



ELEX 7660: Pre-Lab 2

Matrix Keypad Decoder

Table of Contents

1	Modules	3
1.1	Colseq	3
1.2	Kpdecode	4
1.3	Decode7	5
1.4	Simulation Results	5

Table of Figures

Figure 1: Simulation Results	5
------------------------------------	---

1 Modules

1.1 Colseq

// colseq.sv - ELEX 7660 - Sequentially pulls each column of the keypad low until a matching keypress is detected.

// Nicholas Huttemann 2018-01-22

```

module colseq (input logic [3:0] kpr ,
               input logic clk, reset_n,
               output logic [3:0] kpc = 'b0111);

// The four states corresponding to each column
int states [4] = {'b0111, 'b1011, 'b1101, 'b1110};
logic [1:0] state = 0;
logic hold;

always_comb
    if (kpr == 'b1111)                // If there was a keypress, hold
        hold <= 0;
    else
        hold <= 1;

always_ff @ (posedge clk, negedge reset_n) begin
    if (~reset_n)
        kpc <= 'b0111;
    else
        if (~hold) begin                // If not holding, cycle through the columns
            kpc = states[state];
            if (state == 3)
                state = 0;
            else
                state++;
        end
end
endmodule

```

1.2 Kpdecode

```

/* kpdecode.sv - ELEX 7660 - Given the detected row ('kpr') and column ('kpc')
location of a press on the keypad,
set 'num' to the decoded key value and drive the 7-segment with 'kphit'.
Nicholas Huttemann 2018-01-22 */

module kpdecode (input logic [3:0] kpc,
                 input logic [3:0] kpr,
                 output logic [3:0] num,
                 output logic kphit);

// Define a virtual keypad as a 2D array
int keypad [4][4] = '{ {1,2,3,10},
                        {4,5,6,11},
                        {7,8,9,12},
                        {14,0,15,13}};

int row;
int col;

always_comb begin
    num = 0;
    row = 0;
    col = 0;
    if (kpr != 'b1111) begin // A key has been pressed, decode its value
        kphit = 1;
        case (kpc) // Which column was detected?
            'b0111 : col = 0;
            'b1011 : col = 1;
            'b1101 : col = 2;
            'b1110 : col = 3;
            default: col = 0;
        endcase
        case (kpr) // Which row was detected?
            'b0111 : row = 0;
            'b1011 : row = 1;
            'b1101 : row = 2;
            'b1110 : row = 3;
            default: row = 0;
        endcase

        num = keypad[row][col]; // The decoded value is located in 'keypad'
                                // at the detected row and column index.
    end
    else // No key was pressed
        kphit = 0;
end
endmodule

```

1.3 Decode7

// decode7.sv - ELEX 7660 - Converts 'num', an integer 0 - 15, into an 8-bit vector 'leds' that will light up corresponding segments on a 7-segment LED.
 // Nicholas Huttemann 2018-01-22

```
module decode7 ( input logic [3:0] num,
                 output logic [7:0] leds);
```

```
always_comb
  case (num)
    0 : leds = 8'b11000000;
    1 : leds = 8'b111111001;
    2 : leds = 8'b10100100;
    3 : leds = 8'b10110000;
    4 : leds = 8'b10011001;
    5 : leds = 8'b10010010;
    6 : leds = 8'b10000010;
    7 : leds = 8'b111111000;
    8 : leds = 8'b10000000;
    9 : leds = 8'b10010000;
    4'ha: leds = 8'b10001000;
    4'hb: leds = 8'b10000011;
    4'hc: leds = 8'b11000110;
    4'hd: leds = 8'b10100001;
    4'he: leds = 8'b10000110;
    4'hf: leds = 8'b10001110;
    default: leds = 8'b11000000;
  endcase
endmodule
```

1.4 Simulation Results

```
VSIM3> run -all
# on reset kpc = 0111 (should be 0111)
# PASS: key 0 => num 1 (should be 1)
# PASS: num 1 => led f9 (should be f9)
# PASS: key 1 => num 2 (should be 2)
# PASS: num 2 => led a4 (should be a4)
# PASS: key 2 => num 3 (should be 3)
# PASS: num 3 => led b0 (should be b0)
# PASS: key 3 => num a (should be a)
# PASS: num 10 => led 88 (should be 88)
# PASS: key 4 => num 4 (should be 4)
# PASS: num 4 => led 99 (should be 99)
# PASS: key 5 => num 5 (should be 5)
# PASS: num 5 => led 92 (should be 92)
# PASS: key 6 => num 6 (should be 6)
# PASS: num 6 => led 82 (should be 82)
# PASS: key 7 => num b (should be b)
# PASS: num 11 => led 83 (should be 83)
# PASS: key 8 => num 7 (should be 7)
# PASS: num 7 => led f8 (should be f8)
# PASS: key 9 => num 8 (should be 8)
# PASS: num 8 => led 80 (should be 80)
# PASS: key 10 => num 9 (should be 9)
# PASS: num 9 => led 90 (should be 90)
# PASS: key 11 => num c (should be c)
# PASS: num 12 => led c6 (should be c6)
# PASS: key 12 => num e (should be e)
# PASS: num 14 => led 86 (should be 86)
# PASS: key 13 => num 0 (should be 0)
# PASS: num 0 => led c0 (should be c0)
# PASS: key 14 => num f (should be f)
# PASS: num 15 => led 8e (should be 8e)
# PASS: key 15 => num d (should be d)
# PASS: num 13 => led a1 (should be a1)
# ** Note: $stop : C:/Users/A00920439/Desktop/Lab2/lab2_tb.sv(110)
# Time: 113500 ns Iteration: 1 Instance: /lab2_tb
# Break in Module lab2_tb at C:/Users/A00920439/Desktop/Lab2/lab2_tb.sv line 110
```

Figure 1: Simulation Results