```
module Top_Module_main(
    input clkin, input btnR, input btnU, input btnD, input btnC, input btnL,
    input [15:0]sw,
    output [6:0]seg, output dp, output [3:0]an, [15:0]led
    );

    //some module with 3 input, 1 output. IN: 16bit bus, btnC, clk
    wire up_btn, down_btn, btnU_press;
    wire digsel, clk;
    wire utc, dtc, btnC_hold;
    wire [15:0]bit16out;
    assign led = sw;    //turns on the LED to the corresponding switch

    lab4_clks slowit (.clkin(clkin), .greset(btnR), .clk(clk), .digsel(digsel));
//DONE

    Edge_Detector edge_dct1(.clk(clk), .btn(btnU), .out(btnU_press));  //1 input for
OR gate
    assign btnC_hold = (bit16out == 16'b1111111111110011) ? 0  :      //fffc
                       (bit16out == 16'b1111111111111011) ? 0  :      //fffd
                       (bit16out == 16'b1111111111110111) ? 0  :      //fffe
                       (bit16out == 16'b1111111111111111) ? 0  : 1;   //ffff
    assign up_btn = (btnC_hold & btnC) | btnU_press;

    Edge_Detector edge_dct2(.clk(clk), .btn(btnD), .out(down_btn)); //input for
16bit cntr
    countUD16L counter1 (.clk(clk), .Up(up_btn), .Dw(down_btn),
        .LD(btnL), .Din(sw), .UTC(utc), .DTC(dtc), .Q(bit16out));   //may change UTC
and DTC outputs



    wire [3:0]Qring;
    RingCounter ring_cntr (.digsel(digsel), .clk(clk), .Q(Qring));

    assign an[0] = !Qring[0];
    assign an[1] = !Qring[1];
    assign an[2] = !Qring[2];
    assign an[3] = !Qring[3];
    assign dp = ~((Qring[1] & utc) | (Qring[2] & dtc)) ;
   // assign dp = ~(Qring[2] & utc);
    //Selector module with IN: 3bit sel from RingCounter, OUT: 4bit H
    wire [3:0]sel;
    Selector select(.sel(Qring), .N(bit16out), .H(sel));
   // assign dp = (~Qring[2] & ~utc) | (~Qring[1] & ~dtc);//~(utc | dtc) & (Qring[2]
| Qring[3]);
```

```
    //hex7seg module with IN 4bit bus from Selector, OUT: 7bit to svnseg_disp
     hex7seg segment_disp (.n(sel), .seg(seg));




endmodule
```