

现代电子系统设计

FPGA 提高实验

学 号	2017011589
姓 名	吾尔开希
专 业	自动化
日 期	2019.7.1---7.12

目录

1.	实验内容	3
1.1	必做内容	3
1.2	自选内容	3
2.	设计方案	3
2.1	LCD 屏幕	3
2.2	IR 红外遥控器	3
3.	电路图	4
4.	模块选择与参数设置	4
4.1	NIOSII 软核模块	4
4.2	CLK 模块	4
4.3	SDRAM 控制模块	5
4.4	LCD 屏模块	5
4.5	IR 红外接收模块	5
5.	流程图	6
6.	实验结果	8
7.	实验中遇到的问题与解决方法	8
7.1	LCD 屏显示的问题	8
7.2	添加 LCD 屏模块时的问题	8
7.3	Verilog 基本语言错误	9
8.	体会、收获与建议	9

1. 实验内容

1.1 必做内容

设计一个以 NIOSII 软核微处理器和 Avalon 总线为核心的 SOPC 系统，实现用红外遥控器输入数据，然后将结果显示在 LCD 上，要求设计红外遥控和 LCD 与 Avalon 总线接口的 IP 核。

1.2 自选内容

用环形队列存储输入数据，用红外遥控的上下翻页键显示上一个或下一个输入。

2. 设计方案

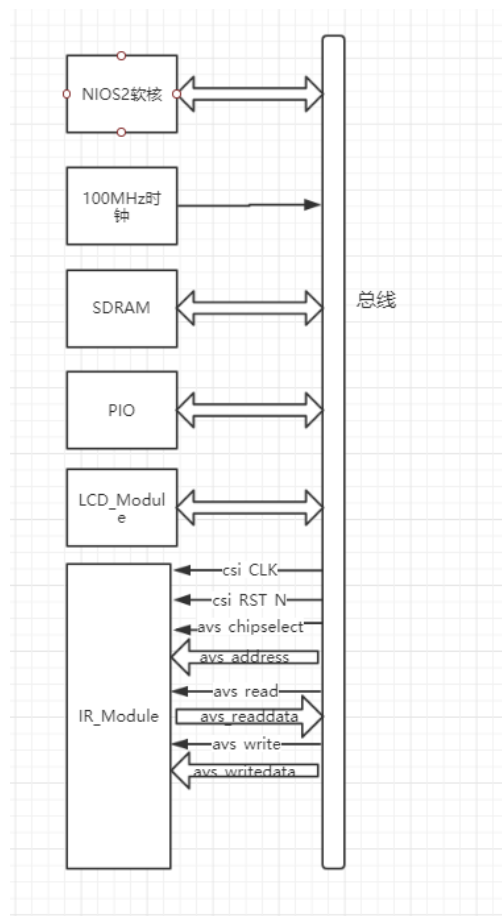
2.1 LCD 屏幕

LCD 提供的代码较为充足，硬件方面只需要将 LCD_Module 挂载在 Avalon 总线上即可，软件方面也只需要调用提供的库函数。

2.2 IR 红外遥控器

IR 模块提供了底层的驱动 Verilog 代码，但是 Module 代码并不全。一方面，需要增加总线接口；另一方面，系统总线上提供的时钟为 100MHz，而底层驱动仅需要 50MHz 时钟，所以需要二分频。

3. 电路图



Avalon 系统总线挂接图如图所示，由于 IR_Module 需要自行设计，所以总线接口详细列出。

4. 模块选择与参数设置

该程序是在 First_Nios 程序的基础上完成的，主要用到以下几个模块

4.1 NIOSII 软核模块

逻辑核心模块，从总线上读取 IR 红外模块的输入，控制 LCD 屏显示特定字符。

4.2 CLK 模块

时钟模块，提供 100MHz 频率时钟。

4.3 SDRAM 控制模块

NIOSII 软核将 SDRAM 作为程序内存使用。

4.4 LCD 屏模块

该模块接口完善，NIOSII 通过总线向该模块按一定的协议写入数据，即可实现 LCD 屏的显示。

4.5 IR 红外接收模块

总线上的 csi_CLK 是 100MHz 的，而 IR 模块的驱动仅需要 50MHz 时钟，所以进行二分频。

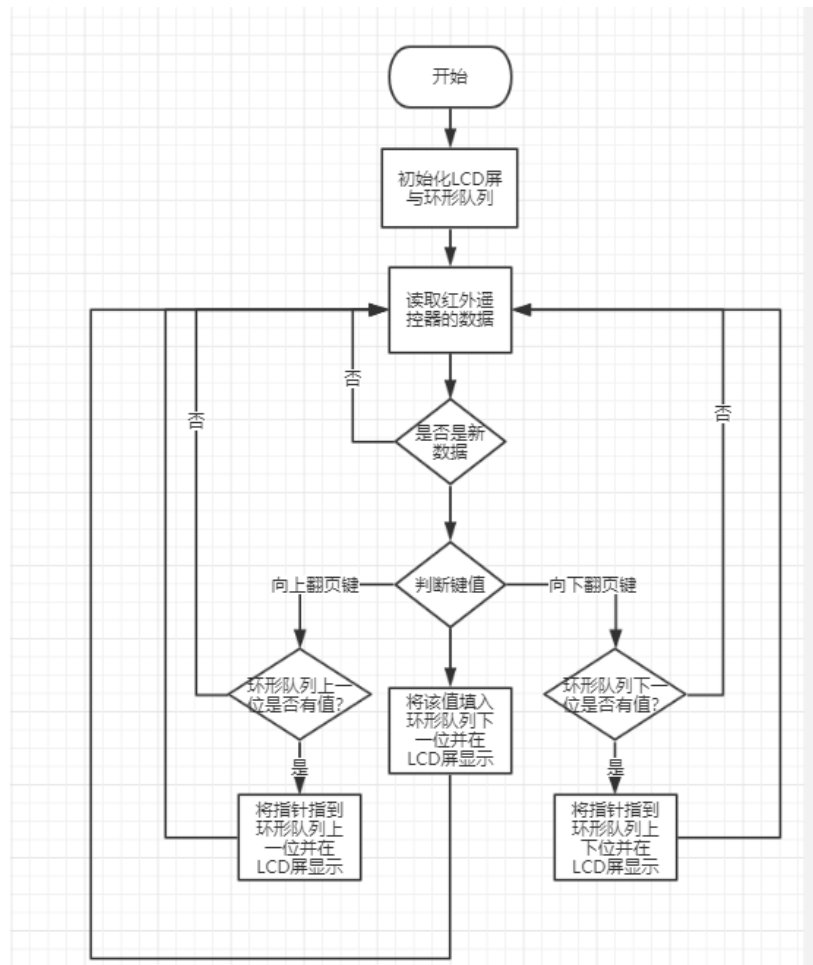
```
//csi_CLK is 100MHz, get 50MHz clk50
always@(posedge csi_CLK or negedge csi_RST_N)
begin
    if(!csi_RST_N)
        clk50 <= 0;
    else
        clk50 <= ~clk50;
end
```

NIOSII 软核可通过总线从该模块读到 32 位数据，其中，第 16 到 23 位是遥控器按钮原码。最高位是 coe_oDATA_READY，表示数据准备完成。

```
// 2 Avalon Read Process
// highest bit is coe_oDATA_READY
always @(posedge csi_CLK or negedge csi_RST_N)
begin
    if(!csi_RST_N) begin
        end
    else
        begin
            if((avs_chipselect == 1) &&(avs_read == 1))
            begin
                avs_readdata <= {coe_oDATA_READY, oDATA_wire[30:0]};
            end
        end
    end
end
```

oDATA_wire 是底层模块处理后的结果。

5. 流程图



NIOS 软核中的逻辑控制流程图如图所示。其中环形队列的初始化是指将环形队列中的值设为-1，而红外遥控器的键值都大于 0，因此可凭环形队列某一位是否为-1 来判断该位是否被写入过键值。该逻辑对应的主函数如下所示：

```
1. int main()
2. {
3.     LCD_Reset();
4.     LCD_Clear();
5.     printf("Hello from Nios II LCD1!\n");
6.     //LCD_Disp(1, 0, LCD_Data1, 5);
7.     delay(30000);
8.     LCD_Disp(1, 5, LCD_Data1, 5);
9.     char ir_data=-1, last_data = -1;
10.
11.     int i;
```

```

12.     for(i=0;i<lenOfQ;i++)
13.     {
14.         myQueue[i]=-1;
15.     }
16.
17.     while(1)
18.     {
19.         int pos_temp;
20.
21.         last_data = ir_data;
22.         ir_data = (*(pIR))>>16;
23.         if((ir_data != last_data))
24.         {
25.             pos_temp=pos;
26.             if(ir_data==26) //向上键
27.             {
28.                 pos = (pos-1+lenOfQ)%lenOfQ;
29.                 if(myQueue[pos]==-1) //上一位没有元素
30.                 {
31.                     pos=pos_temp;
32.                 }
33.             }
34.             else if(ir_data==30) //向下键
35.             {
36.                 pos = (pos+1)%lenOfQ;
37.                 if(myQueue[pos]==-1) //下一位没有元素
38.                 {
39.                     pos=pos_temp;
40.                 }
41.             }
42.             else
43.             {
44.                 pos=(pos+1)%lenOfQ;
45.                 myQueue[pos]=ir_data;
46.             }
47.
48.             char result = getChar(myQueue[pos]);
49.             unsigned char LCD_IRdata[9];
50.             printf("\r\nmy queue:\r\n");
51.             for(i=0;i<lenOfQ;i++)
52.             {
53.                 printf(" %d,",myQueue[i]);
54.             }
55.             sprintf(LCD_IRdata, "IR data:%c", result);

```

```
56.         LCD_Disp(2, 0, LCD_IRdata, 9);
57.         delay(3e4);
58.         LCD_Disp(1, 5, LCD_Data1, 5);
59.         delay(3e3);
60.     }
61.
62. }
63.
64. return 0;
65. }
```

6. 实验结果

达到了必做内容与自选内容的要求，实现了用红外遥控器输入数据，然后将结果显示在 LCD 上。

用环形队列存储输入数据，用红外遥控的上下翻页键显示上一个或下一个输入。效果视频已拷贝在实验室电脑中。

7. 实验中遇到的问题与解决方法

7.1 LCD 屏显示的问题

LCD 屏连续显示两条不同内容时会出现重影、显示错误的问题，解决方法是在 LCD 显示函数调用前用 for 循环延时一段时间。

7.2 添加 LCD 屏模块时的问题

一开始添加 LCD 模块时，我不太理解总线的运作机理与 Quartus 工程的关系，没有在 Quartus 顶层文件中定义 LCD 屏的引脚，这种情况下 LCD 屏一定是无法显示的。之后我回看了基础实验的指导，再联系计算机原理课程中学到的内容，大致明白了一些，Quartus 顶层文件的作用是将 FPGA 的外设与 Avalon 总线

接口模块，以及自定义的其他模块连接起来，这样 NIOS 软核就可以通过总线向这些模块进行读写控制。明白这一点后，FPGA 的综合实验做起来也更顺利了。

7.3 Verilog 基本语言错误

Verilog 定义 wire 或 reg 时，长度要写在变量名前（wire [31:0] output）而不是后面（wire output [31:0]）。

8. 体会、收获与建议

通过这次实验，理解并实践了 FPGA 的另一功能：NIOS 软核通过总线向硬件模块读写数据，感受到了 FPGA 的强大。在计算机原理课程中就学过借助总线的读写操作，这次实验将理论知识应用到实际中，加深了理解，提高了动手能力。

这次实验确实锻炼自主学习与理解能力，虽然基础实验中的步骤很详细，但是要深入理解还是要靠自己摸索，当我悟到总线与 Quartus 顶层文件的关系时确实感觉很有成就感。