طراحی سیستم های دیجیتال
پروژه نهایی
مهندسی کامپیوتر، دانشگاه شهید بهشتی

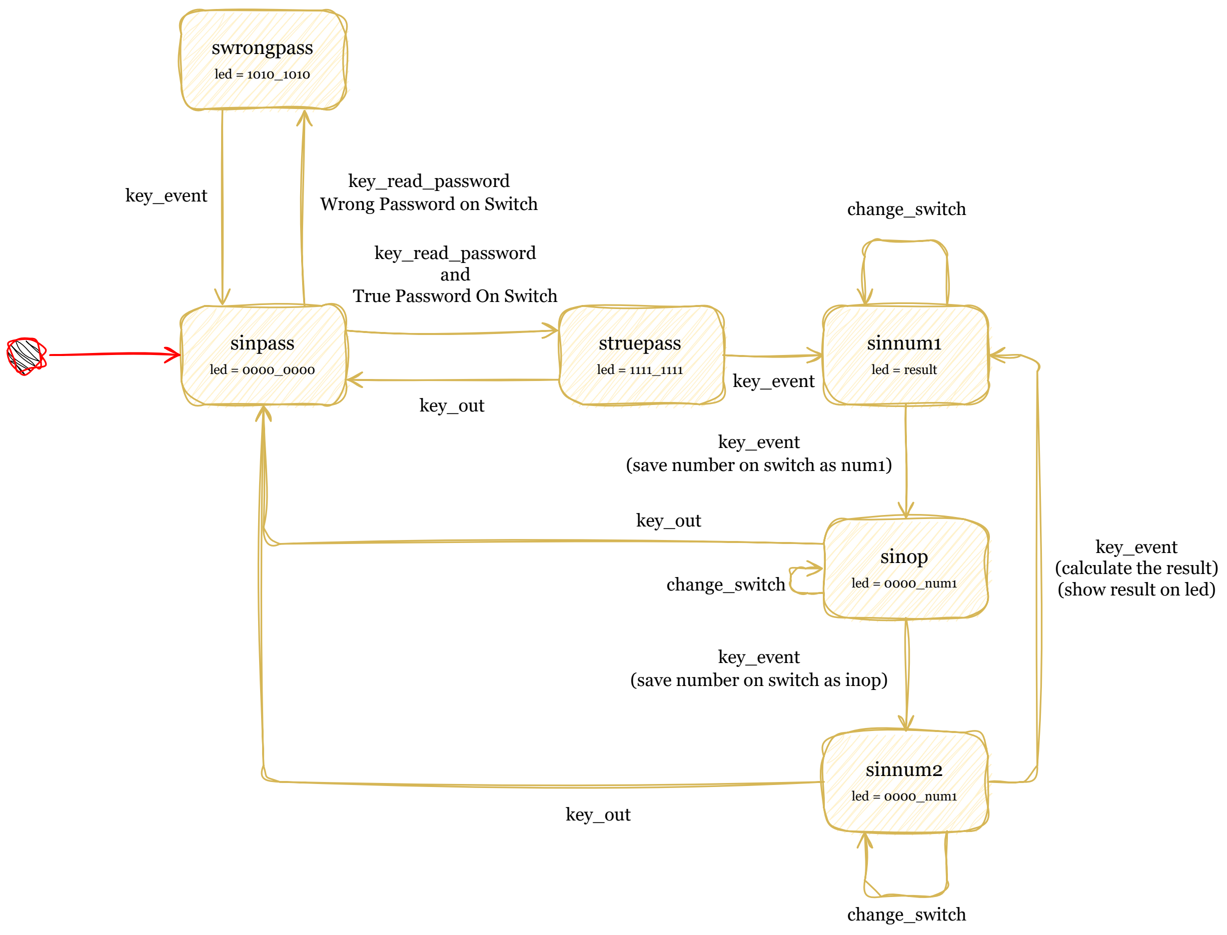
عباس یزدان مهر
۹۹۲۴۳۰۷۷

پدرام رمضان زاده
۹۹۲۴۳۰۸۵

۳۰ دی ۱۴۰۱

۱ ماشین حساب

۱.۱ ماشین حالات



شکل ۱: ماشین حالات

۲.۱ کد

```vhdl
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
USE ieee.numeric_std.ALL;

-- top module(calculator)
ENTITY calcul IS
   PORT (
      clk : IN STD_LOGIC;
      KEY : IN STD_LOGIC_VECTOR(1 DOWNTO 0);
      SW : IN STD_LOGIC_VECTOR(3 DOWNTO 0);
      LED : OUT STD_LOGIC_VECTOR(7 DOWNTO 0)
   );
END calcul;

ARCHITECTURE beh OF calcul IS

   -- operand and operators and result
   SIGNAL num1 : STD_LOGIC_VECTOR(3 DOWNTO 0) := "0000";
   SIGNAL inop : STD_LOGIC_VECTOR(3 DOWNTO 0) := "1111";
   SIGNAL num2 : STD_LOGIC_VECTOR(3 DOWNTO 0) := "0000";
   SIGNAL result : STD_LOGIC_VECTOR(7 DOWNTO 0) := "11110000";
```

```
    -- operator
    SIGNAL op_sum : STD_LOGIC_VECTOR(3 DOWNTO 0) := "0000";
    SIGNAL op_minus : STD_LOGIC_VECTOR(3 DOWNTO 0) := "0001";
    SIGNAL op_product : STD_LOGIC_VECTOR(3 DOWNTO 0) := "0011";
    SIGNAL op_division : STD_LOGIC_VECTOR(3 DOWNTO 0) := "0111";

    -- bad operator
    SIGNAL bad_operator_result : STD_LOGIC_VECTOR(7 DOWNTO 0) := "11000011";

    -- state definition
    TYPE state_type IS (sinpass, swrongpass, struepass, sinnum1, sinop, sinnum2);
    SIGNAL state : state_type := sinpass;

    -- password
    SIGNAL stored_pass : STD_LOGIC_VECTOR(3 DOWNTO 0) := "0101";

    -- KEY meanings:
    SIGNAL key_read_password : STD_LOGIC_VECTOR(1 DOWNTO 0) := "01"; -- press key 1
    SIGNAL key_event : STD_LOGIC_VECTOR(1 DOWNTO 0) := "10"; -- press key 0
    SIGNAL key_out : STD_LOGIC_VECTOR(1 DOWNTO 0) := "00"; -- press key both button

    -- LED meanings:
    SIGNAL led_sinpass : STD_LOGIC_VECTOR(7 DOWNTO 0) := "00000000";
    SIGNAL led_swrongpass : STD_LOGIC_VECTOR(7 DOWNTO 0) := "10101010";
    SIGNAL led_struepass : STD_LOGIC_VECTOR(7 DOWNTO 0) := "11111111";

    -- calculates
    SIGNAL adder_result : STD_LOGIC_VECTOR(7 DOWNTO 0);
    SIGNAL carry_out : STD_LOGIC;

BEGIN

    key_pressed : PROCESS (KEY)
    BEGIN
        IF state = sinpass AND KEY = key_read_password THEN
            IF SW = stored_pass THEN
                state <= struepass;
            ELSE
                state <= swrongpass;
            END IF;
        ELSIF state = swrongpass AND KEY = key_event THEN
            state <= sinpass;
        ELSIF state = struepass THEN
            IF KEY = key_out THEN
                state <= sinpass;
            ELSIF KEY = key_event THEN
                state <= sinnum1;
            END IF;
        ELSIF state = sinnum1 THEN
            IF KEY = key_out THEN
                state <= sinpass;
            ELSIF KEY = key_event THEN
                num1 <= SW;
                state <= sinop;
            END IF;
        ELSIF state = sinop THEN
            IF KEY = key_out THEN
                state <= sinpass;
            ELSIF KEY = key_event THEN
                inop <= SW;
                state <= sinnum2;
            END IF;
        ELSIF state = sinnum2 THEN
            IF KEY = key_out THEN
                state <= sinpass;
```

```vhdl
            ELSIF KEY = key_event THEN
                num2 <= SW;
                state <= sinnum1;
            END IF;
        END IF;
    END PROCESS;

    -- led value based on state
    led_initial : PROCESS (state)
    BEGIN
        CASE state IS
            WHEN sinpass =>
                LED <= led_sinpass;
            WHEN swrongpass =>
                LED <= led_swrongpass;
            WHEN struepass =>
                LED <= led_struepass;
            WHEN sinnum1 =>
                -- calculate
                IF inop = op_sum THEN
                    LED <= STD_LOGIC_VECTOR(unsigned("0000" & num1) + unsigned("0000" & num2));
                ELSIF inop = op_minus THEN
                    LED <= STD_LOGIC_VECTOR(unsigned("0000" & num1) - unsigned("0000" & num2));
                ELSIF inop = op_product THEN
                    LED <= STD_LOGIC_VECTOR(unsigned(num1) * unsigned(num2));
                ELSIF inop = op_division THEN
                    LED <= STD_LOGIC_VECTOR(unsigned("0000" & num1) / unsigned("0000" & num2));
                ELSE
                    LED <= bad_operator_result;
                END IF;
            WHEN sinop =>
                LED <= "0000" & num1;
            WHEN sinnum2 =>
                LED <= "0000" & num1;
        END CASE;
    END PROCESS; -- led_initial

END beh; -- beh
```

Listing 1: code/calcul.vhd

۳.۱ شبیه سازی

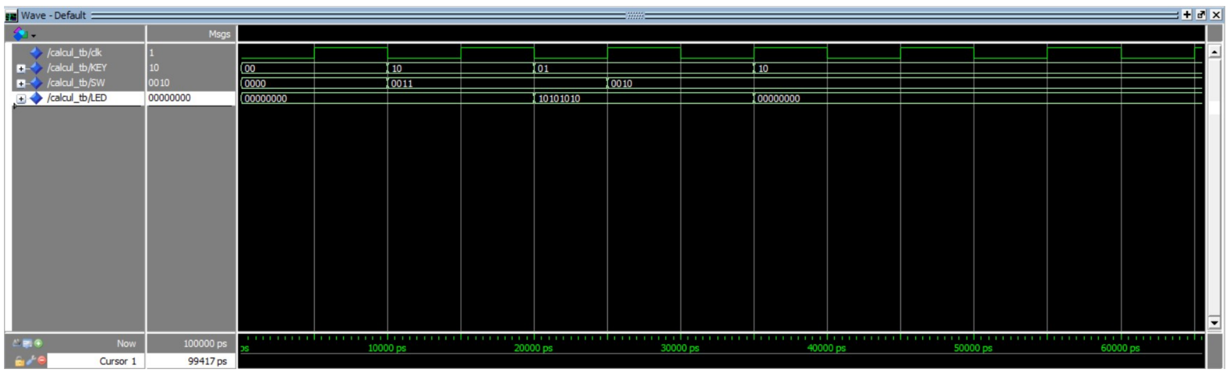
۱.۳.۱ کد تست

```vhdl
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;

ENTITY calcul_tb IS
END calcul_tb;

ARCHITECTURE beh OF calcul_tb IS

    SIGNAL clk : STD_LOGIC := '0';
    SIGNAL KEY : STD_LOGIC_VECTOR(1 DOWNTO 0);
    SIGNAL SW : STD_LOGIC_VECTOR(3 DOWNTO 0);
    SIGNAL LED : STD_LOGIC_VECTOR(7 DOWNTO 0);

BEGIN

    uut : ENTITY work.calcul PORT MAP(clk => clk, KEY => KEY, SW => SW, LED => LED);

    -- clock (no usage)
```

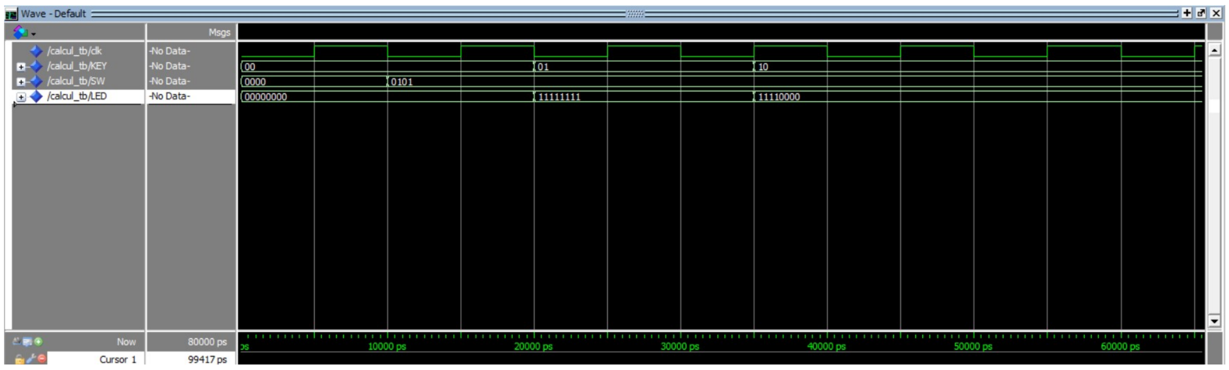
```
   -- clk <= NOT clk AFTER 5 ns;

   -- sim1 - wrong password input
   -- SW <= "0000", "0011" AFTER 10 ns, "0010" AFTER 25 ns;
   -- KEY <= "00", "10" AFTER 10 ns, "01" AFTER 20 ns, "10" AFTER 35 ns;

   -- sim2 - true password input
   -- SW <= "0000", "0101" AFTER 10 ns;
   -- KEY <= "00", "01" AFTER 20 ns, "10" AFTER 35 ns;

   -- sim3 - one complete senario
   tb1 : PROCESS
      CONSTANT select_period : TIME := 5 ns;
      CONSTANT press_period : TIME := 1 ns;
   BEGIN
      SW <= "0000";
      KEY <= "00";
      WAIT FOR select_period;

      SW <= "0101"; -- true password on switch
      WAIT FOR select_period;

      KEY <= "01"; -- key_read_password
      WAIT FOR press_period;
      KEY <= "11";
      WAIT FOR select_period;

      KEY <= "10"; -- key_event go to sinnum1 state
      WAIT FOR press_period;
      KEY <= "11";
      WAIT FOR select_period;

      SW <= "1111"; -- number 31 on switch as num1
      WAIT FOR select_period;

      KEY <= "10"; -- num1 = 31 now state is sinop
      WAIT FOR press_period;
      KEY <= "11";
      WAIT FOR select_period;

      SW <= "0000"; -- choose addition operation --------------- choose operator here --
      WAIT FOR select_period;

      KEY <= "10"; -- op = addition now state is sinnum2
      WAIT FOR press_period;
      KEY <= "11";
      WAIT FOR select_period;

      SW <= "0111"; -- number 7 on switch as num2
      WAIT FOR select_period;

      KEY <= "10"; -- num2 = 7 now state is sinnum1
      WAIT FOR press_period;
      KEY <= "11";
      WAIT FOR select_period;

      -- you should see the led = 7 + 15 = 22 = 0001_0110 here

      WAIT; -- indefinitely suspend process
   END PROCESS;

END beh; -- beh
```

Listing 2: code/calcultb.vhd

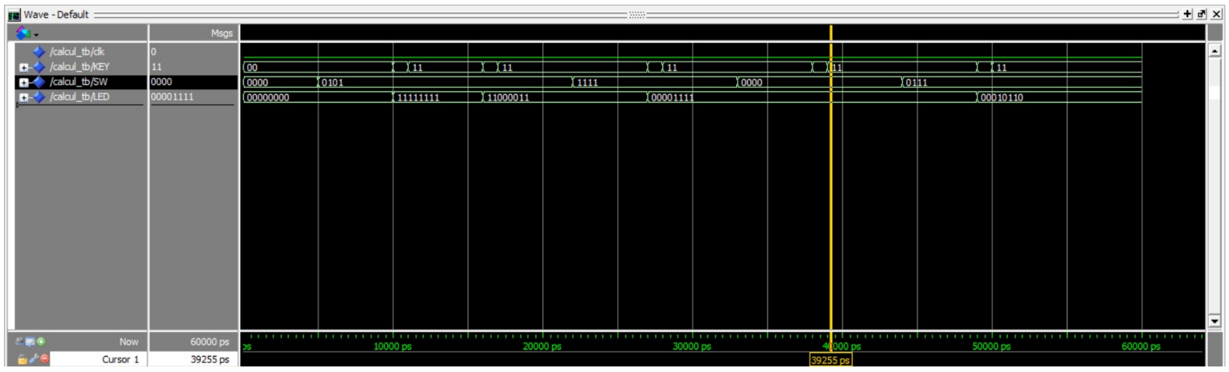
۲.۳.۱ خروجی تست



شکل ۲: شبیه سازی - واردکردن رمز نادرست



شکل ۳: شبیه سازی - واردکردن رمز درست



شکل ۴: شبیه سازی - یک روند از انجام کامل عملیات

پایان