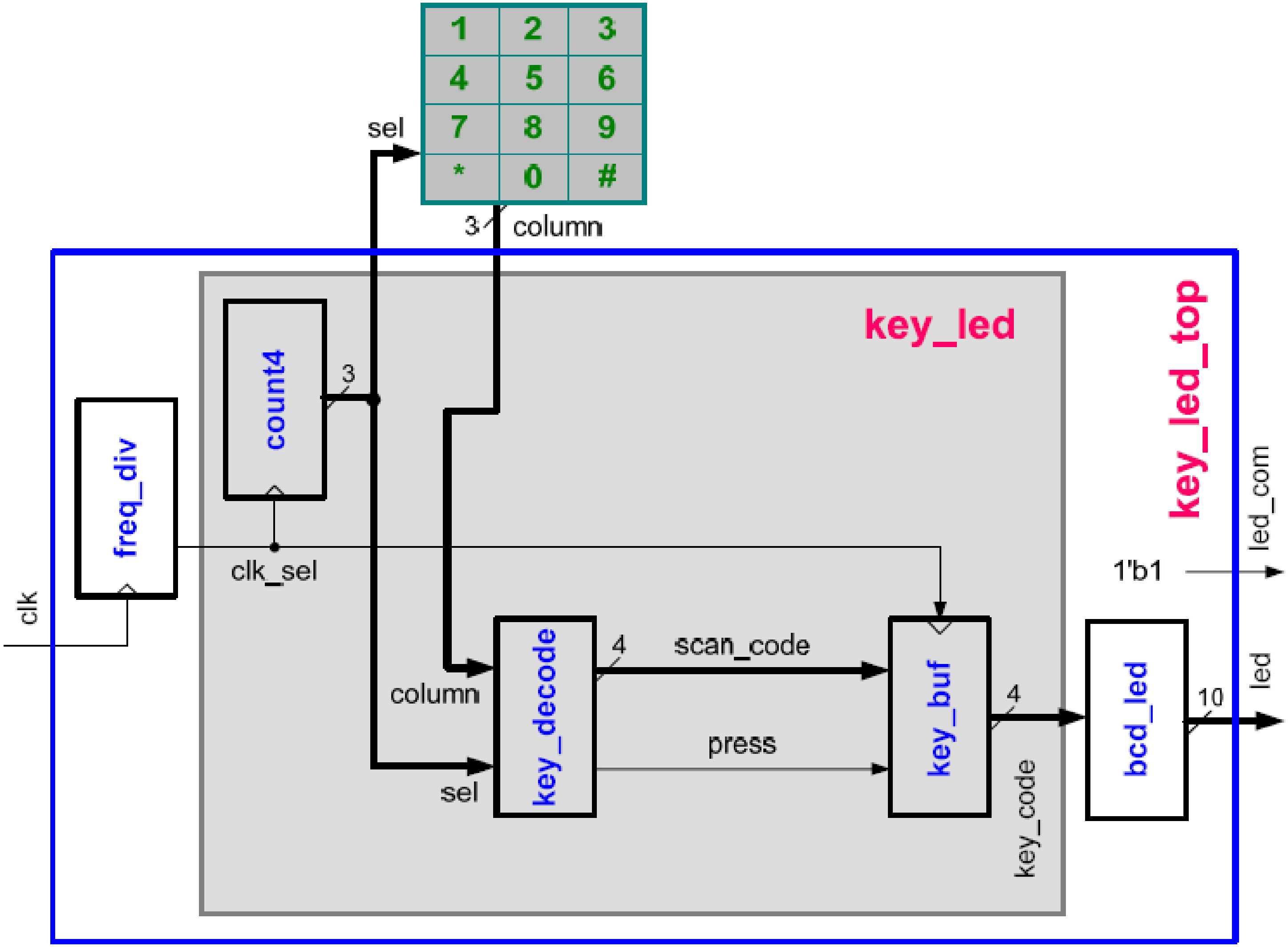
FPGA Lab-05

Keyboard(1)

Lab Content

- Using the keyboard as input tool, the keyed-in numbers from 0 to 9 are indicated by 10 different LEDs.
- Lab 1: Keyboard Detection Circuitry
- Lab 2：Keyboard Buffer
- Lab 3: BCD code to LED output circuitry.

System Block Diagram

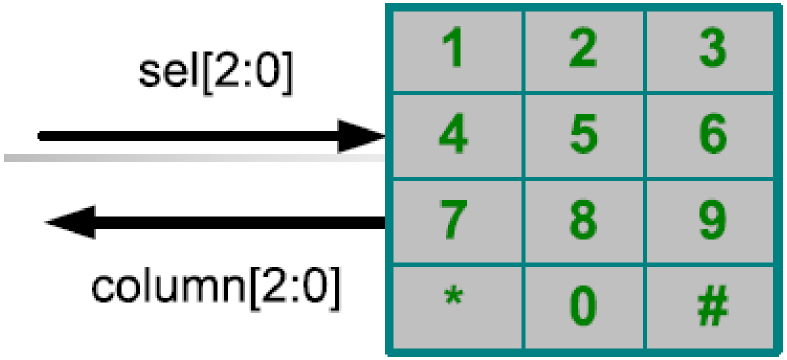


Verilog Codes

- key_decode.v
- key_buf.v
- key_led.v
- key_led_top.v
- bcd_led.v
- count4.v (Please write it yourself)
- freq_div.v(Please write it yourself)

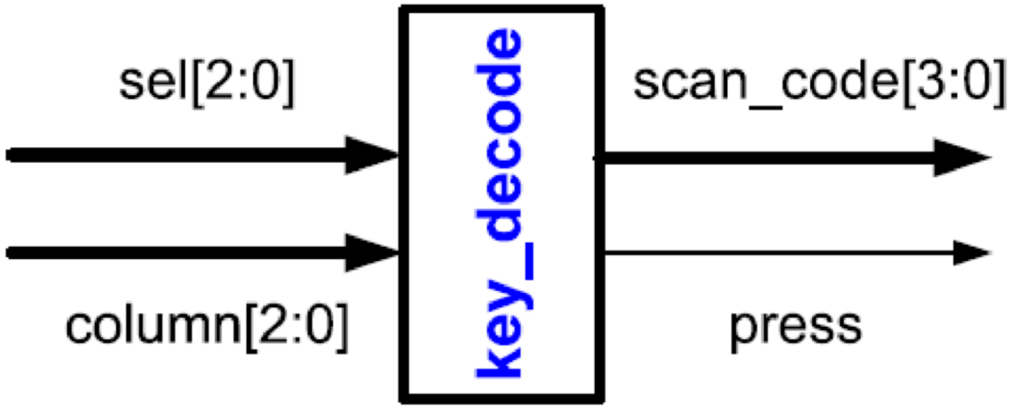
How the keyboard works

- When sel= 000, if column = 011, it means the code typed is 1 (0001).
- When sel= 000, If column = 101, then the code entered is 2 (0010).
- When sel= 000, If column = 110, then the keyed-in code is 3 (0011)
- When sel= 001, If column = 011, then the keyed-in code is 4 (0100)
- When sel= 001, If column = 101, then the keyed-in code is 5 (0101)
- When sel= 001, If column = 110, then the keyed-in code is 6 (0110)
- When sel= 010, if column = 011, then the keyed-in code is 7 (0111)
- When sel= 010, If column = 101, then the keyed-in code is 8 (1000)
- when sel= 010, If column = 110, then the keyed-in code is 9 (1001)
- When sel= 011, if column = 101, then the keyed-in code is 0 (0000).



Step 1: Keyboard detection circuit

- The input signal (sel[2:0]) and the input signal (column[2:0]) of the keyboard are used to determine whether a number is keyed in or not (determined by press).
- If there is a keyed-in number (press is 1), the BCD code (scan_code[3:0]) of the keyed-in number is gene



VerilogCode -key_decode.v(1/3)

```
module key_decode(sel, column, press, scan_code);
input[2:0]sel;
input[2:0] column;
output press;
output[3:0] scan_code;
reg[3:0] scan_code;
reg press;
always@(sel or column) begin
case(sel)
```

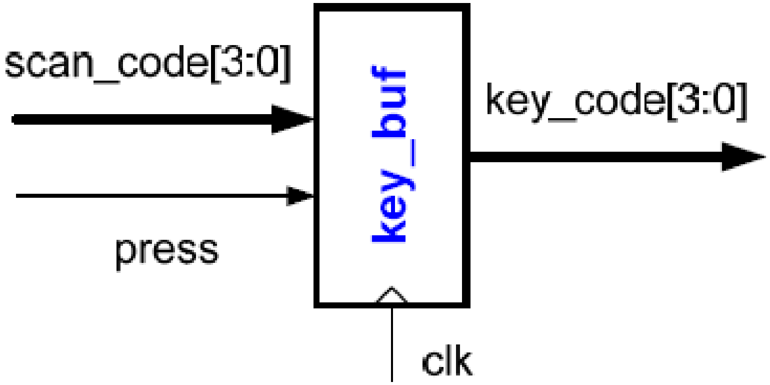
VerilogCode -key_decode.v(2/3)

```
3'b000:
case(column)
3'b011: begin scan_code= 4'b0001; press = 1'b1; end // 1
3'b101: begin scan_code= 4'b0010; press = 1'b1; end // 2
3'b110: begin scan_code= 4'b0011; press = 1'b1; end // 3
default: begin scan_code= 4'b1111; press = 1'b0; end
endcase
3'b001:
case(column)
~~your code~~~
endcase
3'b010:
case(column)
~~your code~~~
endcase
```

VerilogCode -key_decode.v(3/3)

- 3'b011:
- case(column)
- ~~your code~~~
- endcase
- default:
- begin scan_code= 4'b1111; press = 1'b0; end
- endcase
- end
- endmodule

Step 2: Keyboard Buffer



- Store the number typed by the keyboard. Whenever clk is triggered,
- If press is 1, it means there is a number typed on the keyboard (scan_code[3:0]). The scan_code will be stored in key_code[3:0].
- If press is 0 when clk is triggered, it means there is no number typed on the keyboard, and the content of key_code remains unchanged.

VerilogCode -key_buf.v

```
module key_buf(clk, rst, press, scan_code, key_code);
input clk, rst, press;
input[3:0] scan_code;
output[3:0]key_code;
reg[3:0]key_code;
always@(posedge clk or posedge rst) begin
if(rst)
key_code= 4'b1111;// initial value
else
key_code= press ? ~~your code~~~;
end
endmodule
```

Lab 3: BCD code to LED output circuitry.

- bcd_led.v
- key_led.v
- key_led_top.v

bcd_led.v

```
module bcd_led(key_code, led);
input[3:0]key_code;
output[9:0]led;
reg[9:0]led;
always@(key_code) begin
case(key_code)
4'b0000: led = 10'b0000000001;  //0 to 9的LED display
~~your code~~~
default: led = 10'b0000000000;
endcase
end
endmodule
```

key_led

```
module key_led(clk_sel, reset, column, sel, key_code);
input clk_sel, reset;
input[2:0]column;
output[2:0]sel;
output[3:0] key_code;
wire press;
wire[3:0] scan_code, key_code;
count4(~~your code~~~ );
key_decode(~~your code~~~);
key_buf(~~your code~~~);
endmodule
```

key_led_top.v

```
module key_led_top(clk, reset, column, sel, led, led_com);
input clk, reset;      //pinW16,C16
input[2:0]column;    // pin AA13, AB12, Y16
output[2:0]sel;       // pin AB10, AB11, AA12
output[9:0]led;       // pin E2, D3, C2, C1 ,L2, L1, G2, G1, U2, N1
output led_com;       // pin N20
assign led_com= 1'b1;
wire clk_sel;
wire[3:0] key_code;
freq_div#(13) (~~your code~~~ );
key_led(~~your code~~~ );
bcd_led(~~your code~~~ );
endmodule
```