

实验九 键盘扫描显示实验

一、实验目的

- 1、了解普通 4×4 键盘扫描的原理。
- 2、进一步加深七段码管显示过程的理解。

二、硬件要求

- 1、4×4 键盘阵列。
- 2、FPGA 主芯片。
- 3、可变时钟源。
- 4、七段码显示区。

三、实验原理

本实验主要完成的实验是完成 4×4 键盘扫描的，然后获取其键值，并对其进行编码，从而进行按键的识别，并将相应的按键值进行显示。

键盘扫描的实现过程如下：对于 4×4 键盘，通常连接为 4 行、4 列，因此要识别按键，只需要知道是哪一行和哪一列即可，为了完成这一识别过程，我们的思想是，首先固定输出 4 行为高电平，然后输出 4 列为低电平，在读入输出的 4 行的值，通常高电平会被低电平拉低，如果读入的 4 行均为高电平，那么肯定没有按键按下，否则，如果读入的 4 行有一位为低电平，那么对应的该行肯定有一个按键按下，这样便可以获取到按键的行值。同理，获取列值也是如此，先输出 4 列为高电平，然后在输出 4 行为低电平，再读入列值，如果其中有哪一位为低电平，那么肯定对应的那一列有按键按下。

获取到行值和列值以后，组合成一个 8 位的数据，根据实现不同的编码在对每个按键进行匹配，找到键值后在 7 段码管显示。

四、实验内容及步骤

本实验内容是完成 4×4 键盘的扫描，然后将正确的键值进行显示，实验步骤如下：

- 1、编写键盘扫描和显示的 VHDL 代码。
- 2、用 QuartusII 对其进行编译仿真。
- 3、在仿真确定无误后，选择芯片 ACEX1K EP1K10TC100-3。
- 4、给芯片进行管脚绑定，在此进行编译。
- 5、根据自己绑定的管脚，在实验箱上对键盘接口、显示接口和 FPGA 之间进行正确连线。
- 6、给目标板下载代码，在 4×4 键盘输入键值，观看实验结果。

五、实验连线

如果是调用的本书提供的 VHDL 代码，则实验连线如下：

Clk：FPGA 工作时钟信号，大约位 5KHz 至 50KHz 即可。

Kr[0:3]：分别接 4×4 键盘部分的 R1、R2、R3 和 R4。

Kc[0:3]：分别接 4×4 键盘部分的 C1、C2、C3 和 C4。

Sa、Sb、Sc：接七段码显示区的 Sel0、Sel1 和 Sel2。

A、B、C、D、E、F、G：接七段码显示区的 A、B、C、D、E、F 和 G。

六、实验部分 VHDL 代码

```
library ieee;
```

```
-----  
entity keyscan is  
  port( clk      : in      std_logic;  --Clock Signal  
        kr      : inout  std_logic_vector(3 downto 0);  --Row of Key Scan  
        kc      : inout  std_logic_vector(3 downto 0);  --Colume of Key Scan  
        a,b,c,d,e,f,g : out    std_logic;  --7 segment driver  
        sa,sb,sc   : out    std_logic);  --Display Select  
end keyscan;
```

```

architecture behave of keyscan is
    signal display      : std_logic_vector(6 downto 0);
    signal dcount       : std_logic_vector(2 downto 0);
    signal keyr, keyc   : std_logic_vector(3 downto 0);
    signal kcount       : std_logic_vector(2 downto 0);
    signal kflag1, kflag2 : std_logic;
    signal buff0, buff1, buff2, buff3, buff4, buff5, buff6, buff7 : std_logic_vector(6
downto 0);
begin
    process(clk) --In this process, two counts will accumulate
    begin
        if(clk'event and clk='1') then
            dcount<=dcount+1;
            kcount<=kcount+1;
        end if;
    end process;
    process(clk) --In this process, the main task is to scan keyboard
    begin
        if(clk'event and clk='1') then
            if(kcount=0) then
                kr<="1111";
                kc<="0000";
            elsif(kcount=1) then
                keyr<=kr;
                .....
                keyc<=kc;
            end if;
        end if;
    end process;
    process(clk) -- change the key flags
    begin
        if(clk'event and clk='1') then
            if(kcount=4 and keyr="1111") then
                kflag1<='0';
            elsif(kcount=4) then
                kflag1<='1';
            end if;
            kflag2<=kflag1;
        end if;
    end process;
    process(clk) -- shift the displays
    begin
        if(clk'event and clk='1') then
            if(kcount=5 and kflag1='1' and kflag2='0') then
                buff0<=buff1;
                .....
                buff6<=buff7;
            end if;
        end if;
    end process;
end architecture;

```

```

        end if;
    end process;
process(clk) -- get the result of the key
begin
    if(clk'event and clk='1') then
        if(kcount=5) then
            if(keyr="1110") then
                case keyc is
                    when "1110"=>buff7<="0111111"; --display 0
                    .....
                    when "0111"=>buff7<="1110001"; --display f
                    when others=>buff7<=buff7;      --no change
                end case;
            end if;
        end if;
    end if;
end process;
process(clk) --display process
begin
    if(clk'event and clk='1') then
        sa<=dcount(0);
        sb<=dcount(1);
        sc<=dcount(2);
        case dcount is
            when "111"=>display<=buff0;
            .....
            when "110"=>display<=buff7;
            when others=>display<="0000000";
        end case;
    end if;
end process;
process(clk) --In this process, a, b, c, d, e, f, g will output
begin
    if(clk'event and clk='1') then
        a<=display(0);
        .....
        g<=display(6);
    end if;
end process;
end behave;

```

七、实验报告要求

- 1、总结 FPGA 是如何识别按键的？与单片机读取键值有何不同？
- 2、在深入理解了 4×4 键盘实现的原理基础上，试试利用 VHDL 在目标器件 FPGA/CPLD 上实现 PS/2 键盘接口。
- 3、比较 4×4 键盘与 PS/2 键盘接口用 FPGA 实现方法的异同点。