**华中科技大学**

**实验报告**

**课程名称：微机原理与接口技术实验**

**实验项目名称：程序控制并行IO接口**

**指导老师：罗杰**

**专业/班级：电磁1802**

**学生学号：U201813405**

**学生姓名：吴叶赛**

**成绩/等级（批阅教师）：**

**实验执行日期： 2020年9月12日**

**1.实验任务**

嵌入式计算机系统将独立按键以及独立开关作为输入设备，LED灯作为输出设备。修改实验示例程序代码，实现以下功能：  
（1）按下BTNC按键时，计算机读入一组16位独立开关状态作为第一个输入的二进制数据，并即时显示输入的二进制数到16位LED灯上。（没有按下BTNC按键时，开关拨动不读入数据）；  
（2）按下BTNR按键时，计算机读入另一组16位独立开关状态作为第二个输入的二进制数据，并即时显示输入的二进制数到16位LED灯上。（没有按下BTNR按键时，开关拨动不读入数据）；  
（3）按下BTNU按键时，将保存的2组二进制数据做无符号加法运算，并将运算结果输出到LED灯对应位；

（4）按下BTND按键时，将保存的2组二进制数据做无符号乘法运算，并将运算结果输出到LED灯对应位。  
►程序控制方式提示：循环读取按键键值，根据按键的值读取开关状态，并做相应处理。

**2.实验目的**

（1）掌握GPIO工作原理和使用方法；

（2）掌握使用GPIO设计常见并行外设接口；

（3）掌握程序控制方式IO接口C语言控制程序设计。

**3.实验环境**

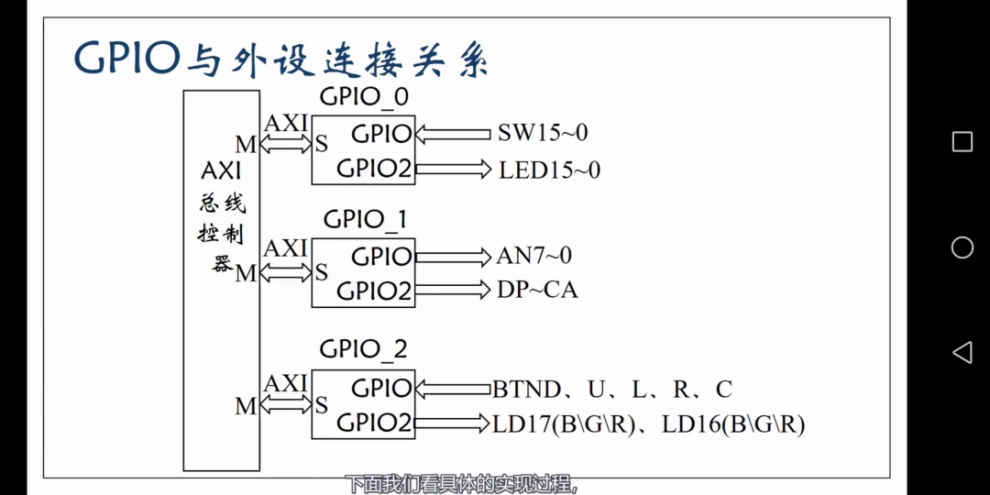
（1）Windows 7 以上操作系统；

（2）编辑工具：Vivado，SDK；

（3）实验开发板：Nexys 4 DDR。

**4.设计方案**

硬件框图：

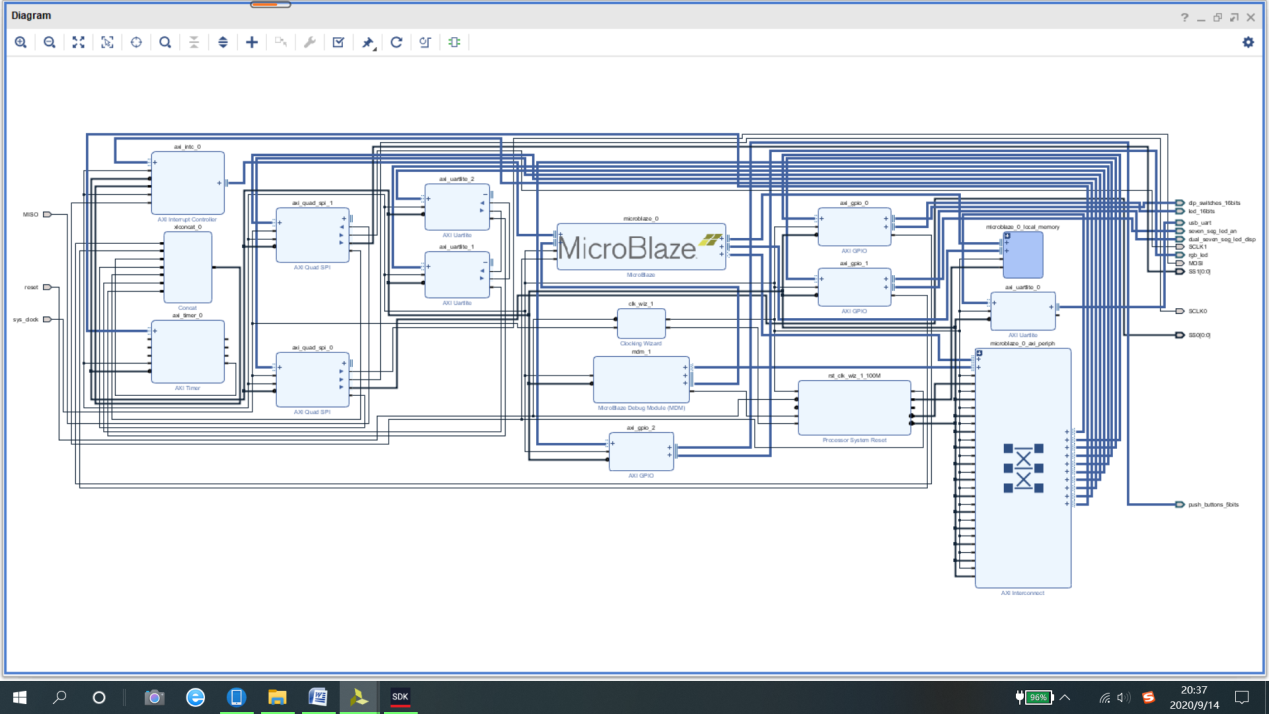


Nexys 4 DDR开发板的16个独立开关SW接GPIO\_0的第一个端口，16个LED发光二极管接GPIO\_0的第二个端口，数码显示管的位选信号AN7~0接GPIO\_1的第一个端口，数码显示管的片选信号DP~CA接GPIO\_1的第二个端口，5个独立按键BTND.BTNU.BTNL.BTNR.BTNC接GPIO\_2的第一个端口。

首先要不断循环读取按键健值，用while(1)包括整个程序，读入写入数据用Xilinx C语言提供的两类基本IO读写函数。设置两个变量last\_sw和current\_sw表示保存的第一组和第二组16位独立开关状态二进制数据，即可实现保存两组数据的加法和乘法运算，当按下BTNU或BTND按键时都先将current\_sw赋值给last\_sw,再读取一组新的开关数据作为即时current\_sw并输出到LED灯。根据不同按键的值读取开关状态作不同处理用if条件语句控制。

**5.实现过程**

（1）硬件平台：



（2）源代码及注释：

/\*

\* ming\_0.c

\*

\* Created on: 2020年9月13日

\* Author: 赛赛

\*/

#include"stdio.h"

#include"xil\_io.h"

#include"xgpio\_l.h"

#define XPAR\_AXI\_GPIO\_0\_BASEADDR 0x40000000

#define XPAR\_AXI\_GPIO\_2\_BASEADDR 0x40020000

int main()

{

unsigned short button,last\_sw,last\_sw1,current\_sw,current\_sw1,he,ji;

Xil\_Out16(XPAR\_AXI\_GPIO\_0\_BASEADDR+XGPIO\_TRI\_OFFSET,0xffff);//初始化

Xil\_Out8(XPAR\_AXI\_GPIO\_2\_BASEADDR+XGPIO\_TRI\_OFFSET,0x1f);

Xil\_Out16(XPAR\_AXI\_GPIO\_0\_BASEADDR+XGPIO\_TRI2\_OFFSET,0x0000);

while(1)

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

{

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

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

xil\_printf("The pushed button's code is 0x%x\n",button);//打印按键输入

if(button==0x1)

{

last\_sw=current\_sw;

current\_sw=Xil\_In16(XPAR\_AXI\_GPIO\_0\_BASEADDR+XGPIO\_DATA\_OFFSET)&0xffff;

if(last\_sw!=current\_sw)

xil\_printf("The switches code is 0x%4x\n",current\_sw);//打印开关输入

Xil\_Out16(XPAR\_AXI\_GPIO\_0\_BASEADDR+XGPIO\_DATA2\_OFFSET,current\_sw);

xil\_printf("The led code is 0x%4x\n",current\_sw);//打印led输出

}

else if(button==0x8)

{

last\_sw1=current\_sw1;

current\_sw1=Xil\_In16(XPAR\_AXI\_GPIO\_0\_BASEADDR+XGPIO\_DATA\_OFFSET)&0xffff;

if(last\_sw1!=current\_sw1)

xil\_printf("The switches code is 0x%4x\n",current\_sw1);//打印开关输入

Xil\_Out16(XPAR\_AXI\_GPIO\_0\_BASEADDR+XGPIO\_DATA2\_OFFSET,current\_sw1);

xil\_printf("The led code is 0x%4x\n",current\_sw1);//打印led输出

}

else if(button==0x2)

{

he=current\_sw+current\_sw1; Xil\_Out16(XPAR\_AXI\_GPIO\_0\_BASEADDR+XGPIO\_DATA2\_OFFSET,he);

xil\_printf("The led he code is 0x%4x\n",he);//打印led输出

}

else if(button==0x10)

{

ji=current\_sw\*current\_sw1;

Xil\_Out16(XPAR\_AXI\_GPIO\_0\_BASEADDR+XGPIO\_DATA2\_OFFSET,ji);

xil\_printf("The led ji code is 0x%4x\n",ji);//打印led输出

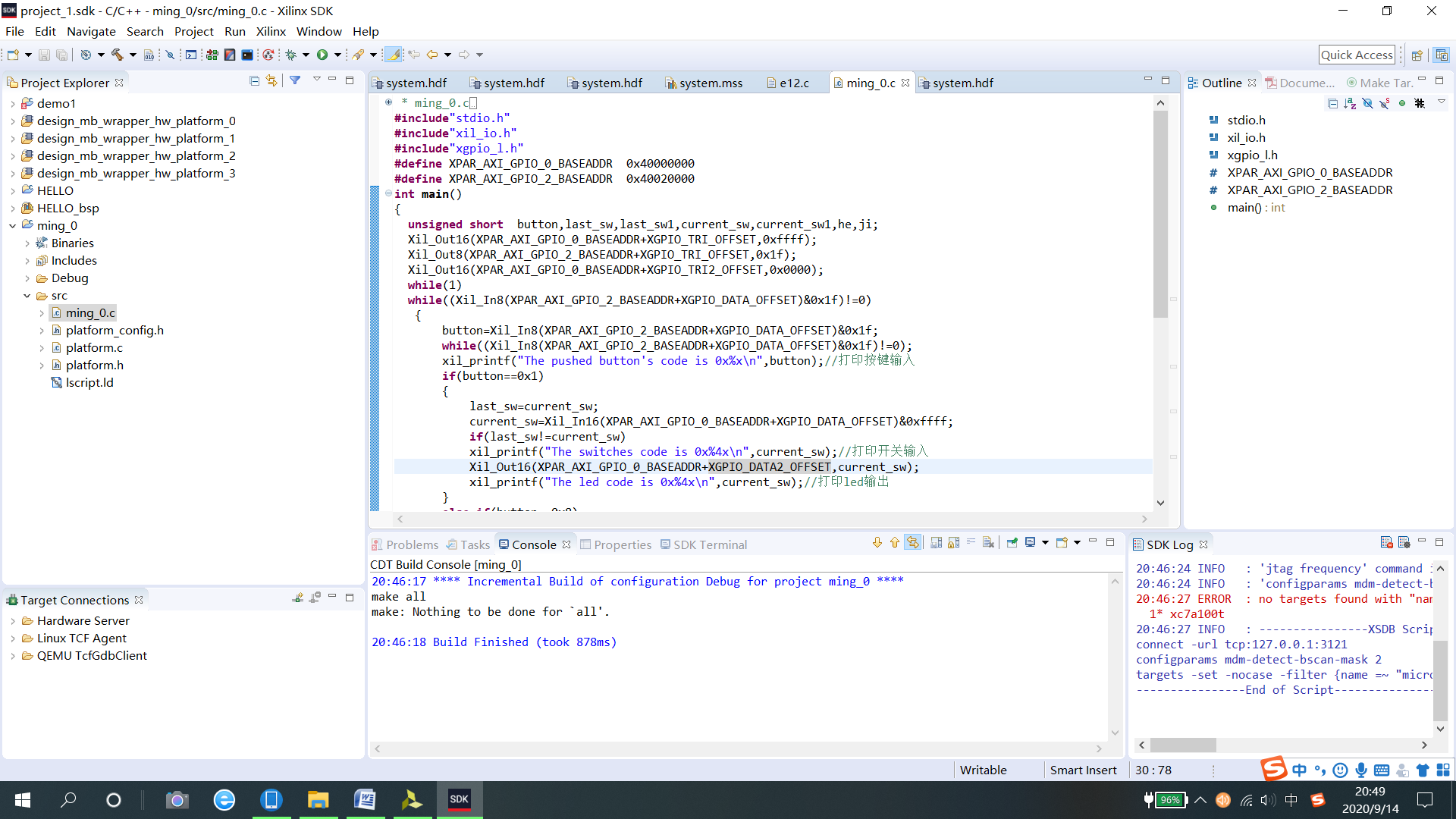
}

}

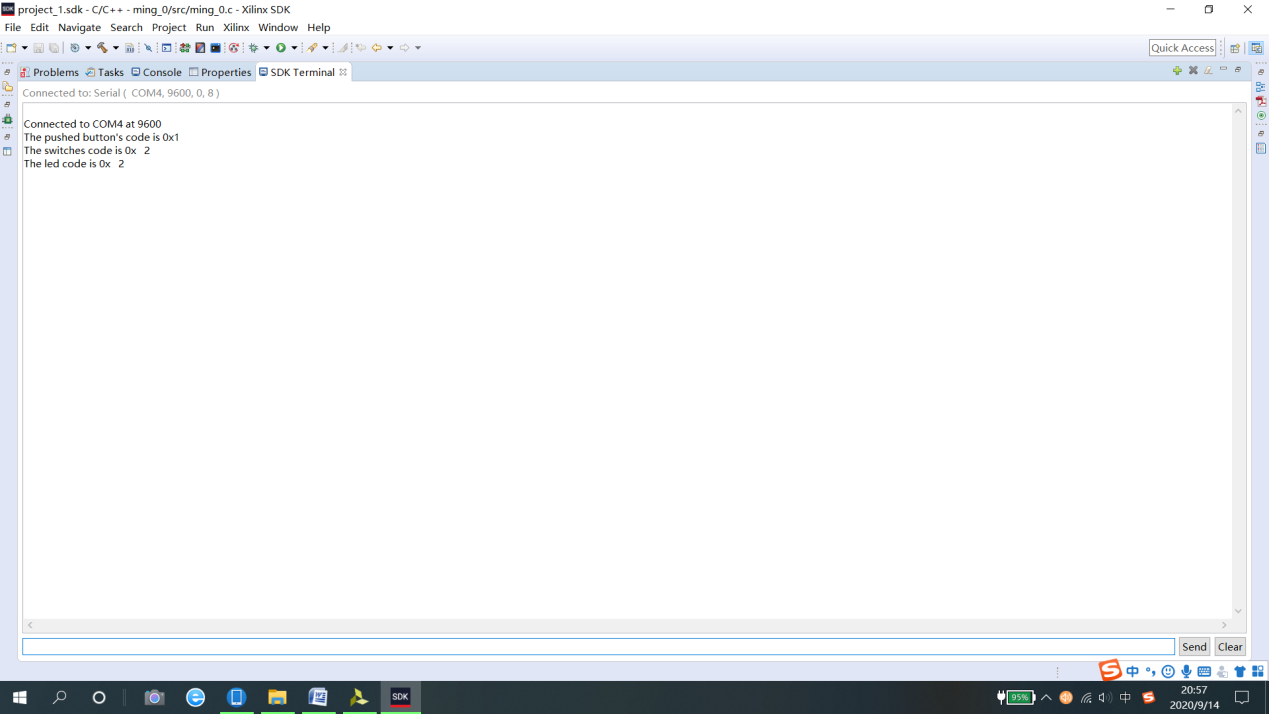
}

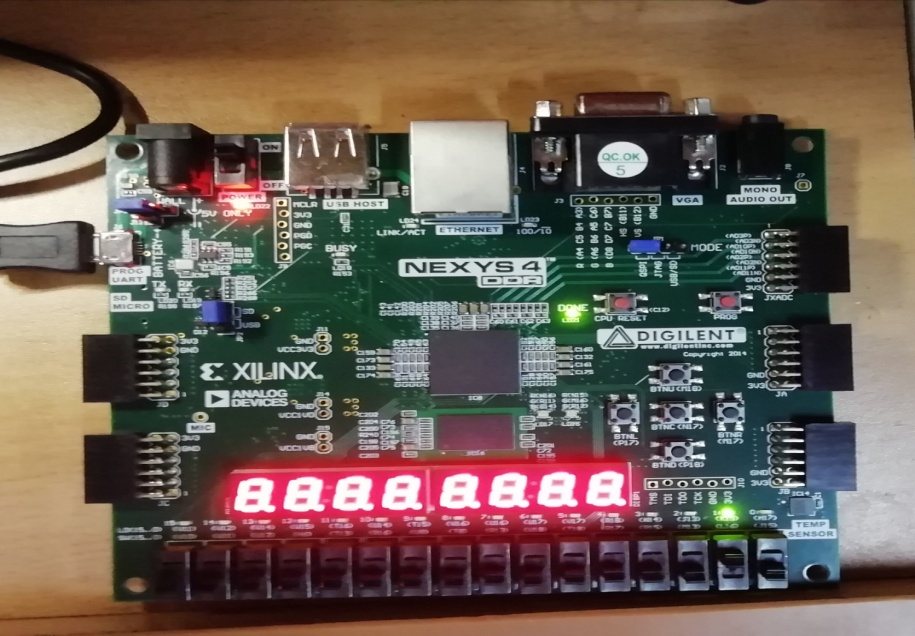
**6.实验结果**

（1）源代码运行debug没有产生错误：

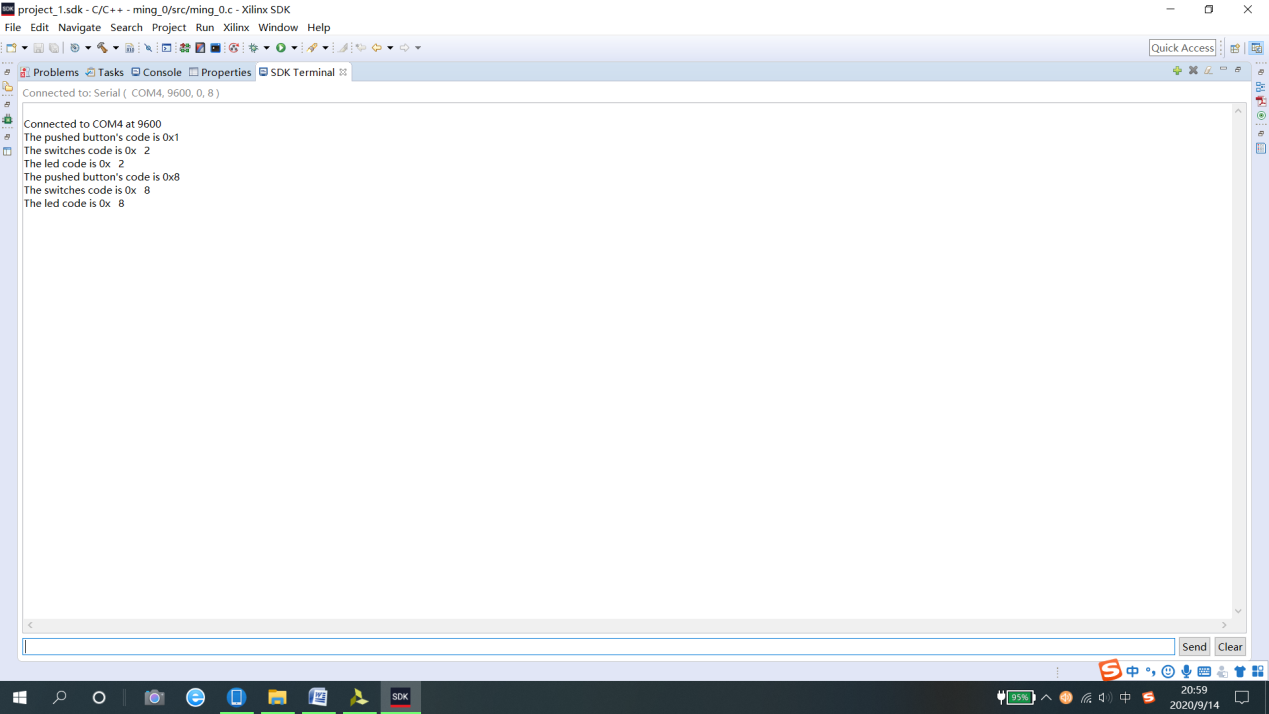


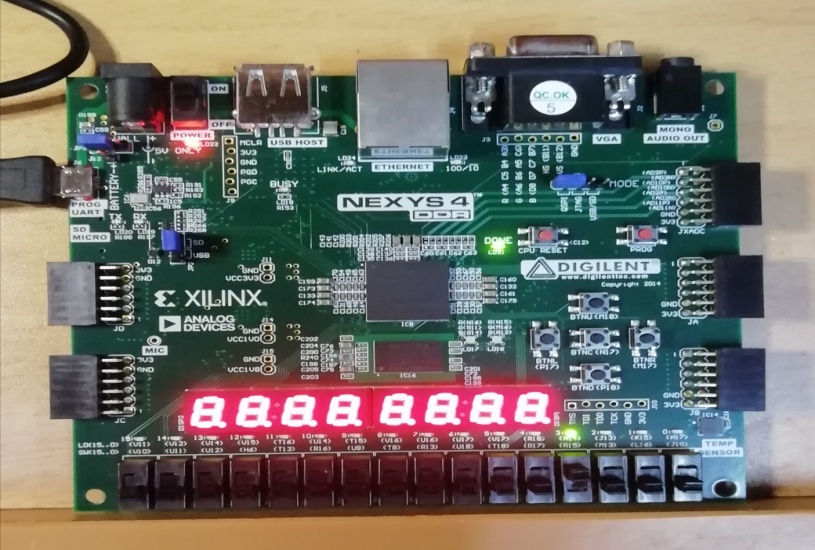
（2）将FPGA控制程序下载到开发板上，拨动第一组开关值为0x2，按下BTNC按键，对应led灯发光并打印出按键值0x1、开关值0x2和led值0x2：



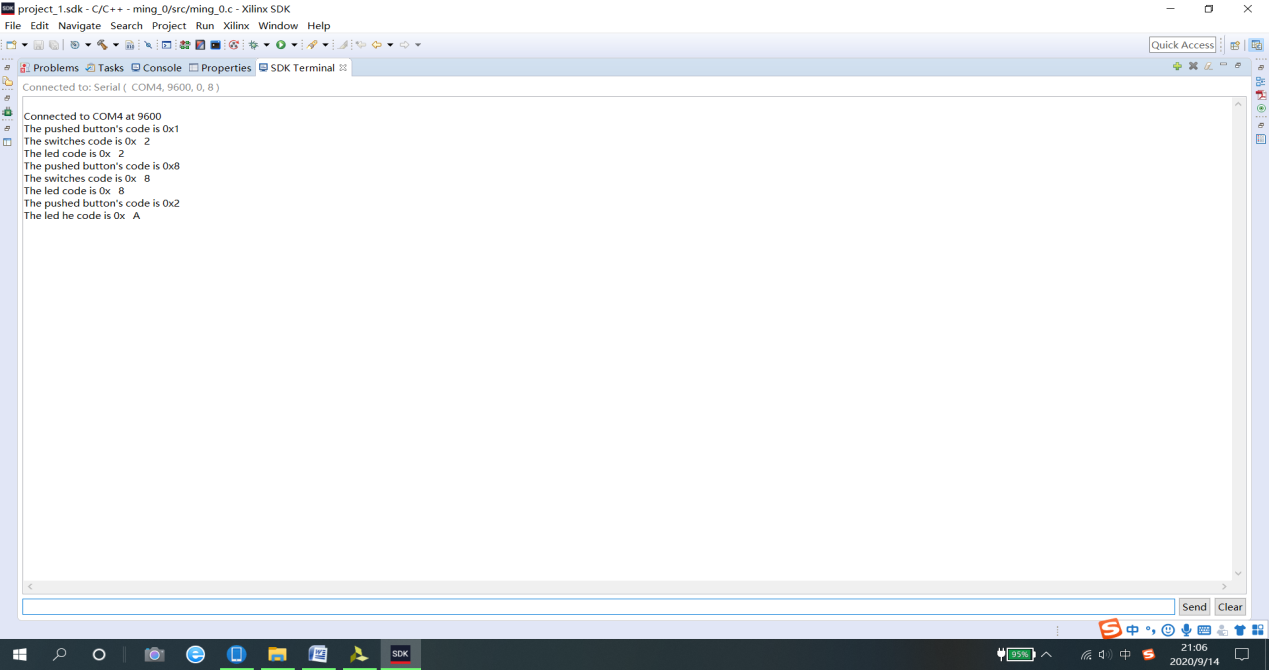


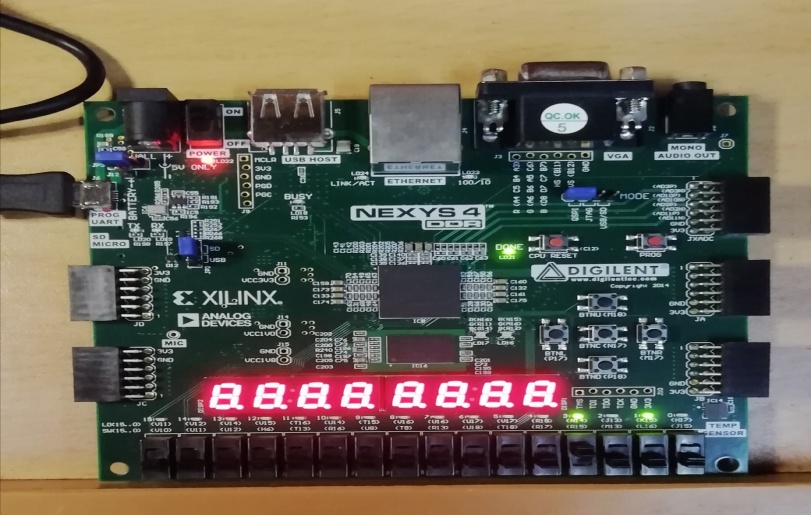
（3）再拨动第二组开关值为0x8, 按下BTNR按键，对应led灯发光并打印出按键值0x8、开关值0x8和led值0x8：



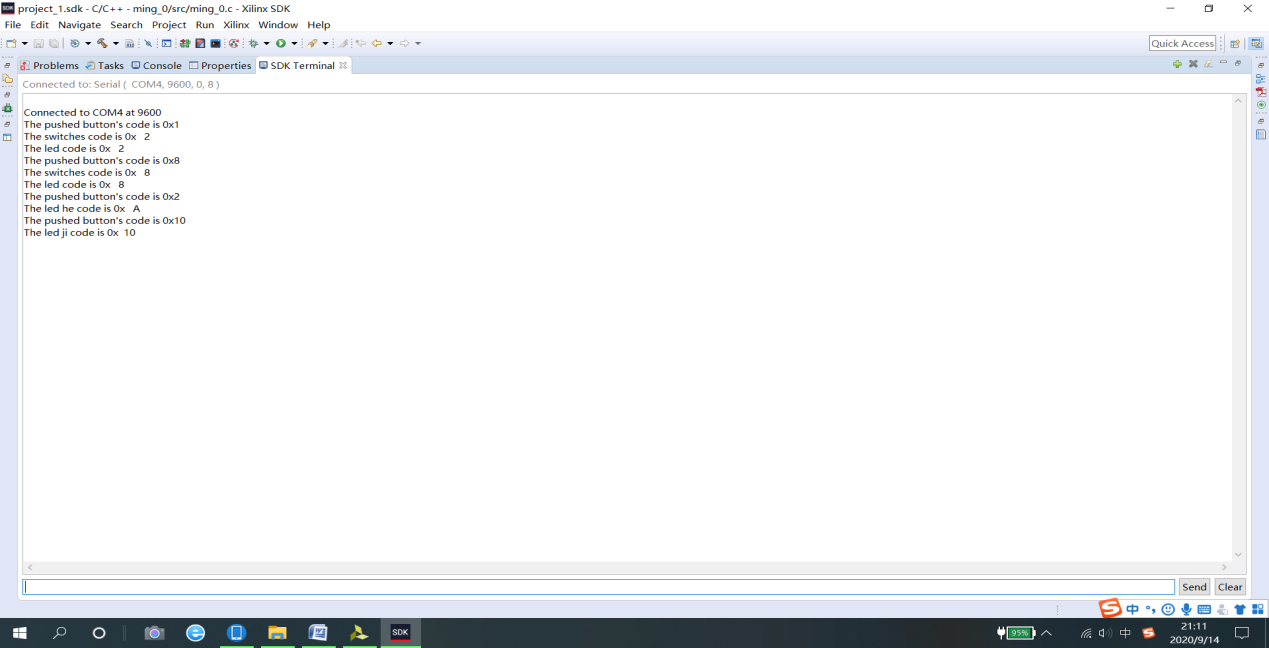


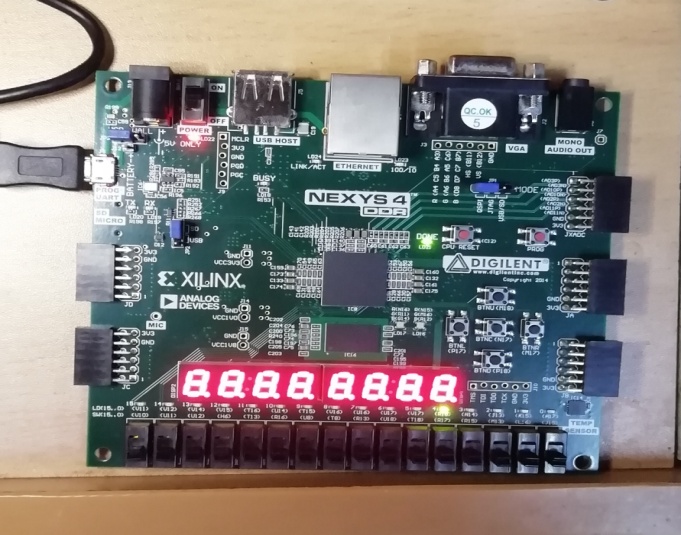
（4）按下BTNU按键，前两组二进制数据做无符号加法运算后对应的led灯发光并打印出按键值0x2、和led值0x2+0x8=0xa：



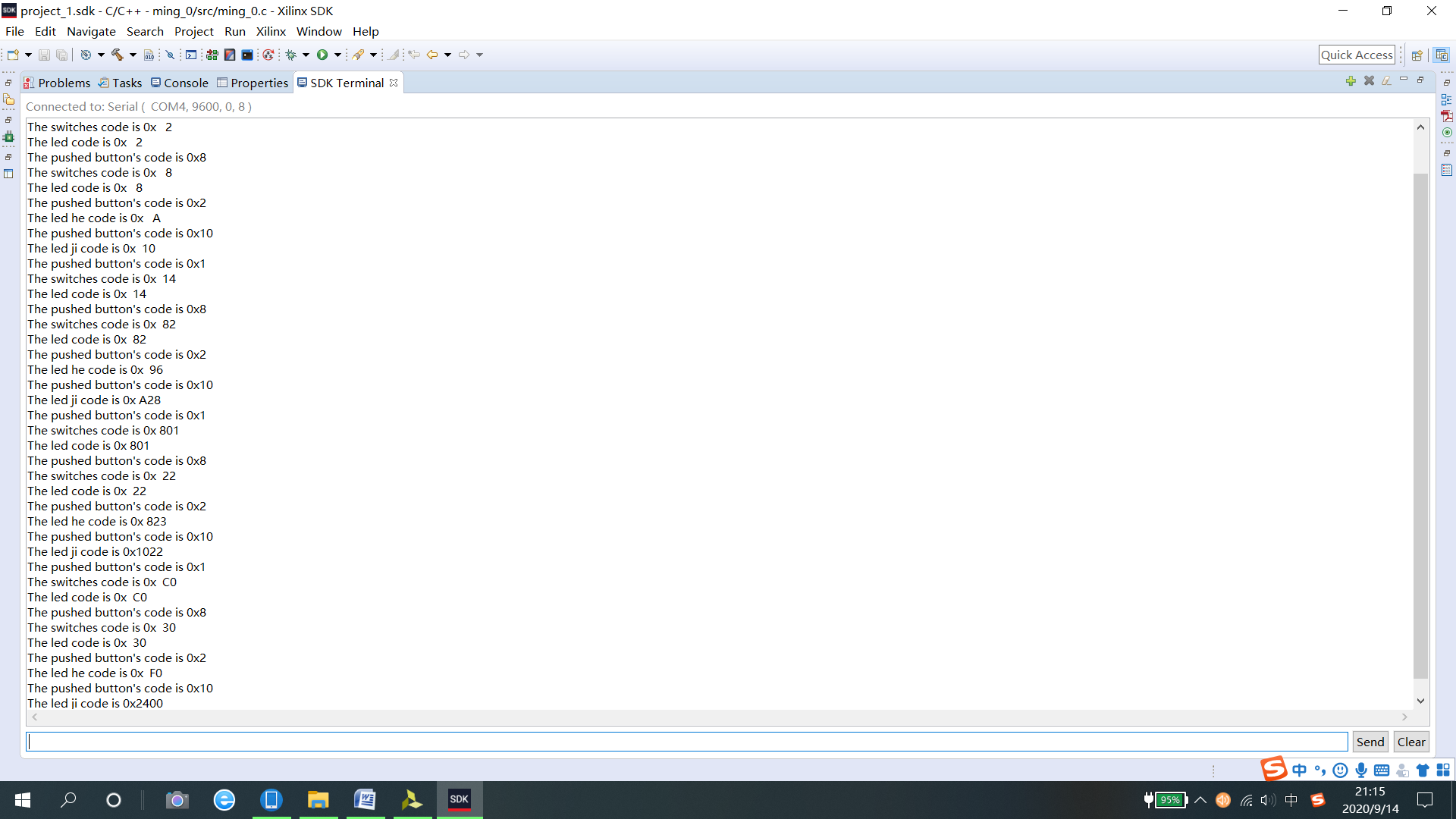


（5）按下BTND按键，前两组二进制数据做无符号乘法运算后对应的led灯发光并打印出按键值0x10、和led值0x2\*0x8=0x10：





（6）重复上述步骤（2）~（5），打印信息如下图所示，均满足实验任务要求：



**七、实验总结**

（1）因为0x2+0x8=0xa, 0x2\*0x8=0x10, 其对应的led灯分别正常发光，SDK打印的数值信息和顺序也正确，所以达到实验目标，成功完成实验任务。

（2）心得体会：用Vivado搭建硬件平台时一定要仔细按照视频讲解一步一步慢慢来，少一步或者看错一步就会功亏一篑而且到最后极难找出错在哪里。我在搭建过程中就漏改了一个端口名，导致后面连接不上，幸好及时发现。此外，在对源代码DEBUG时，总是对我的main函数报错，错误提示在此处不是第一次发现，纠结了好久，最后在同学帮助下发现原因，我的工程是在hello world 基础上建立的，其中hello word.c文件里已经存在了一个main函数，而一个程序里不能同时存在两个主函数，得以解决了这个问题。通过这个实验，我还学会了基本IO读写函数的使用方法，更加熟练了Vivado和SDK的使用方法以及加深了对NEXYS 4 DDR 开发板认识。