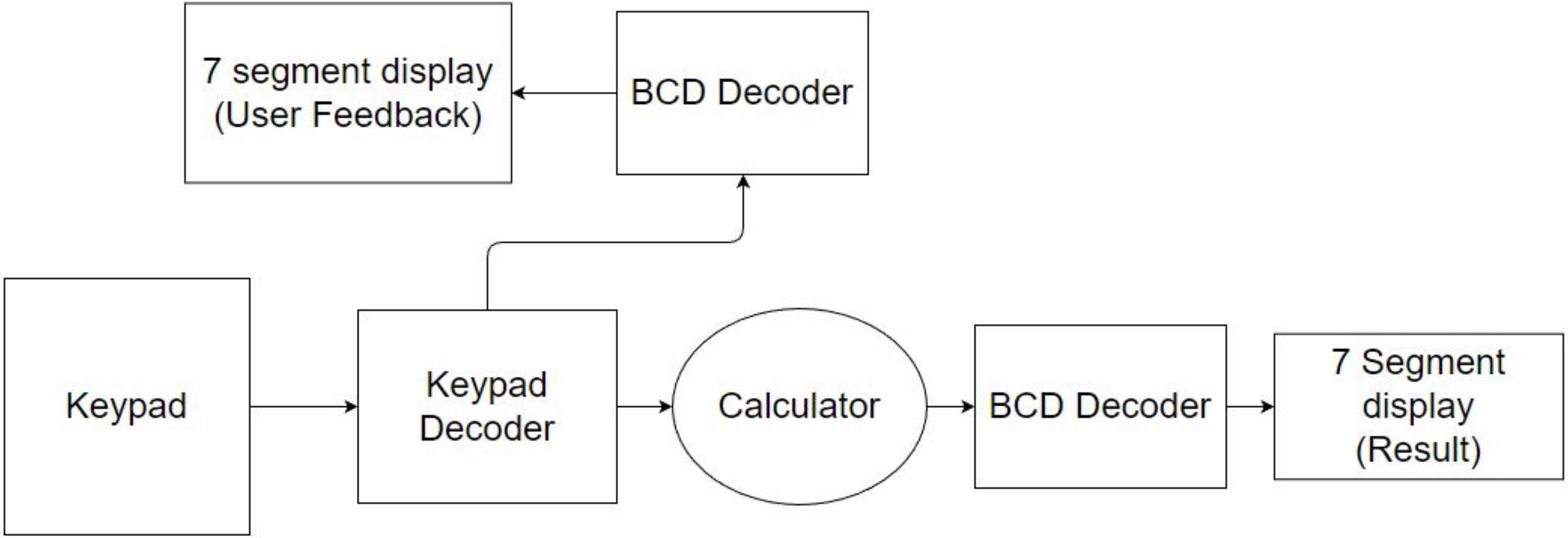
Calculator

Team Pi-Sense

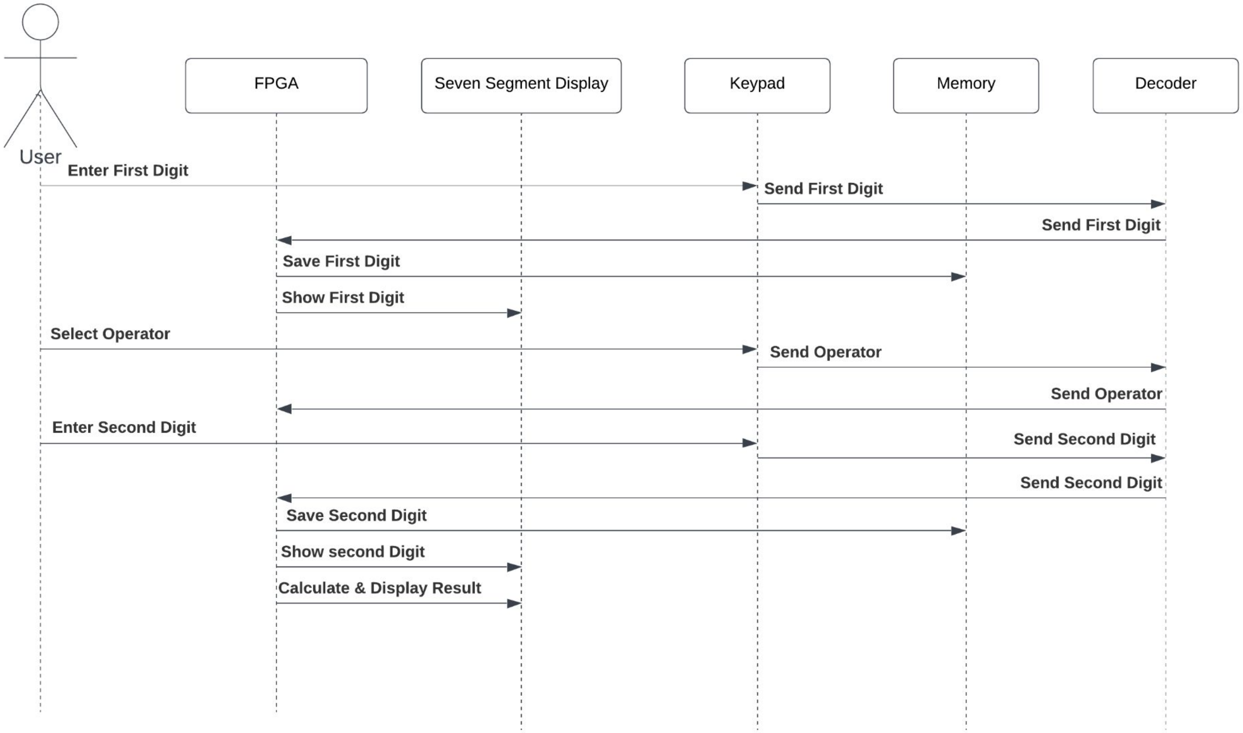
Lochana Abhayawardana, Arsany Girgis and Emirkan Sali

Introduction

Block Diagram

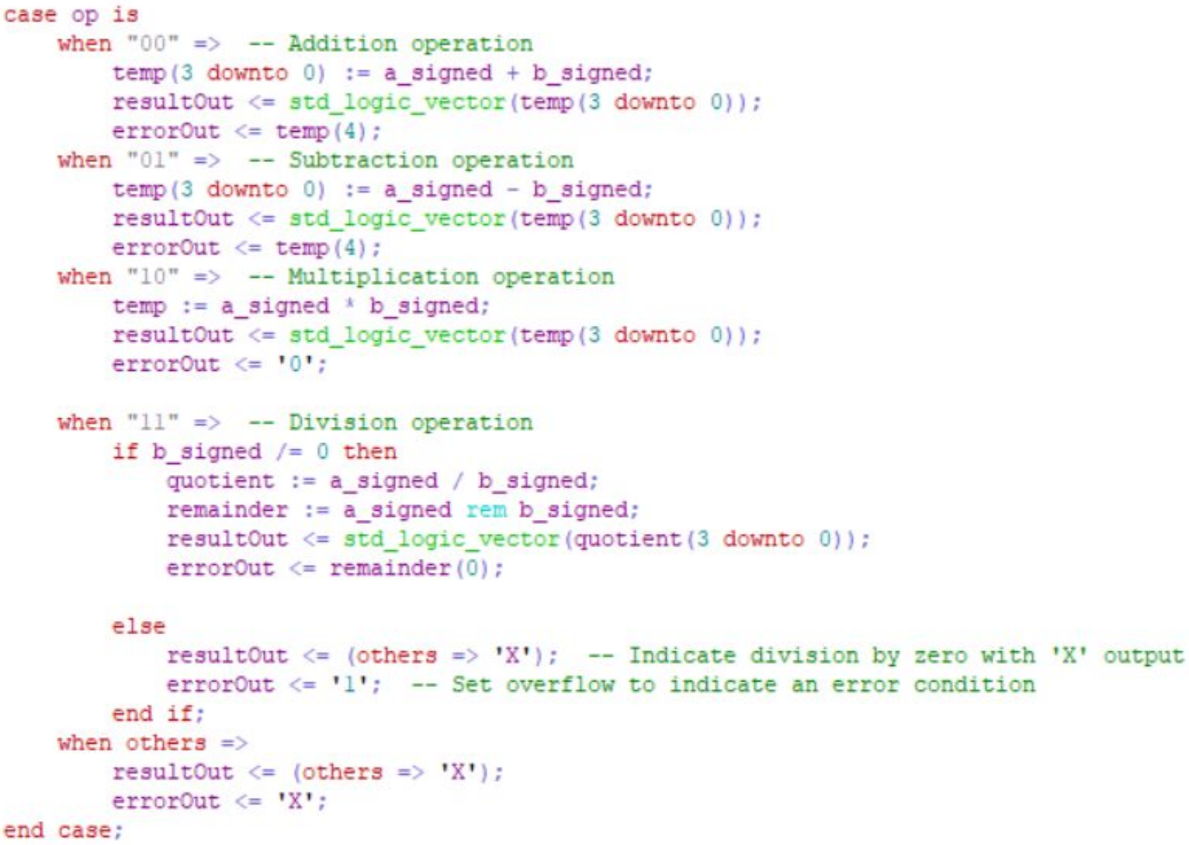


Sequence Diagram



Implementation

Calculator

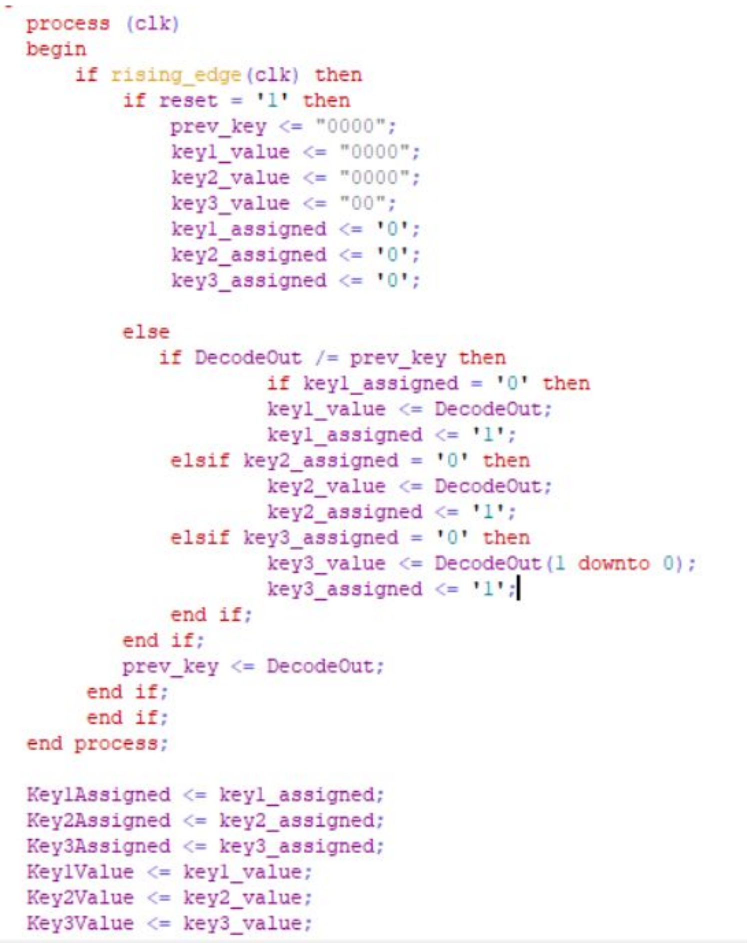


```
case op is
    when "00" =>  -- Addition operation
        temp(3 downto 0) := a_signed + b_signed;
        resultOut <= std_logic_vector(temp(3 downto 0));
        errorOut <= temp(4);
    when "01" =>  -- Subtraction operation
        temp(3 downto 0) := a_signed - b_signed;
        resultOut <= std_logic_vector(temp(3 downto 0));
        errorOut <= temp(4);
    when "10" =>  -- Multiplication operation
        temp := a_signed * b_signed;
        resultOut <= std_logic_vector(temp(3 downto 0));
        errorOut <= '0';

    when "11" =>  -- Division operation
        if b_signed /= 0 then
            quotient := a_signed / b_signed;
            remainder := a_signed rem b_signed;
            resultOut <= std_logic_vector(quotient(3 downto 0));
            errorOut <= remainder(0);

        else
            resultOut <= (others => 'X');  -- Indicate division by zero with 'X' output
            errorOut <= '1';  -- Set overflow to indicate an error condition
        end if;
    when others =>
        resultOut <= (others => 'X');
        errorOut <= 'X';
end case;
```

Decoder FSM

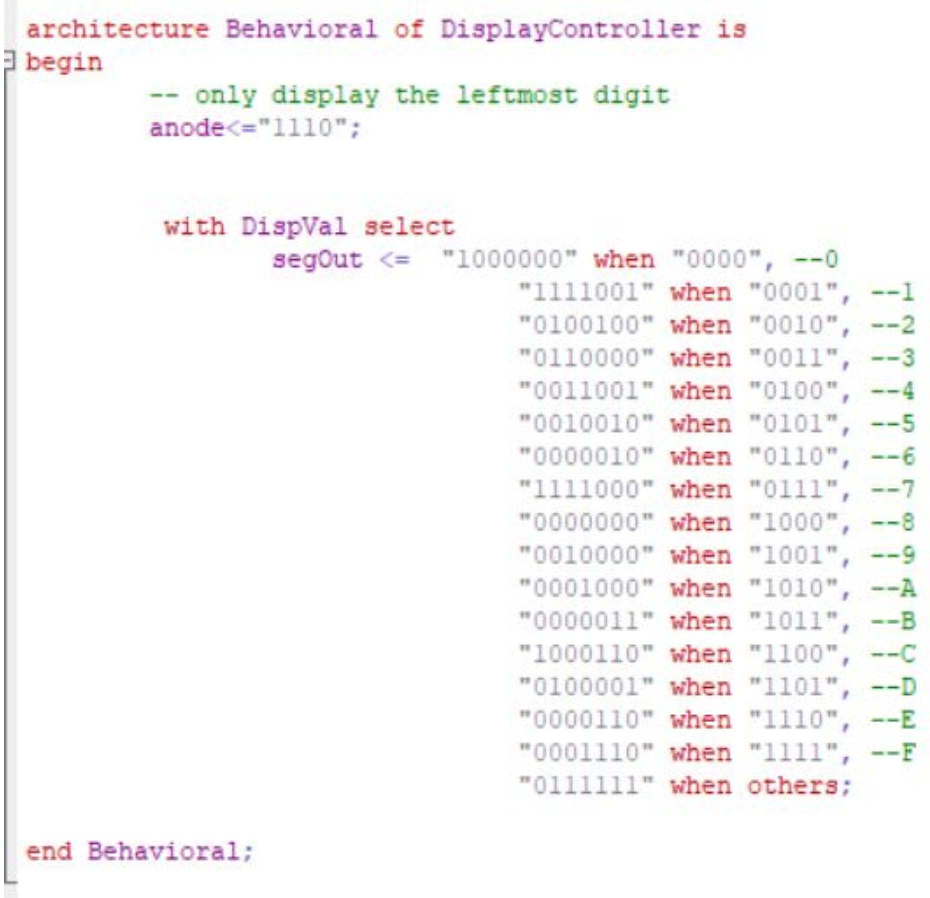


```
process (clk)
begin
    if rising_edge(clk) then
        if reset = '1' then
            prev_key <= "0000";
            key1_value <= "0000";
            key2_value <= "0000";
            key3_value <= "00";
            key1_assigned <= '0';
            key2_assigned <= '0';
            key3_assigned <= '0';

        else
           if DecodeOut /= prev_key then
                    if key1_assigned = '0' then
                    key1_value <= DecodeOut;
                    key1_assigned <= '1';
            elsif key2_assigned = '0' then
                    key2_value <= DecodeOut;
                    key2_assigned <= '1';
            elsif key3_assigned = '0' then
                    key3_value <= DecodeOut(1 downto 0);
                    key3_assigned <= '1';
            end if;
        end if;
        prev_key <= DecodeOut;
     end if;
     end if;
end process;

Key1Assigned <= key1_assigned;
Key2Assigned <= key2_assigned;
Key3Assigned <= key3_assigned;
Key1Value <= key1_value;
Key2Value <= key2_value;
Key3Value <= key3_value;
```

Display

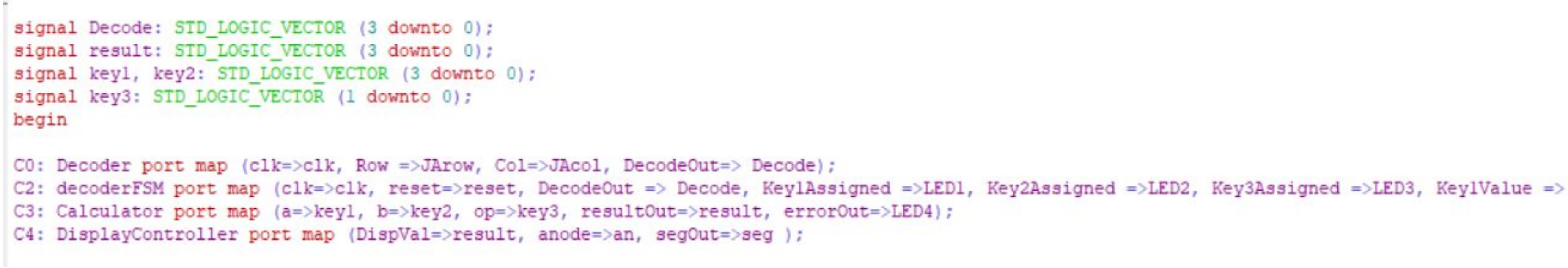


```
architecture Behavioral of DisplayController is
begin
        -- only display the leftmost digit
        anode<="1110";


         with DispVal select
                segOut <=  "1000000" when "0000", --0
                                "1111001" when "0001", --1
                                "0100100" when "0010", --2
                                "0110000" when "0011", --3
                                "0011001" when "0100", --4
                                "0010010" when "0101", --5
                                "0000010" when "0110", --6
                                "1111000" when "0111", --7
                                "0000000" when "1000", --8
                                "0010000" when "1001", --9
                                "0001000" when "1010", --A
                                "0000011" when "1011", --B
                                "1000110" when "1100", --C
                                "0100001" when "1101", --D
                                "0000110" when "1110", --E
                                "0001110" when "1111", --F
                                "0111111" when others;

end Behavioral;
```

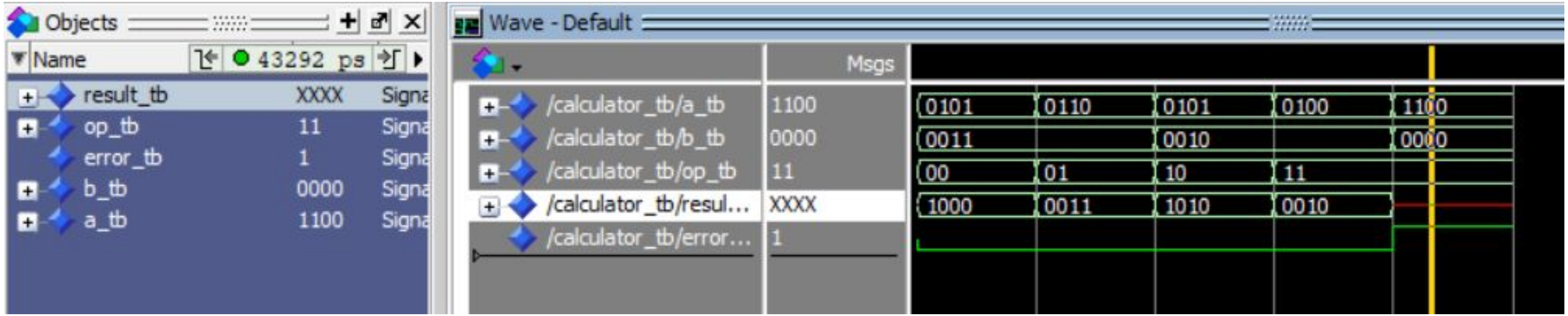
Top module



```
signal Decode: STD_LOGIC_VECTOR (3 downto 0);
signal result: STD_LOGIC_VECTOR (3 downto 0);
signal key1, key2: STD_LOGIC_VECTOR (3 downto 0);
signal key3: STD_LOGIC_VECTOR (1 downto 0);
begin

C0: Decoder port map (clk=>clk, Row =>JArow, Col=>JAcol, DecodeOut=> Decode);
C2: decoderFSM port map (clk=>clk, reset=>reset, DecodeOut => Decode, Key1Assigned =>LED1, Key2Assigned =>LED2, Key3Assigned =>LED3, Key1Value =>
C3: Calculator port map (a=>key1, b=>key2, op=>key3, resultOut=>result, errorOut=>LED4);
C4: DisplayController port map (DispVal=>result, anode=>an, segOut=>seg );
```

Calculator testbench



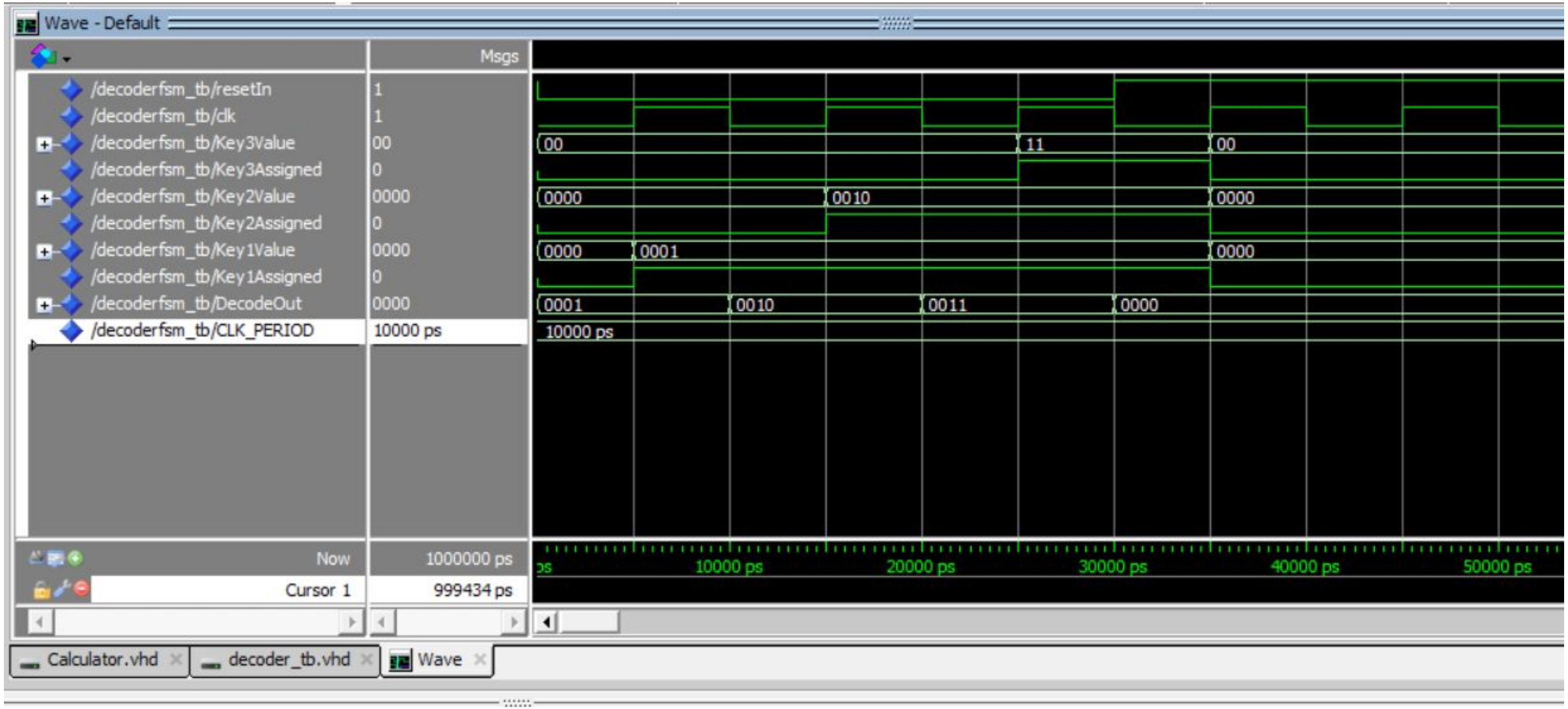
Objects

Name	43292 ps	
result_tb	XXXX	Signa
op_tb	11	Signa
error_tb	1	Signa
b_tb	0000	Signa
a_tb	1100	Signa

Wave - Default

	Msgs
/calculator_tb/a_tb	1100
/calculator_tb/b_tb	0000
/calculator_tb/op_tb	11
/calculator_tb/resul...	XXXX
/calculator_tb/error...	1

Decode FSM testbench



Wave - Default

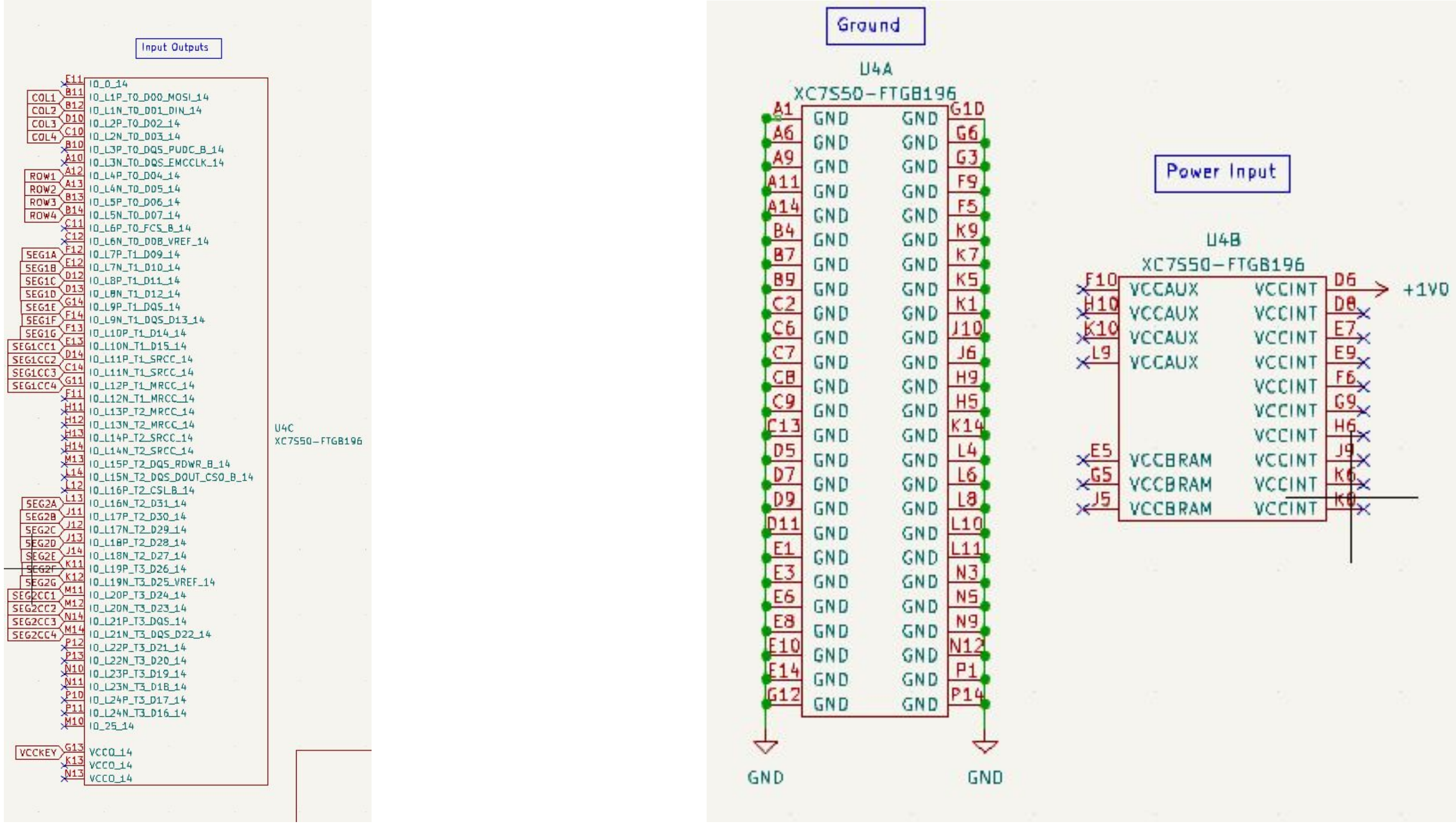
	Msgs
/decoderfsm_tb/resetIn	1
/decoderfsm_tb/clk	1
/decoderfsm_tb/Key3Value	00
/decoderfsm_tb/Key3Assigned	0
/decoderfsm_tb/Key2Value	0000
/decoderfsm_tb/Key2Assigned	0
/decoderfsm_tb/Key1Value	0000
/decoderfsm_tb/Key1Assigned	0
/decoderfsm_tb/DecodeOut	0000
/decoderfsm_tb/CLK_PERIOD	10000 ps

Now	1000000 ps
Cursor 1	999434 ps

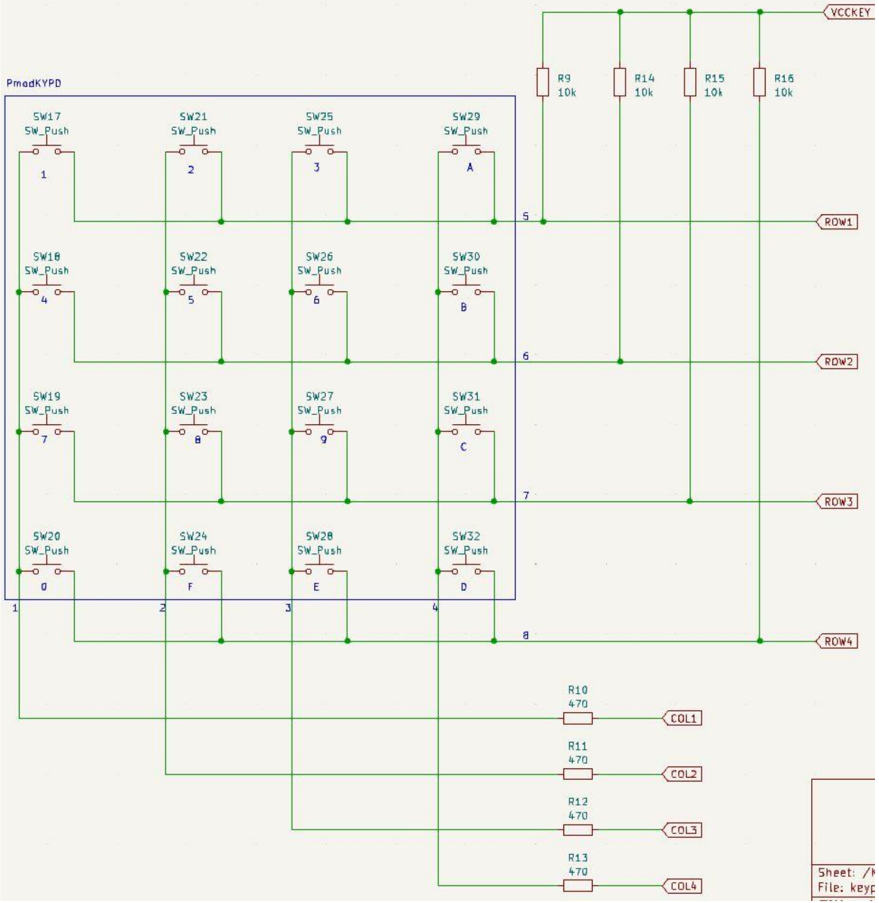
Schematic

Root page consisting of 5 subpages

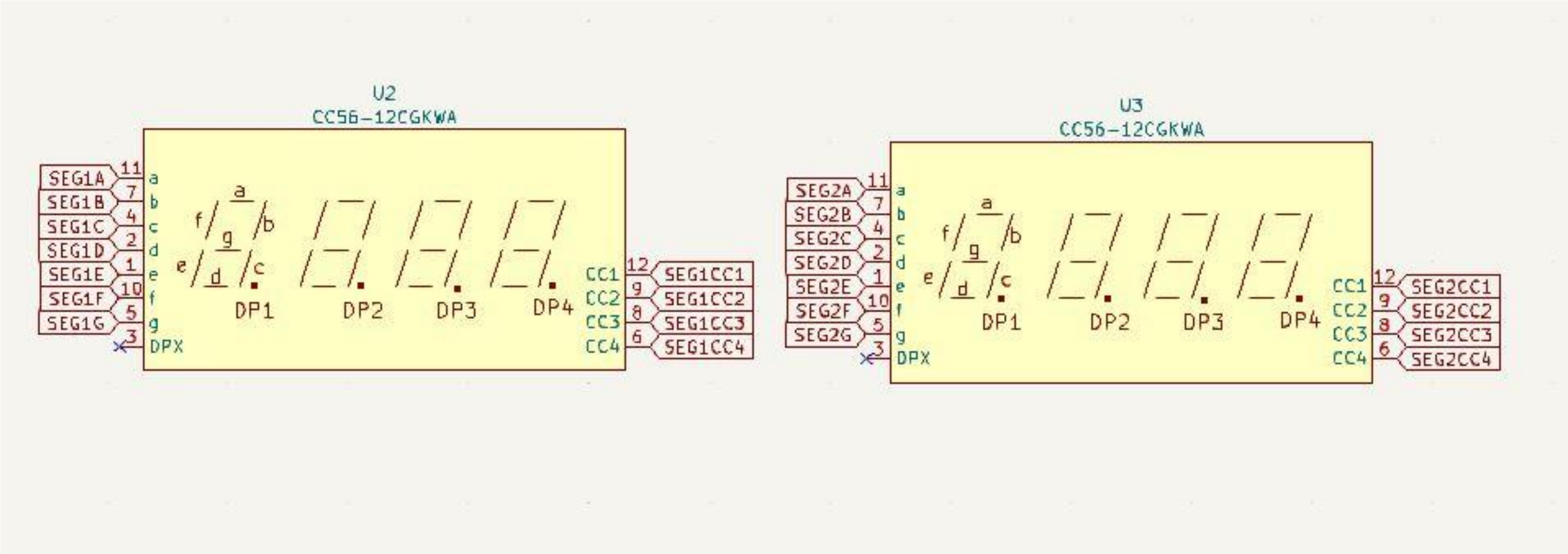
1. Power page with Power supply, switch and LED
2. Keypad Page where we simulated the keypad using pushbuttons
3. FPGA page with the needed components of the FPGA
4. Seven segment display page with both displays
5. FPGA JTAG page for the JTAG Interface


Input Outputs
E11 IO_0_14
COL1 B11 IO_L1P_T0_D00_MOSI_14
COL2 B12 IO_L1N_T0_D01_DIN_14
COL3 D10 IO_L2P_T0_D02_14
COL4 C10 IO_L2N_T0_D03_14
B10 IO_L3P_T0_DQS_PUDC_B_14
A10 IO_L3N_T0_DQS_EMCCLK_14
ROW1 A12 IO_L4P_T0_D04_14
ROW2 A13 IO_L4N_T0_D05_14
ROW3 B13 IO_L5P_T0_D06_14
ROW4 B14 IO_L5N_T0_D07_14
C11 IO_L6P_T0_FCS_B_14
C12 IO_L6N_T0_D08_VREF_14
SEG1A F12 IO_L7P_T1_D09_14
SEG1B E12 IO_L7N_T1_D10_14
SEG1C D12 IO_L8P_T1_D11_14
SEG1D D13 IO_L8N_T1_D12_14
SEG1E G14 IO_L9P_T1_DQS_14
SEG1F F14 IO_L9N_T1_DQS_D13_14
SEG1G F13 IO_L10P_T1_D14_14
SEG1CC1 E13 IO_L10N_T1_D15_14
SEG1CC2 D14 IO_L11P_T1_SRCC_14
SEG1CC3 C14 IO_L11N_T1_SRCC_14
SEG1CC4 G11 IO_L12P_T1_MRCC_14
F11 IO_L12N_T1_MRCC_14
H11 IO_L13P_T2_MRCC_14
H12 IO_L13N_T2_MRCC_14
H13 IO_L14P_T2_SRCC_14
H14 IO_L14N_T2_SRCC_14
M13 IO_L15P_T2_DQS_RDWR_B_14
L14 IO_L15N_T2_DQS_DOUT_CSO_B_14
L12 IO_L16P_T2_CSI_B_14
SEG2A L13 IO_L16N_T2_D31_14
SEG2B J11 IO_L17P_T2_D30_14
SEG2C J12 IO_L17N_T2_D29_14
SEG2D J13 IO_L18P_T2_D28_14
SEG2E J14 IO_L18N_T2_D27_14
SEG2F K11 IO_L19P_T3_D26_14
SEG2G K12 IO_L19N_T3_D25_VREF_14
SEG2CC1 M11 IO_L20P_T3_D24_14
SEG2CC2 M12 IO_L20N_T3_D23_14
SEG2CC3 N14 IO_L21P_T3_DQS_14
SEG2CC4 M14 IO_L21N_T3_DQS_D22_14
P12 IO_L22P_T3_D21_14
P13 IO_L22N_T3_D20_14
N10 IO_L23P_T3_D19_14
N11 IO_L23N_T3_D18_14
P10 IO_L24P_T3_D17_14
P11 IO_L24N_T3_D16_14
M10 IO_25_14
VCCKEY G13 VCCO_14
K13 VCCO_14
N13 VCCO_14
U4C
XC7S50-FTGB196
Ground
U4A
XC7S50-FTGB196
A1 GND GND G10
A6 GND GND G6
A9 GND GND G3
A11 GND GND F9
A14 GND GND F5
B4 GND GND K9
B7 GND GND K7
B9 GND GND K5
C2 GND GND K1
C6 GND GND J10
C7 GND GND J6
C8 GND GND H9
C9 GND GND H5
C13 GND GND K14
D5 GND GND L4
D7 GND GND L6
D9 GND GND L8
D11 GND GND L10
E1 GND GND L11
E3 GND GND N3
E6 GND GND N5
E8 GND GND N9
E10 GND GND N12
E14 GND GND P1
G12 GND GND P14
GND
GND
Power Input
U4B
XC7S50-FTGB196
F10 VCCAUX VCCINT D6
+1V0
H10 VCCAUX VCCINT D8
K10 VCCAUX VCCINT E7
L9 VCCAUX VCCINT E9
VCCINT F6
VCCINT G9
VCCINT H6
E5 VCCBRAM VCCINT J9
G5 VCCBRAM VCCINT K6
J5 VCCBRAM VCCINT K8

FPGA Components in the schematic



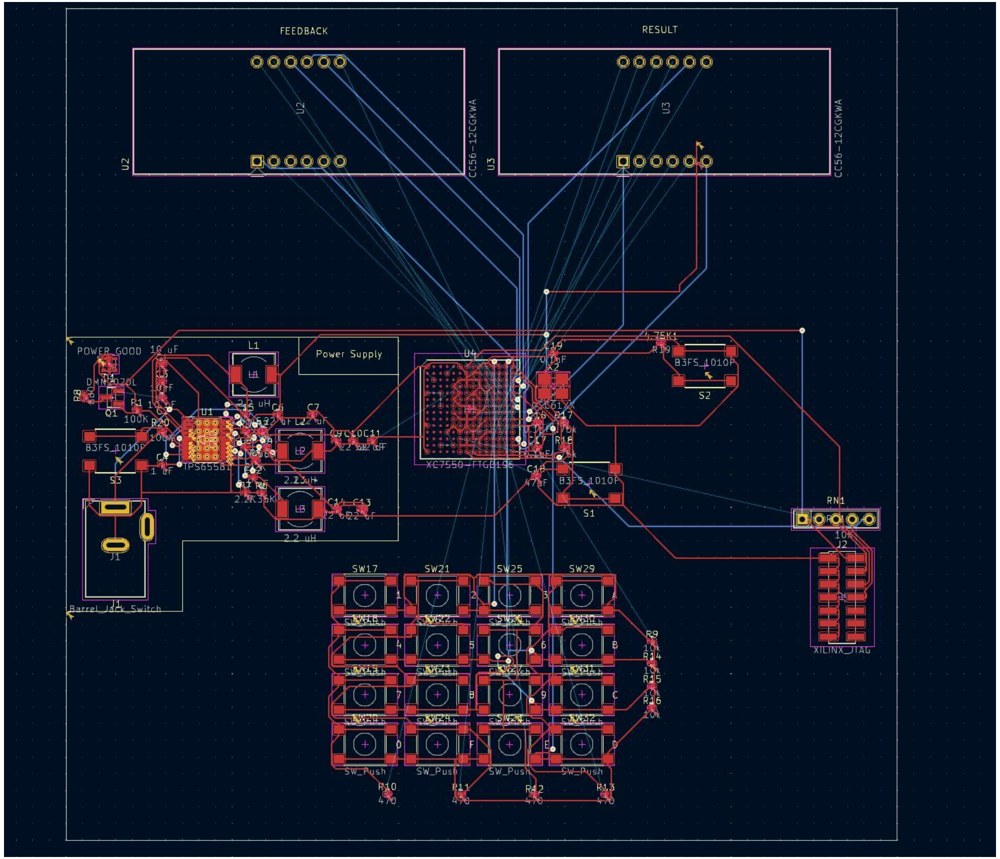
Keypad using push-buttons



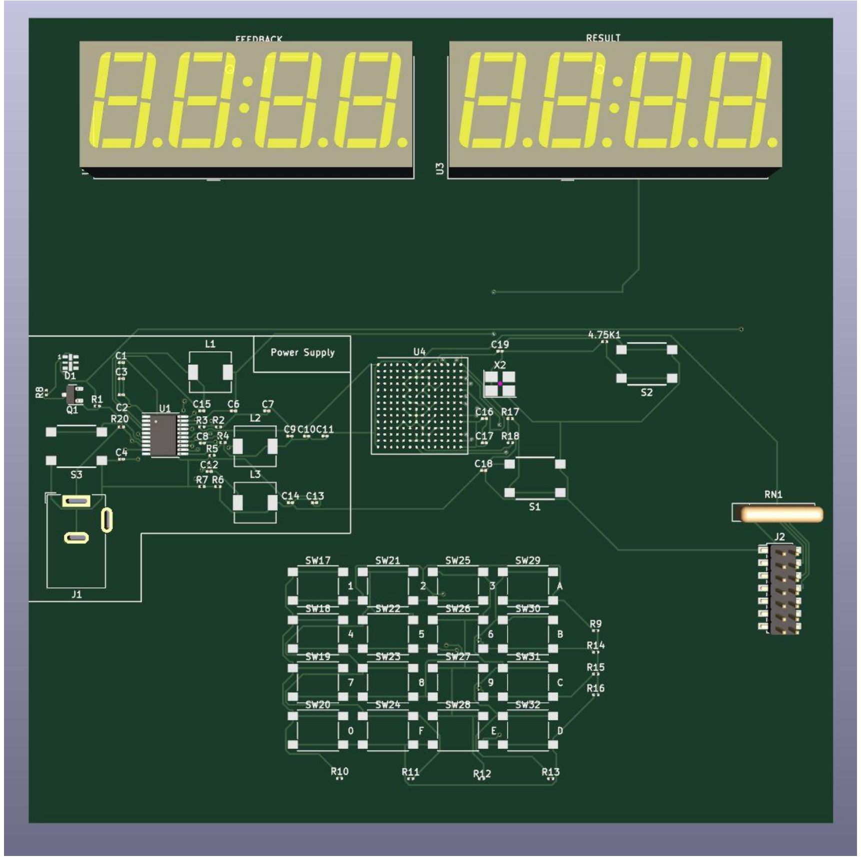
Seven Segment displays connected to the I/O ports of FPGA

PCB Design

- Selecting the right footprints for the symbols
- Arranging the components in a way that makes sense
- Labeling, designing etc.
- Routing of connections using Autorouter



PCB layout with all its layers visible



3D View of the PCB