并行IO接口实验

电磁1702 桂正泰

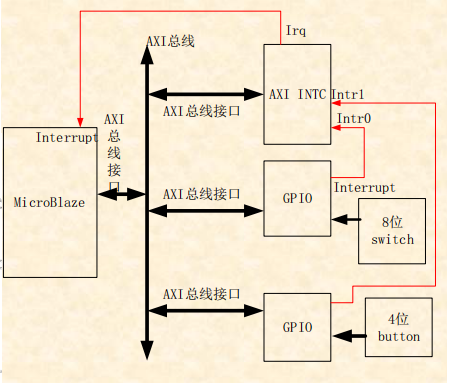
U201713557

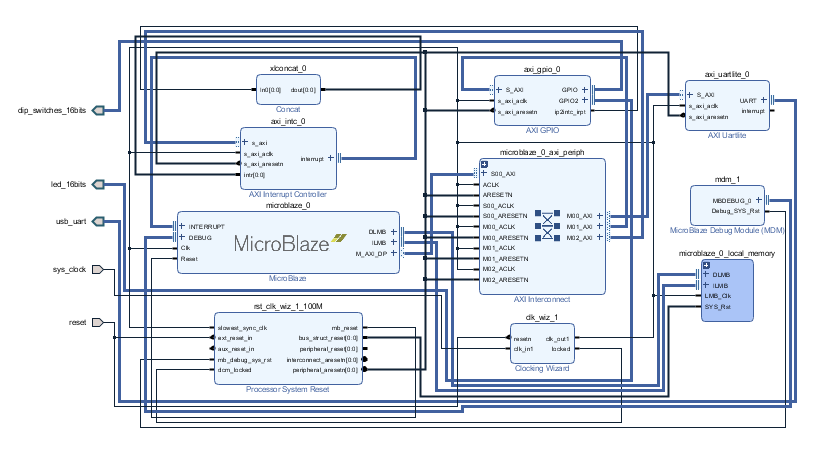
1、实验任务

独立式开关输入，显示开关表示的二进制值到console和LED

·查询、中断

1. 硬件电路框图





1. 硬件实现步骤

·利用Vivado创建最小系统工程

·包含处理器、内存、UART、debug、clk、reset等硬件模块的添加

·添加需要的IP核：GPIO、INTC

·总线控制器、自动连线

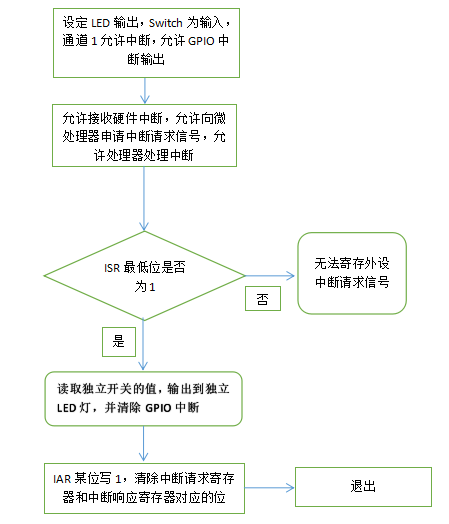
·建立模块连接关系及内存映射

·验证硬件设计

1. 实现实验任务要求

先利用Vivado建立最小系统工程，封装，生成bit流文件，进入SDk软件，分别写入查询、中断的代码，并待一切完毕后，进行接口连接，用开关输入操作LED灯亮灭。

1. 软件流程图



1. 软件源代码加注释

查询方式代码

#includ"stdio.h"  
#include"xil\_io.h"  
int main()  
{  
 unsigned short led;  
 led=0; //初始化输入16进制数  
 unsigned short last\_sw,current\_sw;  
 Xil\_Out16(0x40000004,0xffff); //配置通道低16位输入  
 Xil\_Out16(0x4000000c,0x0); //配置通道低16位输出  
 while(1)  
 {  
 last\_sw=current\_sw; //保存前一开关状态  
 current\_sw=Xil\_In16(0x40000000)&0xffff; //读取当前开关值  
 if(last\_sw!=current\_sw) //判断前后状态是否一致  
 {  
 xil\_printf("The switches'code is 0x%4x\n",current\_sw);  
 led=current\_sw; //LED灯变成当前状态  
  
 }  
 Xil\_Out16(0x40000008,led); //输出16进制数到LED灯  
 }  
}

中断方式代码

/\*

\* gzt.c

\*

\* Created on: 2019年5月21日

\* Author: Administrator

\*/

#include "xil\_io.h"

#include "stdio.h"

#define INTC\_BASEADDR 0x41200000

#define XIN\_ISR\_OFFSET 0

#define XIN\_IER\_OFFSET 8

#define XIN\_IAR\_OFFSET 12

#define XIN\_MER\_OFFSET 28

#define Switch\_LED\_BASEADDR 0x40000000

#define XGPIO\_DATA1\_OFFSET 0x0

#define XGPIO\_TRI1\_OFFSET 0x4

#define XGPIO\_DATA2\_OFFSET 0x8

#define XGPIO\_TRI2\_OFFSET 0xc

#define XGPIO\_IER\_OFFSET 0x128

#define XGPIO\_ISR\_OFFSET 0x120

#define XGPIO\_GIE\_OFFSET 0x11c

void SwitchHandler();

void My\_ISR()\_\_attribute\_\_((interrupt\_handler)); //总中断服务程序

int main()

{

Xil\_Out32(Switch\_LED\_BASEADDR+XGPIO\_TRI2\_OFFSET,0x0); //设定LED为输出方式

Xil\_Out32(Switch\_LED\_BASEADDR+XGPIO\_TRI1\_OFFSET,0xffff); //设定Switch为输入方式

Xil\_Out32(Switch\_LED\_BASEADDR+XGPIO\_IER\_OFFSET,0x1); //通道1允许中断

Xil\_Out32(Switch\_LED\_BASEADDR+XGPIO\_GIE\_OFFSET,0x80000000); //允许GPIO中断输出

Xil\_Out32(INTC\_BASEADDR+XIN\_IER\_OFFSET,0x1); //中断控制器使能中断源INTR0

Xil\_Out32(INTC\_BASEADDR+XIN\_MER\_OFFSET,0x3);

microblaze\_enable\_interrupts(); //允许处理器处理中断

return 0;

}

void My\_ISR()

{

int status;

status=Xil\_In32(INTC\_BASEADDR+XIN\_ISR\_OFFSET); //读取ISR

if((status&0x1)==0x1)

SwitchHandler(); //调用用户中断服务程序

Xil\_Out32(INTC\_BASEADDR+XIN\_IAR\_OFFSET,status); //写IAR

}

void SwitchHandler()

{

Xil\_Out32(Switch\_LED\_BASEADDR+XGPIO\_DATA2\_OFFSET,Xil\_In32(Switch\_LED\_BASEADDR+XGPIO\_DATA1\_OFFSET));

//读取独立开关的值，输出到独立LED灯

Xil\_Out32(Switch\_LED\_BASEADDR+XGPIO\_ISR\_OFFSET,Xil\_In32(Switch\_LED\_BASEADDR+XGPIO\_ISR\_OFFSET));

//清除GPIO中断

}

1. 心得体会

在这个实验当中，主要是建立一个独立式开关输入，显示开关表示的二进制值到console和LED的硬件，分别用查询方式，中断方式来实现。在整个实验，主要分为硬件部分，和软件部分的操作。硬件上的操作设计，因为操作了数遍，所以对于我来说，不是特别难，而进入软件模块后，代码也不是很难打，只是比较难理解代码中许多寄存器相对应的含义，没真正弄懂例如ISR，IAR，MER哪些位对应的如屏蔽，控制中断请求的等等。而通过本次实验，也对接口的利用，认识上有了更多的了解。