实验九 矩阵键盘

1 实验设计目标

在键盘中按键数量较多时，为了减少 I/O 口的占用，通常将按键排列成矩阵形式。在矩阵键盘中，行线和列线不直接连通，而是通过一个按键进行连接。本实 验实现的功能非常简单，为矩阵键盘的十六个按键编号为 0~15，按下哪一个按键，数码管就显示该按键对应编号的十六进制数。

2 实验设计思路

如下是本文所使用的 FPGA 开发板上板载矩阵键盘的原理图。

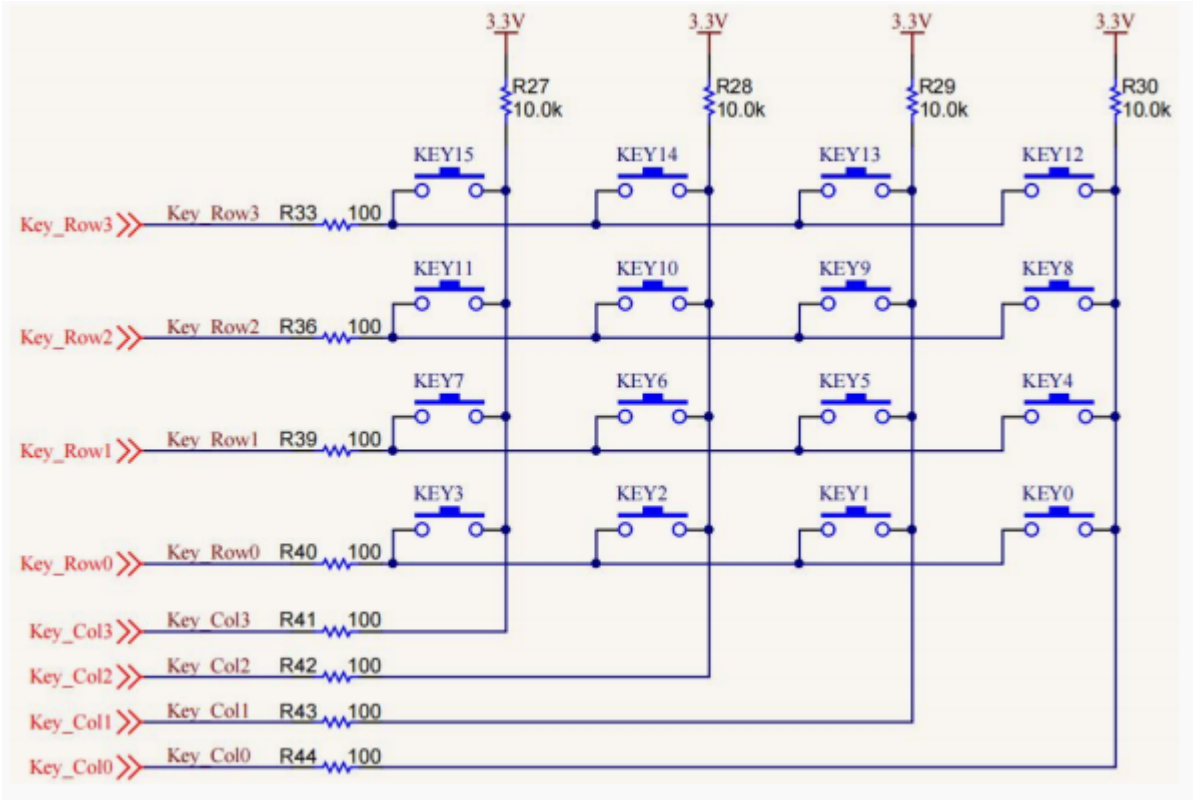
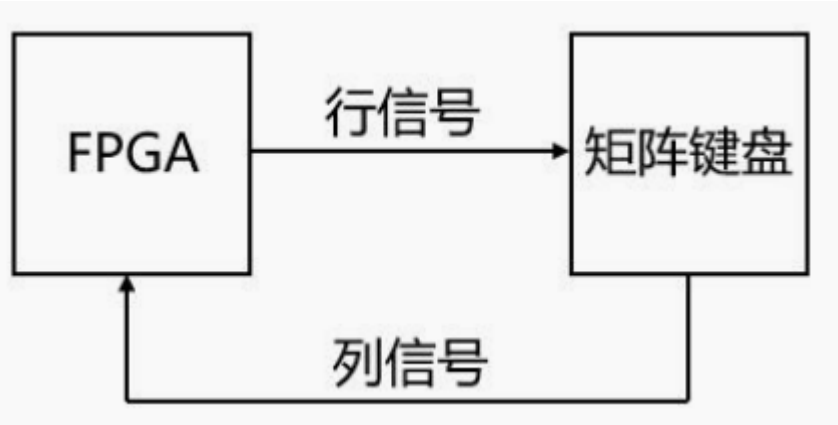


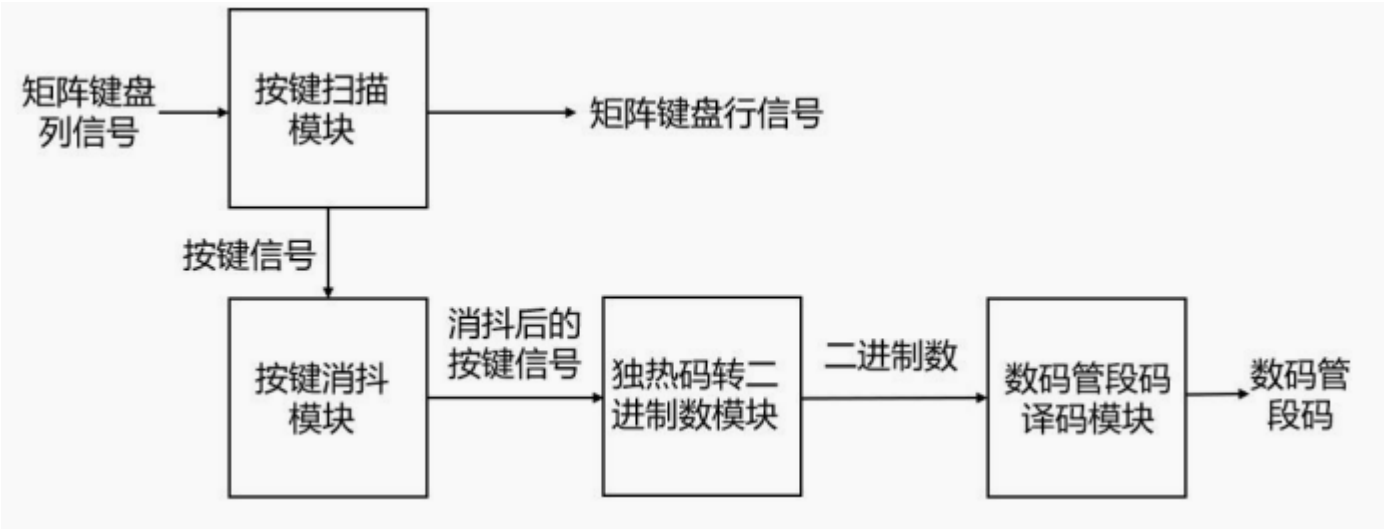
图 9.1 矩阵键盘原理图

从图 9.2 中可以看到，键盘的列线已经都设置了上拉电阻，在没有按下按键时，键盘列线的为高电平。若给键盘的某一行输入低电平，并按下该行的一个按键，那么该按键连接的列线的输出就会被拉低。因此，使用 FPGA 驱动矩阵键盘时，一般通过输送高/低电平给矩阵键盘的行信号，并读取键盘列信号来判断是否有按键按下。举个例子，按照上图中，Key_Row[0] ~ Key_Row[3]为矩阵键盘的行信号，Key_Col[0] ~ Key_Col[3]为矩阵键盘的列信号，若给 Key_Row[0]输送低电平，读取键盘列信号 Key_Col[3:0]的结果为 1011，则说明 KEY2 被按下了。



9.2 基于 FPGA 的矩阵键盘驱动

对于列线已设置上拉的矩阵键盘，应该通过对所用的行线输送低电平信号，读取列信号的值，如果某一列的信号低电平，则说名该行列交叉处的按键被按下。 由于本例中将使用所有的按键，如果对所有的行线都输送低电平信号，即使读到了某一列的电平信号为低也无法判定是哪一行的按键，这时就需要采用行扫描法。其原理也非常简单，即按照合适的扫描频率依次给各行输送低电平信号，当前行列电平信号均为低的按键就是被按下的按键。



9.3 顶层电路框图

根据实例二的需求进行功能划分，一共有四个功能模块，分别是按键扫描模块、按键消抖模块、独热码转 BCD 码模块和数码管段码译码模块。其中，按键扫描模块负责监测矩阵键盘的列信号并对矩阵键盘进行逐行扫描，并将判定后的按键信号输出；按键消抖模块负责接收按键扫描模块得到的按键信号并进行消抖处理；独热码转二进制数模块则用于将消抖后的按键信号转换为二进制数；最后由 数码管段码译码模块进行译码并驱动数码管的显示。

2.1 逻辑功能设计

该工程包含顶层模块 keyboard_instance 与底层模块 keyboard_scan、key_filter、onehot2binary 和Digitron_NumDisplay_module。整个工程的模块功能图参考图 9.3即可。下面介绍一下顶层模块各主要引脚的功能：

1. clk_50M：50MHz 的基准时钟信号输入。
2. RSTn：系统复位输入信号。低电平有效，控制是否开始扫描矩阵键盘。
3. col：矩阵键盘的列信号。
4. row：矩阵键盘的行信号，通过顺序对4个信号赋值 0，获取按键信息。
5. Digitron_Out：七段数码管的显示输出，共有八位总线。
6. DigitronCS_Out：数码管的片选信号，共有四位总线。

2.2 程序设计

2.2.1 顶层模块keyboard_instance.v

```
module keyboard_instance(clk_50M,RSTn,col,row,Digitron_Out,DigitronCS_Out,LED_Out);

    input clk_50M;
    input RSTn ;// SW0
    input [3:0] col;
    output [3:0] row;
    output [7:0] Digitron_Out;
    output [3:0] DigitronCS_Out;
    output [7:0] LED_Out;

    wire [15:0] key;

    keyboard_scan U1(
        .clk(clk_50M),
        .RSTn(RSTn),
        .col(col),
        .row(row),
        .key(key)
    );

    wire [15:0] key_deb;
    key_filter U2(
        .clk(clk_50M),
        .rstn(RSTn),
        .key_in(key),
        .key_deb(key_deb)
    );

    wire [3:0] data_disp;
    onehot2binary U3(
        .clk(clk_50M),
        .onehot(key_deb),
        .binary(data_disp)
    );


    Digitron_TimeDisplay_module U5
    (
        .CLK(clk_50M),

```

```
        .data(data_disp),
        .Digitron_Out(Digitron_Out),
        .DigitronCS_Out(DigitronCS_Out)
    );

    assign LED_Out = 8'b0;

endmodule
```

2.2.2 功能模块keyboard_scan.v

```
module keyboard_scan(clk,RSTn,col,row,key);

    input clk;
    input RSTn ;
    input [3:0] col;
    output reg [3:0] row ;
    output reg [15:0] key;

    reg [31:0] cnt = 0;
    reg scan_clk = 0;

    always@(posedge clk) begin
        if(cnt == 2499) begin
            cnt <= 0;
            scan_clk <= ~scan_clk;
        end
        else
            cnt <= cnt + 1;
    end

    always@(posedge scan_clk)
    begin
        if(!RSTn)
                row <= 4'b1110;
        else
                row <= {row[2:0],row[3]};
    end

    always@(negedge scan_clk)
        case(row)
            4'b0111 : key[15:12] <= col;
            4'b1011 : key[11:8] <= col;
            4'b1101 : key[7:4] <= col;
            4'b1110 : key[3:0] <= col;
            default : key <= 0;
        endcase

endmodule
```

通过移位操作周期性地为矩阵键盘的行信号输送 1110、1101、1011、0111 的电平信号，实现行扫描。可以注意到，在时序设计上，行信号的移位操作是在扫描时钟的上升沿进行的，而列信号的采集是

在扫描时钟的下降沿进行的。分析代码不难得出，如果有按键按下（这里仅考虑一次只按下一个按键），该模块的按键信号输出 key 会是一个 16bit 的独热码。

2.2.3 功能模块onehot2binary.v

```
module onehot2binary(clk,onehot,binary);

    input clk;
    input [15:0] onehot;
    output reg [3:0] binary;

    always@(posedge clk) //濡傛灉娌℃湁鎸夐敭鎸変笅 onehot涓嶅姩,鍒欏緱鍑烘淇濇寔
        case(onehot)
            16'h0001 : binary <= 4'b0000;
            16'h0002 : binary <= 4'b0001;
            16'h0004 : binary <= 4'b0010;
            16'h0008 : binary <= 4'b0011;
            16'h0010 : binary <= 4'b0100;
            16'h0020 : binary <= 4'b0101;
            16'h0040 : binary <= 4'b0110;
            16'h0080 : binary <= 4'b0111;
            16'h0100 : binary <= 4'b1000;
            16'h0200 : binary <= 4'b1001;
            16'h0400 : binary <= 4'b1010;
            16'h0800 : binary <= 4'b1011;
            16'h1000 : binary <= 4'b1100;
            16'h2000 : binary <= 4'b1101;
            16'h4000 : binary <= 4'b1110;
            16'h8000 : binary <= 4'b1111;
        endcase
endmodule
```

独热码转二进制模块，实现将消抖后的按键信号转化为可供现成的数码管段码译码模块进行显示译码的编码。由于位数较少，直接使用查表的方法实现。另外，该模块引入了时序电路，还含有实例一中那个带使能端的寄存器类似的功能，即松开按键后，该模块的输入为 16bit 的全零信号，此时按照如下代码的描述将不会更新编码输出，而是保持上次按下按键时相同的输出，故不需要一直按住按键来保持数码管的显示。

2.2.4 约束文件keyboard.adc

以下是 Anlogic FPGA 的 IO Constraint，CLK 时钟输入信号、数码管片选信号 DigitronCS_Out 和数码管输出显示信号 Digitron_Out 的配置方式与实验六相同， DIG1~DIG2 显示按键信息，复位信号

RSTn 与 SW0 相连，col 和 row 与矩阵键盘 8 个引脚相连。

```
set_pin_assignment      { DigitronCS_Out[0] }   { LOCATION = C9; IOSTANDARD = LVCMOS33; }
set_pin_assignment      { DigitronCS_Out[1] }   { LOCATION = B6; IOSTANDARD = LVCMOS33; }
set_pin_assignment      { DigitronCS_Out[2] }   { LOCATION = A5; IOSTANDARD = LVCMOS33; }
set_pin_assignment      { DigitronCS_Out[3] }   { LOCATION = A3; IOSTANDARD = LVCMOS33; }
set_pin_assignment      { Digitron_Out[0] }     { LOCATION = A4; IOSTANDARD = LVCMOS33; }
set_pin_assignment      { Digitron_Out[1] }     { LOCATION = A6; IOSTANDARD = LVCMOS33; }
set_pin_assignment      { Digitron_Out[2] }     { LOCATION = B8; IOSTANDARD = LVCMOS33; }
set_pin_assignment      { Digitron_Out[3] }     { LOCATION = E8; IOSTANDARD = LVCMOS33; }
set_pin_assignment      { Digitron_Out[4] }     { LOCATION = A7; IOSTANDARD = LVCMOS33; }
set_pin_assignment      { Digitron_Out[5] }     { LOCATION = B5; IOSTANDARD = LVCMOS33; }
set_pin_assignment      { Digitron_Out[6] }     { LOCATION = A8; IOSTANDARD = LVCMOS33; }
set_pin_assignment      { Digitron_Out[7] }     { LOCATION = C8; IOSTANDARD = LVCMOS33; }
set_pin_assignment      { RSTn }        { LOCATION = A9; IOSTANDARD = LVCMOS33; }
set_pin_assignment      { clk_50M }     { LOCATION = R7; IOSTANDARD = LVCMOS33; }
set_pin_assignment      { col[0] }      { LOCATION = E11; IOSTANDARD = LVCMOS33; }
set_pin_assignment      { col[1] }      { LOCATION = D11; IOSTANDARD = LVCMOS33; }
set_pin_assignment      { col[2] }      { LOCATION = C11; IOSTANDARD = LVCMOS33; }
set_pin_assignment      { col[3] }      { LOCATION = F10; IOSTANDARD = LVCMOS33; }
set_pin_assignment      { row[0] }      { LOCATION = E10; IOSTANDARD = LVCMOS33; }
set_pin_assignment      { row[1] }      { LOCATION = C10; IOSTANDARD = LVCMOS33; }
set_pin_assignment      { row[2] }      { LOCATION = F9; IOSTANDARD = LVCMOS33; }
set_pin_assignment      { row[3] }      { LOCATION = D9; IOSTANDARD = LVCMOS33; }

set_pin_assignment      { LED_Out[0] }  { LOCATION = B14; IOSTANDARD = LVCMOS33; }
set_pin_assignment      { LED_Out[1] }  { LOCATION = B15; IOSTANDARD = LVCMOS33; }
set_pin_assignment      { LED_Out[2] }  { LOCATION = B16; IOSTANDARD = LVCMOS33; }
set_pin_assignment      { LED_Out[3] }  { LOCATION = C15; IOSTANDARD = LVCMOS33; }
set_pin_assignment      { LED_Out[4] }  { LOCATION = C16; IOSTANDARD = LVCMOS33; }
set_pin_assignment      { LED_Out[5] }  { LOCATION = E13; IOSTANDARD = LVCMOS33; }
set_pin_assignment      { LED_Out[6] }  { LOCATION = E16; IOSTANDARD = LVCMOS33; }
set_pin_assignment      { LED_Out[7] }  { LOCATION = F16; IOSTANDARD = LVCMOS33; }
```

3 上板实验

3.1 编译

按之前文档进行编译，生成bit文件；

3.2 上板实验

当按下矩阵键盘时候，数码管最低位会以 16 进制显示编号。