به نام خدا

گزارش کار آزمایشگاه معماري کامپیوتر

آزمایش جلسه سوم

عنوان آزمایش:

طراحی یک counter و نمایش روی 7 Segment

نام استاد:

استاد علی جوادی

اعضای گروه:

غزل عربعلی - بهاره کاوسی نژاد

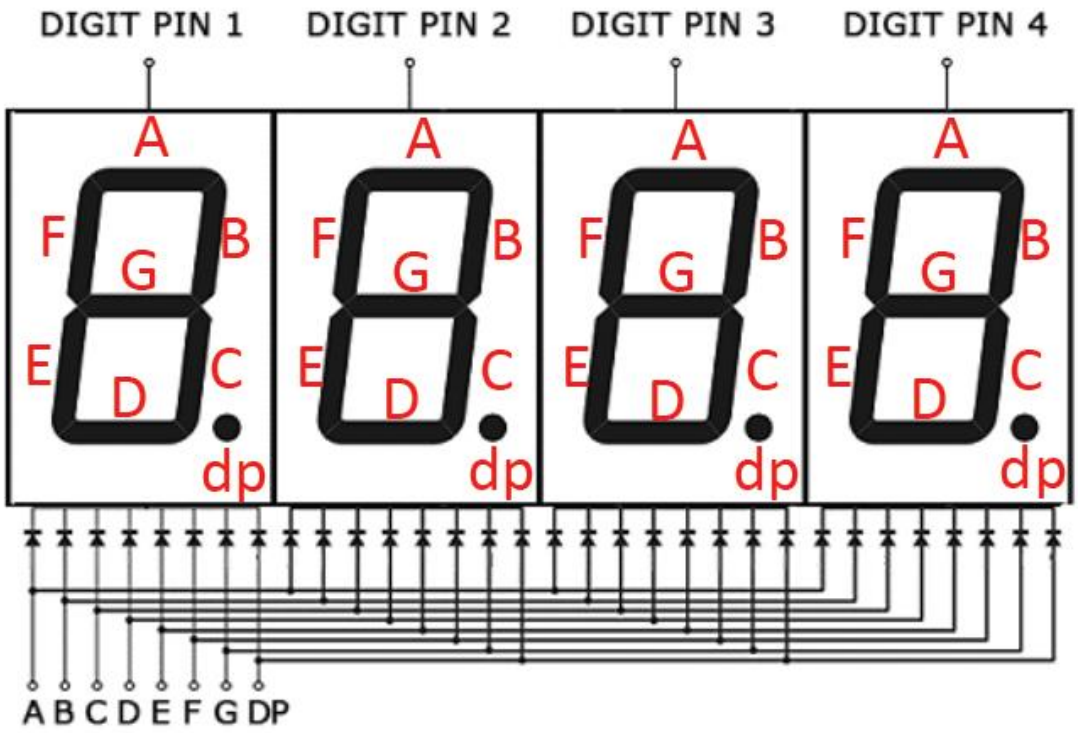
آزمایش: طراحی یک counter و نمایش روی 7 Segment

هدف آزمایش: نمایش اعداد 0 تا 9999 روی بخش 7 Segment برد

تئوري آزمايش:

برای نمایش ارقام می توانیم ۴ تا ورودی ۸تایی برای هر رقم داشته باشیم، اما برای کاهش تعداد ورودی ها می توان از ۸ بیت ورودی استفاده کرد و برای هر رقم یک Common Pin قرار میدهیم.

LED های 7 Segment ها با فرکانسی که قابل تشخیص برای چشم انسان نیست روشن میشوند؛ در واقع، در هر لحظه فقط یکی از ارقام روشن هستند.



ساختار کد :

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL ;
```

کتابخانه "IEEE" شامل پکیج ها و تعریف های بسیاری است . پکیج های موجود در آن برای دیتا تایپ ها و عملیات های ریاضی و همینطور منطقی در طراحی های دیجیتال به کار میرود .

پس از ساختن پروژه ، یک باس خروجی ۸ بیتی، یک باس خروجی ۴ بیتی و یک بیت GCLK را به عنوان ورودی و خروجی ها تعیین میکنیم. این بخش از کد به صورت خودکار پس از تعریف پورت های ورودی و خروجی generate میشود .

```
entity SevenSegmentCounter is
    Port ( GCLK : in  STD_LOGIC;
           SEG_DATA : out std_logic_vector (7 downto 0);
           SEG_SEL : out std_logic_vector(4 downto 0)
           );
end SevenSegmentCounter;
```

سیگنال ها را بین architecture و begin تعریف میکنیم :

```
architecture Behavioral of SevenSegmentCounter is
    signal CLK10MS,CLK1S : STD_LOGIC;
    signal SEGDATABCD1 , SEGDATABCD2,SEGDATABCD3 , SEGDATABCD4 : std_logic_vector ( 3 downto 0 ) ;
    signal SEG_DATA_REG1 , SEG_DATA_REG2 ,SEG_DATA_REG3,SEG_DATA_REG4 : std_logic_vector ( 7 downto 0 ) ;
begin
```

- CLK10MS و CLK1MS به صورت STD_LOGIC تعریف میشوند .
- SEG_DATA_REG1 ، SEG_DATA_REG2 ، SEG_DATA_REG و SEG_DATA_REG4به صورت std_logic_vector که برای مشخص کردن هر کدام از ۸ بیت 7 Segment استفاده میشوند.
- SEGDATABCD1، SEGDATABCD2، SEGDATABCD3 و SEGDATABCD4 که برای مشخص کردن روشن بودن یکی از ارقام 7 Segment استفاده میشوند .

مانند آزمایش قبلی نگاشت اعداد ۴بیتی BCD به 7Segment را انجام میدهیم و در SEG_DATA_REG ها ذخیره میکنیم.

```
with SEGDATABCD1 select
    SEG_DATA_REG1 <= "00111111" when "0000" ,
        "01011011" when "0010" ,
        "00000110" when "0001" ,
        "01001111" when "0011",
        "01100110" when "0100",
        "01101101" when "0101" ,
        "01111101" when "0110",
        "00000111" when "0111",
        "01111111" when "1000",
        "01101111" when "1001",
        "00000000" when others ;
with SEGDATABCD2 select
    SEG_DATA_REG2 <= "00111111" when "0000" ,
        "01011011" when "0010" ,
        "00000110" when "0001" ,
        "01001111" when "0011",
        "01100110" when "0100",
        "01101101" when "0101" ,
        "01111101" when "0110",
        "00000111" when "0111",
        "01111111" when "1000",
        "01101111" when "1001",
        "00000000" when others ;
with SEGDATABCD3 select
    SEG_DATA_REG3 <= "00111111" when "0000" ,
        "01011011" when "0010" ,
        "00000110" when "0001" ,
        "01001111" when "0011",
        "01100110" when "0100",
        "01101101" when "0101" ,
        "01111101" when "0110",
        "00000111" when "0111",
        "01111111" when "1000",
        "01101111" when "1001",
        "00000000" when others ;
with SEGDATABCD4 select
    SEG_DATA_REG4 <= "00111111" when "0000" ,
        "01011011" when "0010" ,
        "00000110" when "0001" ,
        "01001111" when "0011",
        "01100110" when "0100",
        "01101101" when "0101" ,
        "01111101" when "0110",
        "00000111" when "0111",
        "01111111" when "1000",
        "01101111" when "1001",
        "00000000" when others ;
```

دو کلاک 10ms و 1s میسازیم. سپس طي يك process در هر rising edge کلاك GCLK، يك واحد به count که همان مقدار شمارنده است اضافه میکنیم.

```
process(GCLK)
    variable count_div : integer range 0 to 100000 := 0 ;
begin
    if(rising_edge(GCLK)) then
        if count_div < 80000 then
            count_div := count_div+1;
        else
            count_div:=0;
            CLK10MS <= not CLK10MS;
        end if;
    end if;
end process;

process(CLK10MS)
variable count_div1S : integer range 0 to 50 := 0 ;
begin
    if(rising_edge(CLK10MS)) then
        if count_div1S < 50 then
            count_div1S := count_div1S+1;
        else
            count_div1S :=0;
            CLK1S <= not CLK1S;
        end if;
    end if;
end process;
```

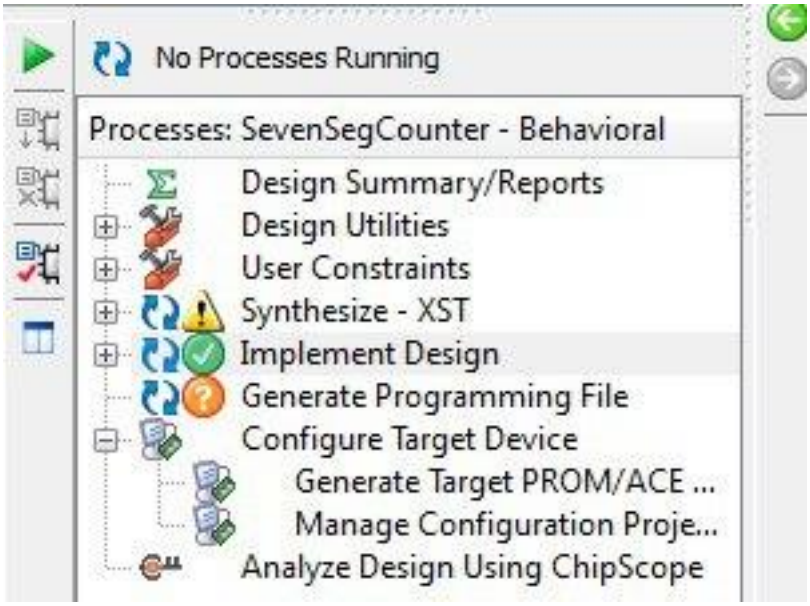
همچنین باید مقدار Digit Pin ها را تعیین کنیم که ارقام به ترتیب روشن و خاموش شوند. این کار با مقداردهي به خروجي‌هاي SEGDATABCD1 ، SEGDATABCD2 ، SEGDATABCD3 و SEGDATABCD4 انجام ميشود .

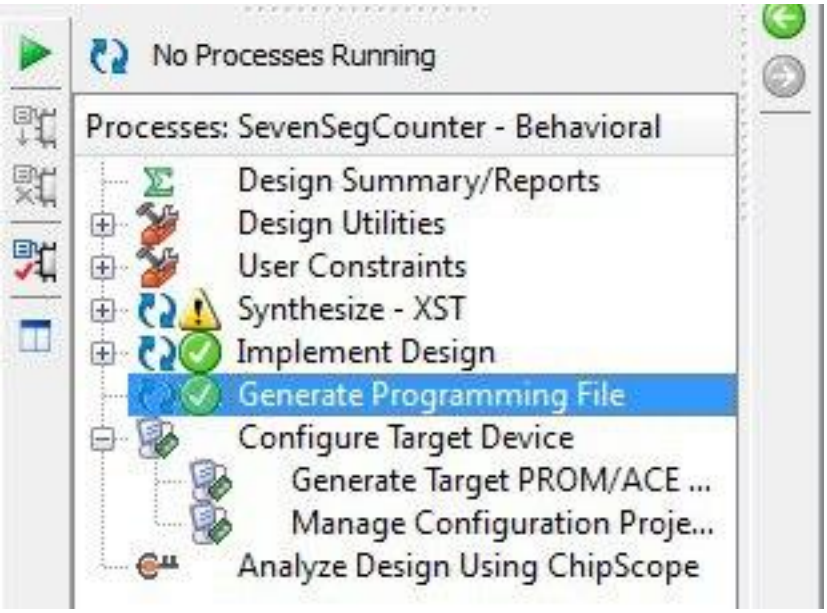
```
process(CLK1S)
variable counter : integer range 0 to 9999 :=0 ;
variable VSEGDATABCD1 ,VSEGDATABCD2 , VSEGDATABCD3 , VSEGDATABCD4 :
std_logic_vector ( 3 downto 0 ) := "0000" ;
begin
    if ( counter < 9999 and rising_edge(CLK1S ) ) then
        if(VSEGDATABCD1 <"1001" ) then
            VSEGDATABCD1 :=(std_logic_vector(VSEGDATABCD1) + 1);
            else
            VSEGDATABCD1 := "0000";
            if ( VSEGDATABCD2 < "1001" ) then
                VSEGDATABCD2 := (std_logic_vector(VSEGDATABCD2) + 1);
            else
                VSEGDATABCD2 := "0000";
                if ( VSEGDATABCD3 < "1001" ) then
                    VSEGDATABCD3 := (std_logic_vector(VSEGDATABCD3) + 1);
                else
                    VSEGDATABCD3 := "0000" ;
                    if( VSEGDATABCD4 < "1001" ) then
                        VSEGDATABCD4 := (std_logic_vector(VSEGDATABCD4) + 1);
                    else
                        VSEGDATABCD4 := "0000" ;
                    end if ;
                end if;
            end if ;
        end if ;
            counter := counter +1 ;

    end if ;
            SEGDATABCD1 <= VSEGDATABCD1;
            SEGDATABCD2 <= VSEGDATABCD2;
            SEGDATABCD3 <= VSEGDATABCD3;
            SEGDATABCD4 <= VSEGDATABCD4;
end process;
```

سپس ورودی و خروجی ها را به برد map میکنیم :

```
1  NET "GCLK" LOC = P184;
2  net "GCLK" CLOCK_DEDICATED_ROUTE = FALSE;
3  NET "SEG_DATA[0]" LOC = P10;
4  NET "SEG_DATA[1]" LOC = P7;
5  NET "SEG_DATA[2]" LOC = P11;
6  NET "SEG_DATA[3]" LOC = P5;
7  NET "SEG_DATA[4]" LOC = P4;
8  NET "SEG_DATA[5]" LOC = P12;
9  NET "SEG_DATA[6]" LOC = P9;
10 NET "SEG_DATA[7]" LOC = P3;
11 NET "SEG_SEL[0]" LOC = P15;
12 NET "SEG_SEL[1]" LOC = P20;
13 NET "SEG_SEL[2]" LOC = P19;
14 NET "SEG_SEL[3]" LOC = P18;
15 NET "SEG_SEL[4]" LOC = P16;
```

سپس مانند قبل به اجرا کردن آن میپردازیم :





بحث و نتیجه گیری :

در انتها مشاهده میکنیم که نمایشگر 7 segment شروع به نمایش و شمردن اعداد میکند.