

به نام خدا

گزارش کار آزمایشگاه معماری کامپیوتر

آزمایش جلسه دهم

عنوان آزمایش:

LED Light Control With Counter

نام استاد:

استاد علی جوادی

اعضای گروه:

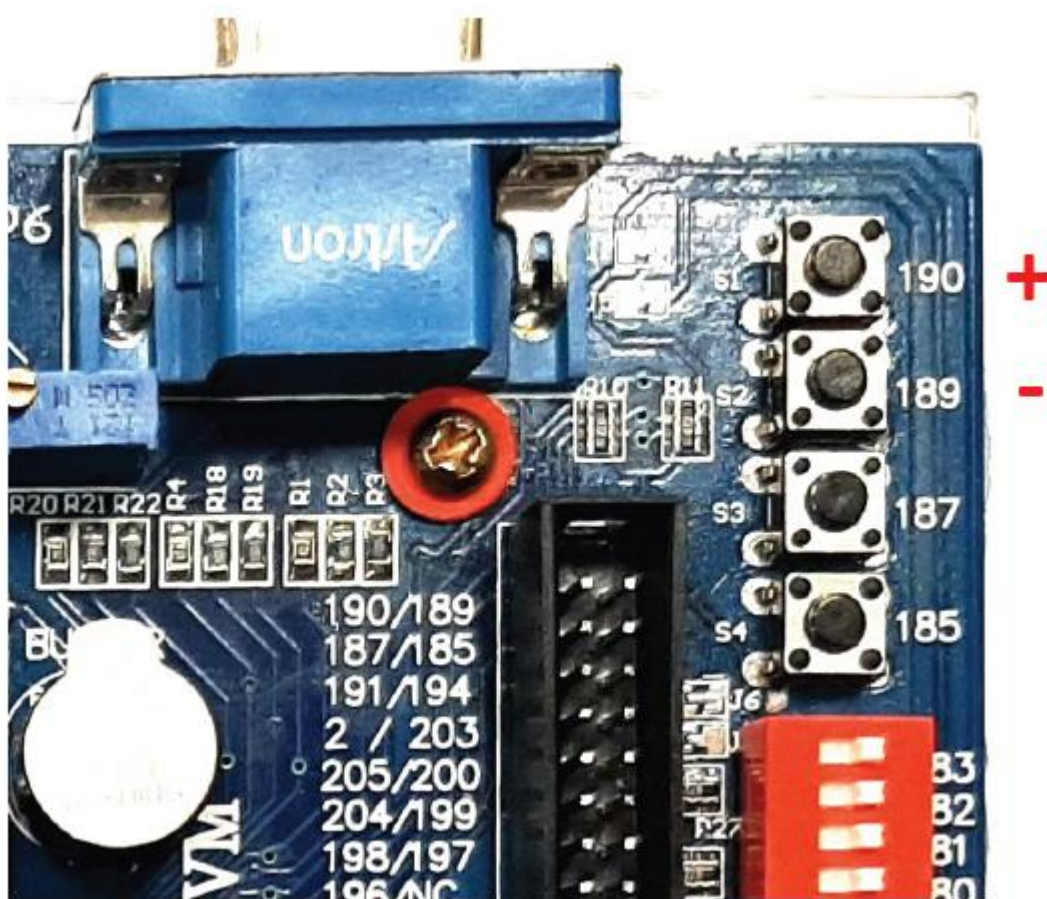
غزل عربعلی - بهاره کاوسی نژاد

## آزمایش: LED Light Control With Counter

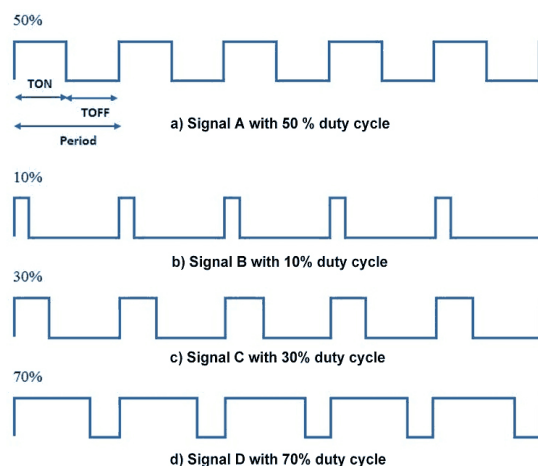
هدف آزمایش: کنترل نور LED با فشردن دکمه و نمایش مقدار نور با 7-Segment

تئوری آزمایش:

در این آزمایش می‌خواهیم نور یک LED را با فشردن یک دکمه کم و با فشردن دکمه دیگر زیاد کنیم. همچنین عدد نسبت داده شده به نور LED را به صورت 7 Segment نمایش دهیم.



برای تنظیم نور LED، مقدار duty cycle را تغییر می‌دهیم تا زمان روشن بودن LED کم و زیاد شود و این تغییرات توسط چشم انسان به شکل کم نور و پر نور شدن LED دیده می‌شوند.



روش و چگونگی انجام آزمایش:

در ابتدا ورودی‌ها و خروجی را مشخص می‌کنیم:

- ورودی: GCLK
- ButtonSwitchAdd و ButtonSwitchSub: ورودی
- SevenSegment: خروجی
- SEG\_SEL: خروجی
- LED: خروجی
- OUTPUT1, OUTPUT2, OUTPUT3 و OUTPUT4: خروجی

```
entity LEDControlButton is
  Port ( GCLK : in  STD_LOGIC;
        ButtonSwitchAdd : in  STD_LOGIC;
        ButtonSwitchSub : in  STD_LOGIC;
        SevenSegment : out  STD_LOGIC_VECTOR (7 downto 0);
        SEG_SEL : out  STD_LOGIC_VECTOR (4 downto 0);
        LED : out  STD_LOGIC;
        OUTPUT1 : out  STD_LOGIC_VECTOR (3 downto 0);
        OUTPUT2 : out  STD_LOGIC_VECTOR (3 downto 0);
        OUTPUT3 : out  STD_LOGIC_VECTOR (3 downto 0);
        OUTPUT4 : out  STD_LOGIC_VECTOR (3 downto 0);
  );
end LEDControlButton;
```

چند سیگنال تعریف می‌کنیم:

- CLOCK10MS : کلاک 10 میلی ثانیه
- SEGDATA1 تا 4
- SEGDATA1REG1 تا 4
- brightness

```
architecture Behavioral of LEDControlButton is
shared variable STARTSTOPTOGGLE : STD_LOGIC ;
signal CLOCK10MS : STD_LOGIC;
signal SEGDATA1 , SEGDATA2,SEGDATA3 , SEGDATA4 : integer range 0 to 10 := 0 ;
signal SEGDATA1REG1 , SEGDATA1REG2 ,SEGDATA1REG3,SEGDATA1REG4 , SEGDATA1REG5 : STD_LOGIC_VECTOR( 7 downto 0 ) ;
--signal second_process_clock : std_logic := '0';
signal brightness : integer range 0 to 9999 := 0;

begin
```

دریافت دیتا و تنظیم نور LED:

```
process (GCLK)
variable counter_clock : integer range 0 to 88 := 0;-- half a clock
variable duty_cycle : integer range 0 to 255 := 0;
variable brightness_temp : integer range 0 to 255 := 0;
begin
    if (rising_edge(GCLK)) then
        -- if (RX = '0') then -- should start reading 8 digit
        --     is_started <= '1';
        -- end if;
        --if (duty_cycle = "11111111") then -- each 255 times
        brightness_temp := to_integer(brightness/40);
        if (duty_cycle < 254 ) then
            duty_cycle := 0;
        elsif ( duty_cycle < brightness_temp) then
            LED <= '1';
            duty_cycle := duty_cycle + 1;
        else
            LED <= '0';
            duty_cycle := duty_cycle + 1;
        end if;
        if (counter_clock < 87) then -- counter_clock < 88
            counter_clock := counter_clock + 1;
        else
            counter_clock := 0;
            second_process_clock <= not second_process_clock;
        end if;
    end if;
end process;
```

## تنظیم مقدار عدد نمایش داده شده روی SevenSegment

```
process(ButtonSwitchAdd, ButtonSwitchSub,CLOCK10MS)
variable counter : integer range 0 to 9999 := 0;
variable VSEGDAT1 , VSEGDAT2 ,VSEGDAT3 ,VSEGDAT4 := STD_LOGIC_VECTOR(3 downto 0) := "0000";
begin
    if(counter < 9999 and rising_edge(CLOCK10MS)) then
        if( VSEGDAT1 < "1001") then
            VSEGDAT1 := (STD_LOGIC_VECTOR(VSEGDAT1) + 1);
        else
            VSEGDAT1 := "0000";
            if( VSEGDAT2 < "1001" ) then
                VSEGDAT2 := (STD_LOGIC_VECTOR(VSEGDAT2) + 1);
            else
                VSEGDAT2 := "0000";
                if ( VSEGDAT3 < "1001" ) then
                    VSEGDAT3 := (STD_LOGIC_VECTOR(VSEGDAT3) + 1);
                else
                    VSEGDAT3 := "0000";
                    if ( VSEGDAT4 < "1001" ) then
                        VSEGDAT4 := (STD_LOGIC_VECTOR(VSEGDAT4) + 1);
                    else
                        VSEGDAT4 := "0000";
                    end if;
                end if;
            end if;
        end if;
        if (ButtonSwitchAdd = '0') then

        end if;

        if (ButtonSwitchAdd = '0') then

        end if;
    end if;
end if;
end process;
```

تنظیم عملکرد دکمه‌ها:

```
-- second clock set
process(ButtonSwitchAdd, ButtonSwitchSub)
variable temp_data1 : integer range 0 to 9 := 0;
variable temp_data2 : integer range 0 to 9 := 0;
variable temp_data3 : integer range 0 to 9 := 0;
variable temp_data4 : integer range 0 to 9 := 0;
begin

    if (ButtonSwitchAdd = '0') then -- press to increase
        brightness := brightness + 10;
    end if;
    if (ButtonSwitchSub = '0') then -- press to decrease
        brightness := brightness - 10;
    end if;
    temp_data1 := brightness /1;
    SEGDATA1 := to_integer(temp_data1);
    temp_data2 := brightness /10;
    SEGDATA2 := to_integer(temp_data2);
    temp_data3 := brightness /100;
    SEGDATA3 := to_integer(temp_data3);
    temp_data4 := brightness /1000;
    SEGDATA4 := to_integer(temp_data4);

end process;
```

تعریف کلاک 10 میلی ثانیه:

```
----- CLOCK 10MS -----
process (GCLK)
    variable COUNTERDIVISION10MS : integer range 0 to 100000 := 0 ;
begin
    if rising_edge(GCLK) then
        if COUNTERDIVISION10MS < 80000 then
            COUNTERDIVISION10MS := COUNTERDIVISION10MS + 1;
        else
            COUNTERDIVISION10MS := 0;
            CLOCK10MS <= not CLOCK10MS;
        end if;
    end if;
end process;
```

انتقال داده‌ها به منظور نمایش روی SevenSegment:

```
with SEGDATA1 select
    SEGDATA1REG1 <= "00111111" when 0,
                    "00000110" when 1,
                    "01011011" when 2,
                    "01001111" when 3,
                    "01100110" when 4,
                    "01101101" when 5,
                    "01111101" when 6,
                    "00000111" when 7,
                    "01111111" when 8,
                    "01101111" when 9,
                    "00000000" when others ;

with SEGDATA2 select
    SEGDATA1REG2 <= "00111111" when 0,
                    "00000110" when 1,
                    "01011011" when 2,
                    "01001111" when 3,
                    "01100110" when 4,
                    "01101101" when 5,
                    "01111101" when 6,
                    "00000111" when 7,
                    "01111111" when 8,
                    "01101111" when 9,
                    "00000000" when others ;

with SEGDATA3 select
    SEGDATA1REG3 <= "00111111" when 0,
                    "00000110" when 1,
                    "01011011" when 2,
                    "01001111" when 3,
                    "01100110" when 4,
                    "01101101" when 5,
                    "01111101" when 6,
                    "00000111" when 7,
                    "01111111" when 8,
                    "01101111" when 9,
                    "00000000" when others ;

with SEGDATA4 select
    SEGDATA1REG4 <= "00111111" when 0,
                    "00000110" when 1,
                    "01011011" when 2,
                    "01001111" when 3,
                    "01100110" when 4,
                    "01101101" when 5,
                    "01111101" when 6,
                    "00000111" when 7,
                    "01111111" when 8,
                    "01101111" when 9,
                    "00000000" when others ;

OUTPUT1 <= SEGDATA1;
OUTPUT2 <= SEGDATA2;
OUTPUT3 <= SEGDATA3;
OUTPUT4 <= SEGDATA4;
```

هر SevenSegment در هر لحظه به صورت جدا روشن می‌شود. در این قسمت مشخص می‌کنیم که کدامیک در هر لحظه روشن باشند.

```
process(CLOCK10MS)
variable RefreshSEG : integer range 0 to 4 :=0 ;
begin
    if rising_edge(CLOCK10MS) then
        if RefreshSEG < 4 then
            RefreshSEG := RefreshSEG + 1;
        else RefreshSEG :=0;
        end if ;
        case RefreshSEG is
            when 0 =>
                SEG_SEL(4) <='0';
                SEG_SEL(0) <='1';
                SEG_DATA <= SEGDATAREG1;
            when 1 =>
                SEG_SEL(0) <='0';
                SEG_SEL(1) <='1';
                SEG_DATA <= SEGDATAREG2;
            when 2 =>
                SEG_SEL(1) <='0';
                SEG_SEL(2) <='1';
                SEG_DATA <= SEGDATAREG3;
            when 3 =>
                SEG_SEL(2) <='0';
                SEG_SEL(3) <='1';
                SEG_DATA <= SEGDATAREG4;
            when 4 =>
                SEG_SEL(3) <= '0';
                SEG_SEL(4) <= '1';
                SEG_DATA <= SEGDATAREG5;
            when others => null ;
        end case ;
    end if ;
end process;

end Behavioral;
```

بحث و نتیجه گیری:

در انتها با استفاده از دکمه‌ها نور LED را کم و زیاد می‌کنیم و مقدار روشنایی را روی SevenSegment نمایش می‌دهیم.