

电子信息工程系实验报告

课程名称：单片机原理及接口

实验项目名称：中断系统实验

实验时间：2011. 11. 28

班级：通信

姓名：

学号：

成绩：

指导教师（签名）：

实验目的：

熟悉 keil 仿真软件、proteus 仿真软件的使用和单片机外部中断的使用。了解并熟悉 51 单片机中中断的概念，中断处理系统的工作原理。理解 51 单片机中断管理系统处理五种中断源，特别是对外部中断的设置与控制方法。熟悉中断处理特别是外部中断处理的过程，掌握中断处理子程序的书写格式和使用方法。

实验环境：

KEIL C51 单片机仿真调试软件，proteus 系列仿真调试软件

实验内容及过程：

所谓中断是指，单片机内部有一个中断管理系统，它对内部的定时器事件、串行通信的发送和接收事件及外部事件（如键盘按键动作）等进行自动的检测判断，当有某个事件产生时，中断管理系统会置位相应标志通知 CPU，请求 CPU 迅速去处理。CPU 检测到某个标志时，会停止当前正在处理的程序流程，转去处理所发生的事件（针对发生的事件，调用某一特定的函数，称为该事件的中断服务函数），处理完以后，再回到原来被中断的地方，继续执行原来的程序。这个过程称为中断。（CPU 对中断标志的检测是在程序指令执行的周期中顺带进行的，不影响指令的连续执行。）

中断管理系统可以处理的事件称为中断源。一般计算机系统允许有多个中断源，当几个中断源同时向 CPU 请求中断，要求为它们服务的时候，就存在 CPU 优先响应哪一个中断请求源的问题，一般根据中断源（所发生的实时事件）的轻重缓急排队，优先处理最紧急事件的中断请求，于是规定每一个中断源都有自己的中断优先级。

MCS-51 单片机最典型的有 5 个中断源（外部中断 0、1，内部定时器中断 0、1，串口中断），具有两个中断优先级。两个外部中断：（P3.2 — INT0、P3.3 — INT1）上输入的外部中断源，低电平或负跳变有效，置位 TCON 中的 IE0 和 IE1 中断请求标志位。

1. 中断允许寄存器 IE：

D7	D6	D5	D4	D3	D2	D1	D0
EA	—	—	ES	ET1	EX1	ET0	EX0

EA：CPU 的中断开放标志。EA = 1，CPU 开放中断；EA = 0，CPU 屏蔽所有的中断申请。

EX0：外部中断 0 中断允许位。EX0 = 1，允许中断；EX0 = 0，禁止中断。

ET0：T0 的溢出中断允许位。ET0 = 1，允许 T0 中断；ET0 = 0，禁止 T0 中断。

EX1：外部中断 1 中断允许位。EX1 = 1，允许外部中断 1 中断；EX1 = 0，禁止外部中断 1 中断。

ET1：定时器 / 计数器 T1 的溢出中断允许位。ET1 = 1，允许 T1 中断；ET1 = 0 禁止 T1 中断。

ES： 串行口中断允许位。ES = 1，允许串行口中断；ES = 0 禁止串行口中断。

2. 中断控制寄存器 TCON 的各位（可位寻址）：

D7	D6	D5	D4	D3	D2	D1	D0
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

IE0：外部中断 0 请求源（INT0）标志。

IE0=1，外部中断 0 正在向 CPU 请求中断，当 CPU 响应该中断时由硬件清零 IE0（边沿触发方式）。

IT0：外部中断源 INT0 触发方式控制位。IT0 = 0，外部中断 0 程控为电平触发方式，当 P3.2 输入低电平时，置位 IE0。IT0 = 1，外部中断 0 程控为边沿触发方式，相继的两次采样，一个周期中采样到为高电平，接着的下一个周期中采样到为低电平，则置 '1' IE0。

IE1: 外部中断 1 请求 (INT1, P3.3) 标志。IE1 = 1 外部中断 1 向 CPU 请求中断, 当 CPU 响应外部中断时, 由硬件清 '0' IE1 (边沿触发方式)。

IT1: 外部中断 1 触发方式控制位。IT1 = 0, 外部中断 1 程控为电平触发方式, IT1 = 1, 外部中断 1 为边沿触发方式。其功能和 IT0 类似。

TR0: 定时 / 计数器 T0 运行控制位。(启动/停止)

TF0: 定时 / 计数器 T0 溢出中断标志位, CPU 执行中断服务程序时由硬件复位。

TR1: 定时 / 计数器 T1 运行控制位。

TF1: 定时 / 计数器 T1 溢出中断标志位, CPU 执行中断服务程序时由硬件复位。

3. 中断优先级寄存器 IP

D7	D6	D5	D4	D3	D2	D1	D0
—	—	—	PS	PT1	PX1	PT0	PX0

PS: 串行口中断优先级控制位。PS = 1, 串行口中断定义为高优先级中断; PS = 0, 行口中断定义为低优先级中断。

在 CPU 接收到同样优先级的几个中断请求源时, 一个内部的硬件查询序列确定优先服务于哪一个中断申请, 这样一个优先级里, 由查询序列确定了优先级结构, 其优先级排列的顺序依次为: 外部中断 0, 定时器 T0, 外部中断 1, 定时器 T1, 串行口中断。

使用 MCS-51 的中断, 要使用到的中断源编写中断服务程序。C51 为中断服务程序的编写提供了方便的方法。C51 的中断服务程序是一种特殊的函数, 它的说明形式为:

```
void 函数名(void) interrupt n using m
{ 函数体语句 }
```

这里, interrupt 和 using 是为编写 C51 中断服务程序而引入的关键字, interrupt 表示该函数是一个中断服务函数, interrupt 后的整数 n 表示该中断服务函数是对应哪一个中断源。

每个中断源都有系统指定的中断编号:

中 断 源	中 断 编 号
外部中断 0	0
定时器 T0	1
外部中断 1	2
定时器 T1	3
串行口中断	4

using 指定该中断服务程序要使用的工作寄存器组号, m 为 0~3。优先级为先 0 后 1, 先外后内, 如上所排顺序。

实验结果及分析:

设计思路: 在单片机的 P0 端口接一个共阴极数码管, 将按键接到 INT0 端口, 利用按键产生中断请求信号, 每按下一次就产生一次中断, 使得数字加一, 循环显示 0~9。

操作过程:

(1) 打开 proteus 新建空白文件, 点击 P 进入零件拾取界面, 输入 AT89C51, 点击 OK, 用鼠标拖动到合适的位置。7seg 共阴极数码管, 电阻包 RESPACK-7, button 按钮, 点击右键 Place, Terminal 选取电源 POWER 接地 GROUND, 进行连线, 保存。如图 1

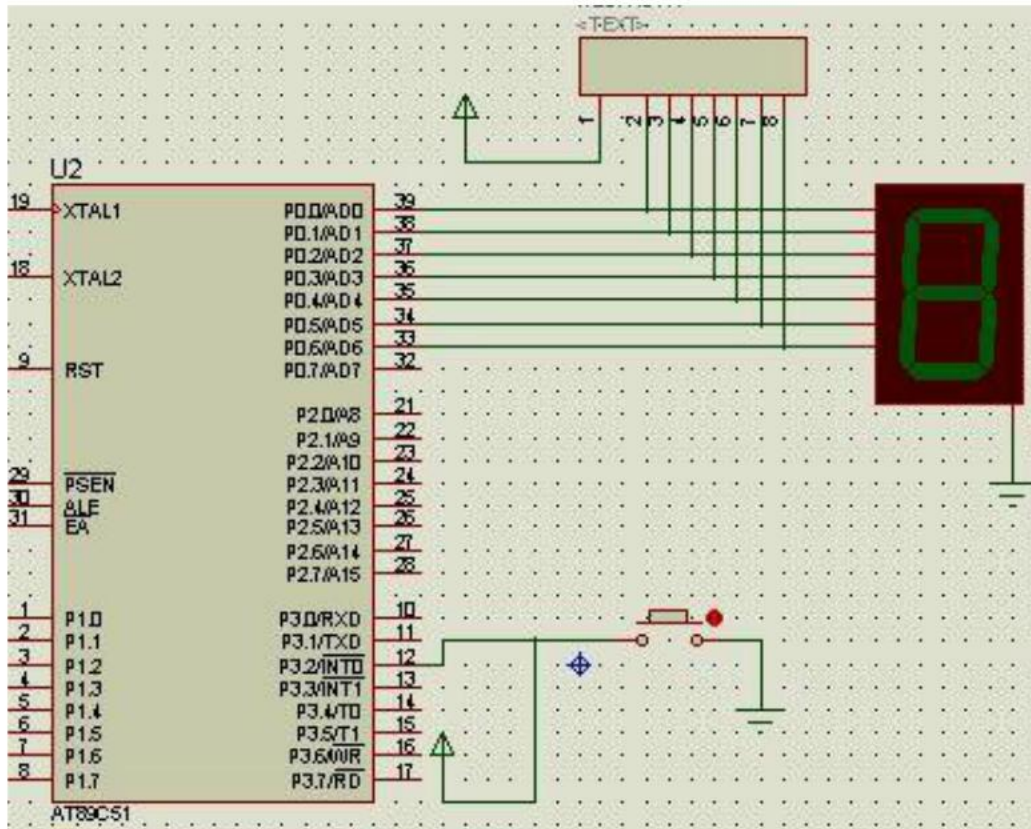


图 1

- (2) 开启 keil 并在 project 中选择 New Project, 新建项目, 然后在 File 中选择 New...
- (3) 保存为 shiyan3, 选择 Atmel, 选择 AT89C51, 点击确定, 弹出一个对话框选择“否”。
- (4) 在新建的空页中编写程序 shiyan3.c 如下, 并保存。

```
#include "reg51.h"
unsigned int i=0;
unsigned char code
tab[]={0x3f,0x06,0x5b,0x4f,0x66,0x6d,0x7d,0x07,0x7f,0x6f};
void int0() interrupt 0
{ i++;
if(i==10)
i=0;
P0=tab[i];
}
main()
{P0=tab[0];
EA=1;EX0=1;IT0=1;
while(1);}

```

- (6) 点击 如图操作, 选择生成的 .hex 文件, 点击 运行, 如果显示无错误就编译成功
- (7) 双击 AT89C51 点击 选择 shiyan5.hex 点击 OK 点击左下角的 运行。

中断显示结果如图 2



图 2

实验心得:

掌握了中断的设计, 会利用中断进行简单的加减设计。同理还可以扩展设计成多位数码管的加减运算, 并能得到广泛的应用。