实验报告

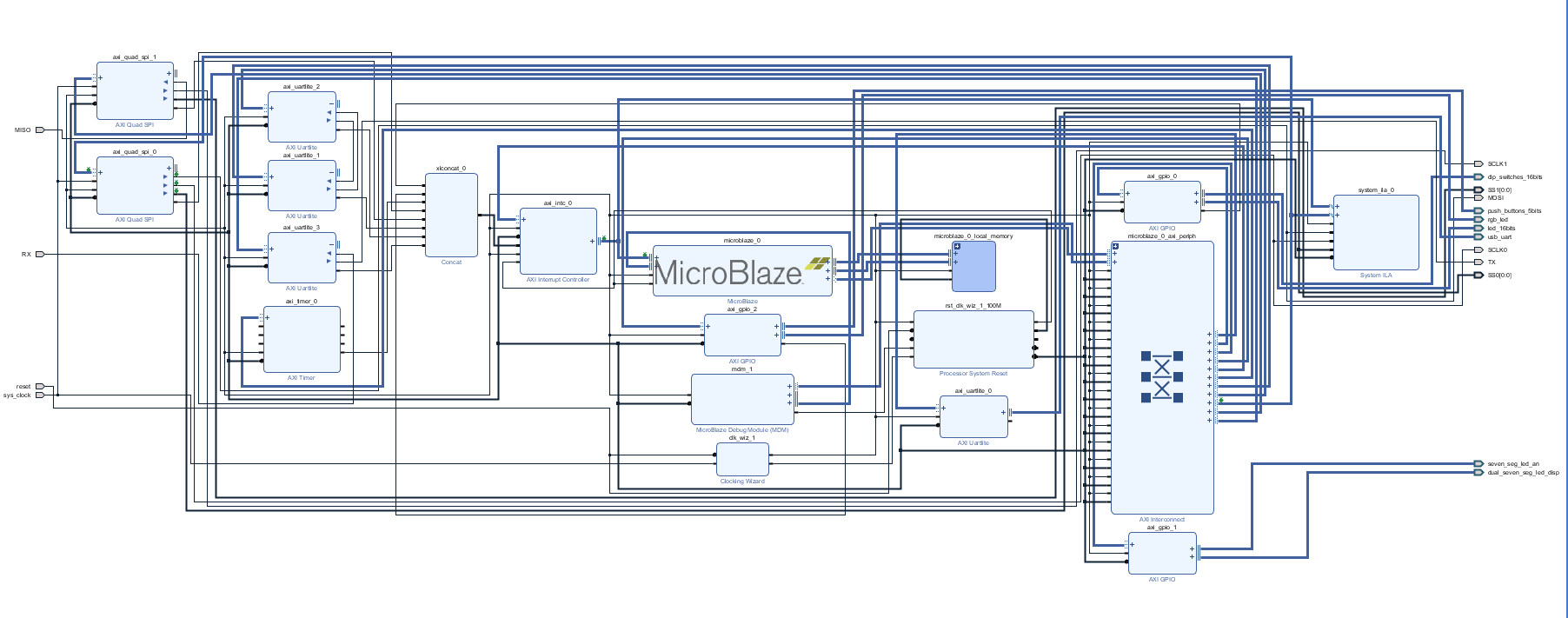
通信1801 李何涛

U201813403

# 实验任务

采用UART IP 核，实现Nexys4 或Nexys4 DDR 实验板UART接口之间的通信。要求当拨动开关时，将开关对应的值通过UART1发送到UART2，同时利用LED 灯指示UART2接收到的当前开关的值；当按下按键时，将按键对应的值通过UART2发送到UART1，同时利用数码管指示UART1接收到的当前按下的按键位置码（C,U,d,L,r)。UART 波特率为9600bps。

# 硬件电路框图



# 程序源代码

#include "xil\_io.h"

#include "stdio.h"

#include "xintc\_l.h"

#include "xgpio\_l.h"

#include "xuartlite\_l.h"

void UART\_SR();

void BtnHandler();

void My\_ISR() \_\_attribute\_\_ ((interrupt\_handler));

int main()

{

Xil\_Out32(XPAR\_AXI\_GPIO\_2\_BASEADDR+XGPIO\_TRI\_OFFSET,0x1f); //设定BUTTON为输入方式

Xil\_Out16(XPAR\_GPIO\_0\_BASEADDR+XGPIO\_TRI2\_OFFSET,0x0); //LED灯输出

Xil\_Out32(XPAR\_AXI\_GPIO\_2\_BASEADDR+XGPIO\_IER\_OFFSET,XGPIO\_IR\_CH1\_MASK); //通道1允许中断

Xil\_Out32(XPAR\_AXI\_GPIO\_2\_BASEADDR+XGPIO\_GIE\_OFFSET,XGPIO\_GIE\_GINTR\_ENABLE\_MASK); //允许GPIO中断输出

Xil\_Out32(XPAR\_AXI\_UARTLITE\_3\_BASEADDR+XUL\_CONTROL\_REG\_OFFSET,XUL\_CR\_ENABLE\_INTR|XUL\_CR\_FIFO\_RX\_RESET|XUL\_CR\_FIFO\_TX\_RESET);//使能中断，清除RX，TX寄存器

Xil\_Out32(XPAR\_AXI\_INTC\_0\_BASEADDR+XIN\_IER\_OFFSET,XPAR\_AXI\_GPIO\_2\_IP2INTC\_IRPT\_MASK|XPAR\_AXI\_UARTLITE\_3\_INTERRUPT\_MASK);//对中断控制器进行中断源使能

Xil\_Out32(XPAR\_AXI\_INTC\_0\_BASEADDR+XIN\_MER\_OFFSET,XIN\_INT\_MASTER\_ENABLE\_MASK|XIN\_INT\_HARDWARE\_ENABLE\_MASK);

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

return 0;

}

void My\_ISR()

{

int status;

status=Xil\_In32(XPAR\_AXI\_INTC\_0\_BASEADDR+XIN\_ISR\_OFFSET);//读取ISR

xil\_printf("0x%x\n",status);

if((status&XPAR\_AXI\_UARTLITE\_3\_INTERRUPT\_MASK)==XPAR\_AXI\_UARTLITE\_3\_INTERRUPT\_MASK)

{

UART\_SR();

}

else if((status&XPAR\_AXI\_GPIO\_2\_IP2INTC\_IRPT\_MASK)==XPAR\_AXI\_GPIO\_2\_IP2INTC\_IRPT\_MASK)

BtnHandler(); //调用按键中断

Xil\_Out32(XPAR\_AXI\_INTC\_0\_BASEADDR+XIN\_IAR\_OFFSET,status);//写IAR

}

void UART\_SR()

{

Xil\_Out16(XPAR\_GPIO\_0\_BASEADDR+XGPIO\_DATA2\_OFFSET,Xil\_In32(XPAR\_AXI\_UARTLITE\_3\_BASEADDR+XUL\_RX\_FIFO\_OFFSET));

}

void BtnHandler()

{

unsigned short btncode;

btncode=Xil\_In8(XPAR\_AXI\_GPIO\_2\_BASEADDR+XGPIO\_DATA\_OFFSET)&0x1f;

while((Xil\_In32(XPAR\_AXI\_GPIO\_2\_BASEADDR+XGPIO\_DATA\_OFFSET)&0x1f)!=0)

{

xil\_printf("The pushed button's code is 0x%x\n",btncode);//打印输入的按键序号

}

switch(btncode)

{

case 0x1:

Xil\_Out32(XPAR\_AXI\_UARTLITE\_3\_BASEADDR+XUL\_TX\_FIFO\_OFFSET,btncode);

break;

case 0x2:

Xil\_Out32(XPAR\_AXI\_UARTLITE\_3\_BASEADDR+XUL\_TX\_FIFO\_OFFSET,btncode);

break;

case 0x4:

Xil\_Out32(XPAR\_AXI\_UARTLITE\_3\_BASEADDR+XUL\_TX\_FIFO\_OFFSET,btncode);

break;

case 0x8:

Xil\_Out32(XPAR\_AXI\_UARTLITE\_3\_BASEADDR+XUL\_TX\_FIFO\_OFFSET,btncode);

break;

case 0x10:

Xil\_Out32(XPAR\_AXI\_UARTLITE\_3\_BASEADDR+XUL\_TX\_FIFO\_OFFSET,btncode);

break;

default:

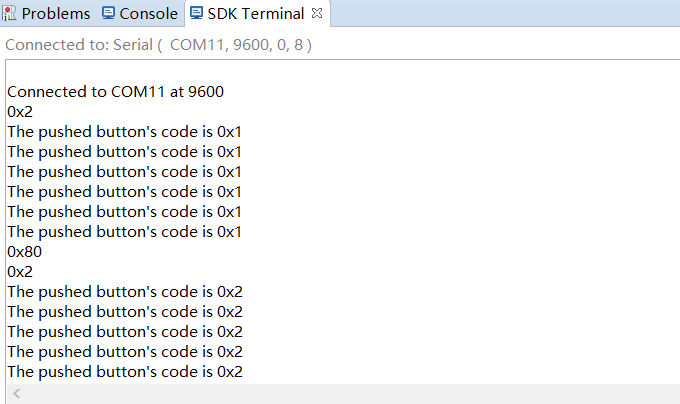
break;

}

Xil\_Out32(XPAR\_AXI\_GPIO\_2\_BASEADDR+XGPIO\_ISR\_OFFSET,Xil\_In32(XPAR\_AXI\_GPIO\_2\_BASEADDR+XGPIO\_ISR\_OFFSET)); //清除中断

}

# 实验结果



# 五．实验心得

在这个实验当中，主要是运用UART模块实现Nexys4 DDR 实验板UART接口之间的通信，在实验进程中，硬件平台搭设时，需要增加新的UART IP核作为标准输入输出接口，另外在管脚约束时，需要自己重新定义。在软件平台编写代码后，程序能执行但是led灯不亮，且断开自发自收的连接线后，需要重新复位程序才能继续执行，在检查了下程序之后，发现可以去掉检验标准输入输出接口是否中断的while循环后程序正确执行。总的来说，此次实验让我对UART通信有了更加深刻的认识。