

# 操作考试

读取 16 位开关低 4 位的值，将其对应的 16 进制数显示在 8 位数码管上

只有按下 BTNC 按键，才读取开关值并显示在第 1 位数码管上

循环右移显示：按下 BTNR 按键，右移一位显示，到达第 8 位后，再按下 BTNR，回到第 1 位

要有按键消抖

要求中断方式实现

```
#include "xil_io.h"
#include "stdio.h"
#include "xintc_l.h"
#include "xgpio_l.h"

void btnHandler(void) __attribute__((interrupt_handler));

u8 seg_code[17] = {0xc0, 0xf9, 0xa4, 0xb0, 0x99, 0x92, 0x82, 0xf8, 0x80, 0x90,
                  0x88, 0x83, 0xc6, 0xa1, 0x86, 0x8e, 0xff};
u8 pos_code[8] = {0x7f, 0xbf, 0xdf, 0xef, 0xf7, 0xfb, 0xfd, 0xfe};

int pos = 0;
u8 btn_status;
u32 sw_status;

int main()
{
    // GPIO 输入/输出配置
    Xil_Out16(XPAR_AXI_GPIO_2_BASEADDR + XGPIO_TRI_OFFSET, 0x1f); // 读取按钮状态
    Xil_Out16(XPAR_AXI_GPIO_0_BASEADDR + XGPIO_TRI_OFFSET, 0xffff); // 读取开关状态
    Xil_Out16(XPAR_AXI_GPIO_1_BASEADDR + XGPIO_TRI_OFFSET, 0x0); // 输出数码管位选信号
    Xil_Out16(XPAR_AXI_GPIO_1_BASEADDR + XGPIO_TRI2_OFFSET, 0x0); // 输出数码管段选信号

    // GPIO 中断配置
    Xil_Out32(XPAR_AXI_GPIO_2_BASEADDR + XGPIO_ISR_OFFSET, 0x1); // 清除GPIO_2的通道1（按钮）的中断状态
    Xil_Out32(XPAR_AXI_GPIO_2_BASEADDR + XGPIO_IER_OFFSET, 0x1); // 使能通道1中断
    Xil_Out32(XPAR_AXI_GPIO_2_BASEADDR + XGPIO_GIE_OFFSET, 0x80000000); // 使能中断输出

    // INTC初始化
    Xil_Out32(XPAR_AXI_INTC_0_BASEADDR + XIN_IAR_OFFSET, 0xffffffff); // 清除所有通道的中断状态
```

```

Xil_Out32(XPAR_AXI_INTC_0_BASEADDR + XIN_IER_OFFSET,
          XPAR_AXI_GPIO_2_IP2INTC_IRPT_MASK ); // 使能按钮中断
Xil_Out32(XPAR_AXI_INTC_0_BASEADDR + XIN_MER_OFFSET, 0x3); // 使能硬件中断irq
输出

microblaze_enable_interrupts();
return 0;
}

void btnHandler()
{
    btn_status = Xil_In8(XPAR_AXI_GPIO_2_BASEADDR + XGPIO_DATA_OFFSET) & 0x1f;

    if (btn_status == 0x1) // 读取开关状态并显示在最高位
    {
        pos = 0;
        sw_status = Xil_In8(XPAR_AXI_GPIO_0_BASEADDR + XGPIO_DATA_OFFSET) & 0xf;
        Xil_Out32(XPAR_GPIO_1_BASEADDR + XGPIO_DATA_OFFSET, pos_code[pos]);
        Xil_Out32(XPAR_GPIO_1_BASEADDR + XGPIO_DATA2_OFFSET,
seg_code[sw_status]);
    }

    if (btn_status == 0x8) // 右移一位显示
    {
        pos++;
        if (pos == 8)
        {
            pos = 0;
        }
        while ((Xil_In8(XPAR_AXI_GPIO_2_BASEADDR + XGPIO_DATA_OFFSET) & 0x1f) !=
0); // 等待按键松开
        Xil_Out32(XPAR_GPIO_1_BASEADDR + XGPIO_DATA_OFFSET, pos_code[pos]);
        Xil_Out32(XPAR_GPIO_1_BASEADDR + XGPIO_DATA2_OFFSET,
seg_code[sw_status]);
    }

    Xil_Out32(XPAR_AXI_GPIO_2_BASEADDR + XGPIO_ISR_OFFSET,
              Xil_In32(XPAR_AXI_GPIO_2_BASEADDR + XGPIO_ISR_OFFSET));
}

```