 RETI 时，EINT 为 0，使系统再次进入状态 0。在中断服务期间中断源若不及时撤销中断请求（使 INT 为 1），则中断返回后系统会再次进入中断状态，造成“中断未返回”假象。

ICOE 向下经“与门”控制 PC+1 信号，如下图。当感知中断请求（ICOE=0）时，使 PC 值不变（确定断点），进入服务程序后（ICOE=1），PC 恢复自动加 1 功能，保证服务程序的顺序执行。ICOE 另一个作用是通过③号“或门”控制 EM 中指令的输出，保证在输出中断指令 B8H 时，EM 不输出。

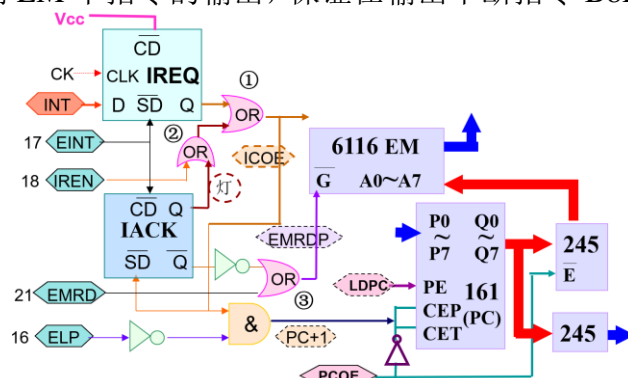


图 2

ICOE 有效时，使 EM 的锁存器输出中断指令 B8H 到 μ PC， μ PC 再把 μ M 中的中断微指令（INT）的控制信号输出。INT 指令在第一个周期保存 PC 值（断点）到 ST 堆栈寄存器。在第二个周期把 IA 寄存器的内容送 PC，即 IA 中应事先存入中断服务子程序的入口地址。所以 IA 称为中断向量寄存器，IA 的值由红色的码盘输入。

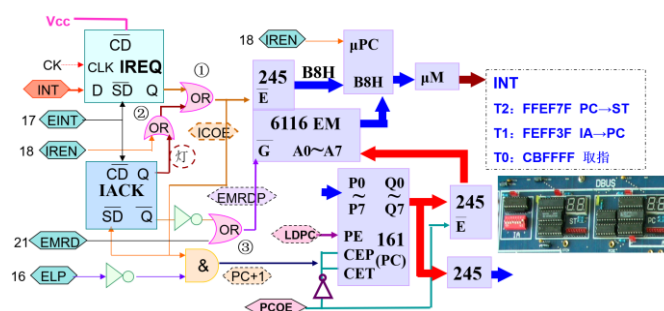
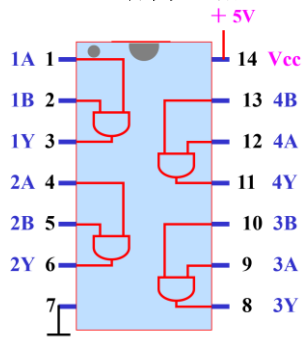


图 3

3. 实验箱外扩系统



4. 74LS08 结构: 俯视图



三、实验内容

1. 实验任务一（用 74LS08 芯片搭建当电键 K1 和 K2 都为 1 时不产生中断请求信号的外部电路）

(1) 实验步骤

- ① 打开实验箱电源，检验实验箱的基本功能是否正常。确认无误后，打开实验箱，按三下 TV-ME 按键，将实验箱调整到手动模式。
- ② 将 74LS08 芯片接入实验箱外部扩展系统，用导线将接口 1、2 与开关 K1、K2 相连接，接口 3 连接 INT。接口 7 连接地端 GND，接口 40 连接 VCC/+5V。

接口 1	接口 2	接口 3	接口 7	接口 40
K1	K2	INT	GND	VCC

- ③ 按照下表给出对应的信号，拨动开关 K1、K2，观察不同的输入下 INT 信号灯的情况：

接口 1	接口 2	接口 3	接口 7	接口 40
K1	K2	INT	GND	VCC
0 或 1	0 或 1			

- ④ 记录实验现象并分析。

(2) 实验现象

当且仅当 K1、K2 输入值为 11 时，INT 信号灯亮。

其余三种情况，即 K1、K2 输入值 00、01、10 时，INT 信号灯灭。

(3) 数据记录、分析与处理

K1	K2	INT
0	0	灭
0	1	灭
1	0	灭
1	1	亮

(4) 实验结论

成功在手动模式下，实现子用 74LS08 芯片搭建外部电路，使得当电键 K1 和 K2 都为 1 时不产生中断请求信号的外部电路。

完成实验任务一的目的。

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。源程序为上次实验完成的交替显示 11 和 55 的程序。

(1).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 灯的情况及 ST 寄存器的值及改变情况。

(2).运行上述程序，在完成 AA、BB 交替显示时不恢复 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。

分析上述二种显示现象的原因。[要求：R0：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 打开实验箱电源，检验实验箱的基本功能是否正常。确认无误后，启动 CP226 软件。
- ② 连接 PC 机与实验箱的通信口 COM4。
- ③ 在 CP226 汇编语言程序集成开发环境下编写如下汇编程序：

主程序

```
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:   JMP LOOP
```

中断程序

```
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:   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
      END
```

- ④ 保存文件，文件扩展名为.asm（这一步一定要添加，软件不会自己带上后缀）。
- ⑤ 中断服务程序的入口地址在汇编源程序中用伪指令 **ORG** 指定（此处为 60H），程序运行前把指定的地址值通过红色的码盘送入 IA。
- ⑥ 编译并下载源程序至实验箱，调试并运行程序，观察并记录实验结果。

(2) 实验现象

REQ 为中断请求信号指示灯，ACK 为中断相应信号指示灯；

R0 中存储的是中断时 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 完成 AA、BB 交替显示三次之前恢复 K1K2 都为 1 的状态

主程序执行状态下，OUT 交替显示 11H 和 55H，REQ 灯和 ACK 灯均为熄灭状态；拨动中断请求开关，产生中断请求信号后，REQ 灯和 ACK 灯均为点亮状态。OUT 交替显示 AAH 和 BBH 三次后，REQ 灯和 ACK 灯熄灭，ST 寄存器存储和显示下一步 PC 指令的地址码。

② 完成 AA、BB 交替显示时不恢复 K1K2 都为 1 的状态

主程序执行状态下，OUT 交替显示 11H 和 55H，REQ 灯和 ACK 灯均为熄灭状态；拨动中断请求开关，产生中断请求信号后，OUT 交替显示 AAH 和 BBH，REQ 灯和 ACK 灯始终保持点亮状态，直到中断程序结束。ST 寄存器存储和显示下一步 PC 指令的地址码。

(3) 数据记录、分析与处理

汇编程序的具体分析如下所示

主程序		
	ORG 00H	通知汇编软件主程序从 00H 始存放
LOOP:	MOV A, #11H	将立即数 11H 送至寄存器 A
	MOV R1, A	将寄存器 A 中的值送至寄存器 R1
	OUT	将寄存器 A 中内容送至 OUT
	MOV A, #20H	将立即数 20H 送至寄存器 A
z1:	SUB A, #01H	从寄存器 A 中内容减去立即数 01H
	MOV R2, A	将寄存器 A 中的值送至寄存器 R2
	JZ z2	根据零标志位条件跳转 z2
	JMP z1	无条件跳转 z1
z2:	MOV A, #55H	将立即数 55H 送至寄存器 A
	MOV R1, A	将寄存器 A 中的值送至寄存器 R1
	OUT	将寄存器 A 中内容送至 OUT
	MOV A, #20H	将立即数 20H 送至寄存器 A
z3:	SUB A, #01H	从寄存器 A 中内容减去立即数 01H
	MOV R2, A	将寄存器 A 中的值送至寄存器 R2
	JZ z4	根据零标志位条件跳转 z4
	JMP z3	无条件跳转 z3
z4:	JMP LOOP	无条件跳转 LOOP

中断程序

	ORG 60H	通知汇编软件主程序从 60H 始存放
	MOV R0, #03H	将立即数 03H 送至寄存器 R0
LOOP1:	MOV A, #AAH	将立即数 AAH 送至寄存器 A
	OUT	将寄存器 A 中内容送至 OUT
	MOV A, #10H;	将立即数 10H 送至寄存器 A
T1:	SUB A, #01H	从寄存器 A 中内容减去立即数 01H
	JZ T2	根据零标志位条件跳转 T2
	JMP T1	无条件跳转 T1
T2:	MOV A, #BBH	将立即数 BBH 送至寄存器 A
	OUT	将寄存器 A 中内容送至 OUT
	MOV A, #10H;	将立即数 10H 送至寄存器 A
T3:	SUB A, #01H	从寄存器 A 中内容减去立即数 01H
	JZ T4	根据零标志位条件跳转 T4
	JMP T3	无条件跳转 T3
T4:	MOV A, R0	将寄存器 R0 中的值送至寄存器 A
	SUB A, #01H	从寄存器 A 中内容减去立即数 01H
	JZ EXIT	根据零标志位条件跳转 EXIT
	MOV R0, A	将寄存器 A 中的值送至寄存器 R0
	JMP LOOP1	无条件跳转 LOOP1
EXIT:	MOV A, R1	将寄存器 R1 中的值送至寄存器 A
	OUT	将寄存器 A 中内容送至 OUT
	MOV A, R2	将寄存器 R2 中的值送至寄存器 A
	SUB A, #00H	从寄存器 A 中内容减去立即数 00H
	RETI	中断返回
	END	汇编程序终止
		(如果不加程序不会停止)

(4) 实验结论

成功利用 CP226 软件编写汇编程序完成实验任务二, 实现了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。

完成实验任务二的目的。

四、建议和体会

体会

通过本次实验，我更好地理解中断系统的工作流程。在实验 2 中，通过亲自编写程序，手动引发中断，模拟了中断请求的处理过程。在编写汇编代码的过程中，我一开始因为错误地设置了两个 **LOOP** 标志导致错误的产生，后来在检查代码时才发现这个问题，这给我留下了深刻的印象。正是因为无法通过实验箱检测代码的正确性，才需要我在书写代码的时候格外注意。我相信在以后的代码编写过程中，会有更好的编程规范，不会再犯相同的错误。与此同时，随着实验次数的增加，与理论课知识的教授，我认为自己对计算机组成原理的认识越发全面，做实验时也更加清楚地认识到每一步操作的意义与重要性。

由于疫情的影响，本次实验是居家完成。通过观看线上的实验讲解视频与学习实验内容 PPT，并未能亲身操作实验箱和 CP226 软件来进行实验，这会在一定程度上影响学习效果，因此需要课前高质量的预习与课后的总结以加深理解。并且如果有机会，在返回学校进入实验室后，应该重新实验以验证自己实验报告中的内容。

五、思考题

实验箱的中断服务程序中是否可以嵌套一般的子程序吗？

答：各系统实现堆栈的技术各不相同。学校使用的 CP226 实验箱系统用一个锁存器（574 芯片）构成堆栈寄存器（ST）。由于 574 芯片只能存放一个字节，所以，本系统的子程序调用深度只有 1 级，不能形成子程序嵌套。