

پیش‌گزارش طراحی چراغ راهنمایی برای چهارراه (قسمت اول)

نام: آناهیتا
نام خانوادگی: اسدی
شماره دانشجویی: 9724453
نام مدرس: سرکار خانم فرشته کلانتری
روز و ساعت کلاس: یکشنبه‌ها، ساعت 13:30 الی 15:30

پاسخ مرحله 1-1

1.

```
library IEEE;  
USE IEEE.std_logic_1164.ALL;  
USE IEEE.std_logic_arith.ALL;  
USE IEEE.std_logic_unsigned.ALL;  
USE STD.TEXTIO.ALL;
```

اضافه کردن کتابخانه‌های موردنیاز؛ کتابخانه اول برای گیت‌های منطقی (در این آزمایش not کردن یک سیگنال)، کتابخانه دوم برای عملیات ریاضی (مانند جمع کردن i و count با 1) و کتابخانه سوم برای اعداد غیرعلامت دار (برای مثال 1000 را 8 (و نه 7) در نظر می‌گیرد) است. کتابخانه آخر، برای خواندن فایل است که عملیات readline و... و تایپ‌های file و line در آن تعریف شده.

```
entity sevensegment is port(  input1, input2: in std_logic_vector(6 downto 0);  
                             mode: in std_logic_vector(1 downto 0);  
                             clk: in std_logic;  
                             output: out bit_vector(7 downto 0);  
                             com: out std_logic_vector(3 downto 0));  
end sevensegment;
```

تعریف پورت‌های ورودی و خروجی. Output به این دلیل bit_vector تعریف شده که خروجی فایل (که از همین جنس است) مستقیماً به آن وصل شده است. تفاوت bit_vector با std_logic_vector در آن است که تایپ اول، در بیت‌های خود تنها 0 و 1، و تایپ دوم علاوه بر 0 و 1، انواع دیگر مانند U و X و... هم قبول می‌کند

```
architecture behavioral of sevensegment is
```

```
type RAM_type is array (0 to 9) of bit_vector(7 downto 0);
```

```
function ram_initialize (filename: in string) return RAM_type is  
    file file1: text is in filename;  
    variable line1: line;  
    variable RAM1: RAM_type;  
    begin for i in RAM_type'range loop  readline(file1, line1);  
                                         read(line1, RAM1(i));  
    end loop;  
    return RAM1;  
end function;
```

```
signal RAM: RAM_type := ram_initialize("values.txt");
```

در این قسمت، زیربرنامه خواندن از فایل (که مقادیر 0 تا 9 باینری به سون سگمنت در آن نوشته شده اند) نوشته شده است و سیگنال RAM با مقادیر آن مقداردهی شده است.

```
signal ones: std_logic_vector(6 downto 0) := "0000000";
signal tens: std_logic_vector(3 downto 0) := "0000";

type state is (M0, M1, M2, M3, M4, M5, M6, M7, M8, M9, M10, M11);
signal pr_state1: state := M0;
signal pr_state2: state := M6;

begin

process(clk)
begin
    if rising_edge(clk) then
        if (mode = "01") then
            case pr_state1 is
                when M0 => ones <= input1;
                           tens <= "0000";
                           pr_state1 <= M1;
                when M1 => if (ones >= "1010") then
                           ones <= ones - "1010";
                           tens <= tens + "0001";
                           pr_state1 <= M1;
                           else
                           pr_state1 <= M2;
                           end if;
                when M2 => com <= "0001";
                           output <= RAM(conv_integer(ones(3 downto 0)));
                           pr_state1 <= M3;
                when M3 => com <= "0010";
                           output <= RAM(conv_integer(tens));
                           pr_state1 <= M4;
                when M4 => com <= "0100";
                           output <= "10111111";
                           pr_state1 <= M5;
                when M5 => com <= "1000";
                           output <= "10111111";
                           pr_state1 <= M2;
                when others => pr_state1 <= M0;
            end case;
        end if;
    end if;
end process;
```

کلا دو حالت $mode = "01"$ و $mode = "10"$ تعریف شده اند. برای هریک، pr_state (حالت فعلی) جداگانه تعریف شده است. این سیگنال در اصل مانند پوینتر حالت های مختلف ماشین حالت عمل میکند.

ابتدا سیگنال ورودی در $ones$ گذاشته میشود، و مقدار $ones$ (یکان) 10 تا 10 تا کم میشود (و با هر بار کم شدن، $tens$ (دهگان) یک واحد زیاد میشود). این عمل تا جایی ادامه میابد که یکان از 10 کوچکتر شود. سپس به حالت بعدی میرود.

4 حالت بعدی، هریک متعلق به یکی از سون سگمنت ها هستند (از مقداردهی com آنها مشخص است) در لوپ قرار میگیرند. و آنقدر سریع سوییچ میشوند که چشم آنها را بصورت 4 رقم همزمان میبیند.

M2 رقم یکان، M3 دهگان، M4 و M5 برای زوج سون سگمنت $input2$ هستند که بصورت "-" نشان داده میشوند.

در صورت ایجاد حالات غیرمترقبه، ماشین حالت برای "01" mode = (pr_state1) به M0 می‌رود.

```

elseif (mode = "10") then
  case pr_state2 is
    when M6 => ones <= input2;
                  tens <= "0000";
                  pr_state2 <= M7;
    when M7 => if (ones >= "1010") then
                  ones <= ones - "1010";
                  tens <= tens + "0001";
                  pr_state2 <= M7;
                else
                  pr_state2 <= M8;
                end if;
    when M8 => com <= "0001";
                  output <= RAM(conv_integer(ones(3 downto 0)));
                  pr_state2 <= M9;
    when M9 => com <= "0010";
                  output <= RAM(conv_integer(tens));
                  pr_state2 <= M10;
    when M10 => com <= "0100";
                  output <= "10111111";
                  pr_state2 <= M11;
    when M11 => com <= "1000";
                  output <= "10111111";
                  pr_state2 <= M8;
    when others => pr_state2 <= M6;
  end case;
end if;
end if;
end process;
end behavioral;

```

موارد توضیح داده شده، برای "10" mode = (pr_state2) نیز اجرا شده‌اند.

پاسخ مرحله 2-1

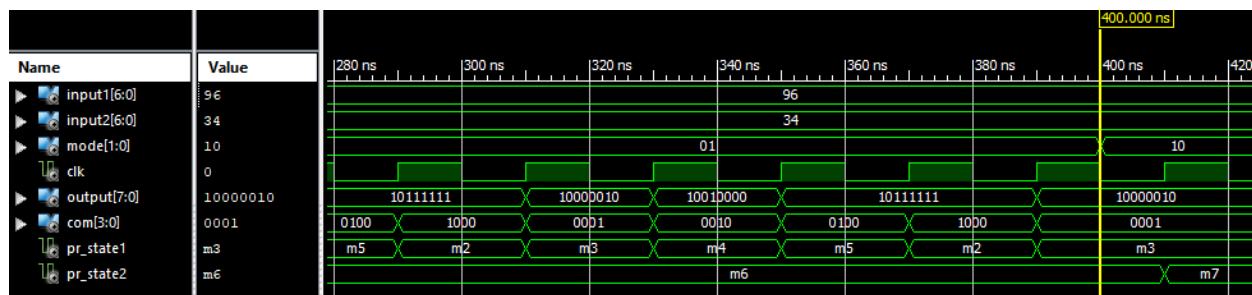
کد تست بنچ:

```

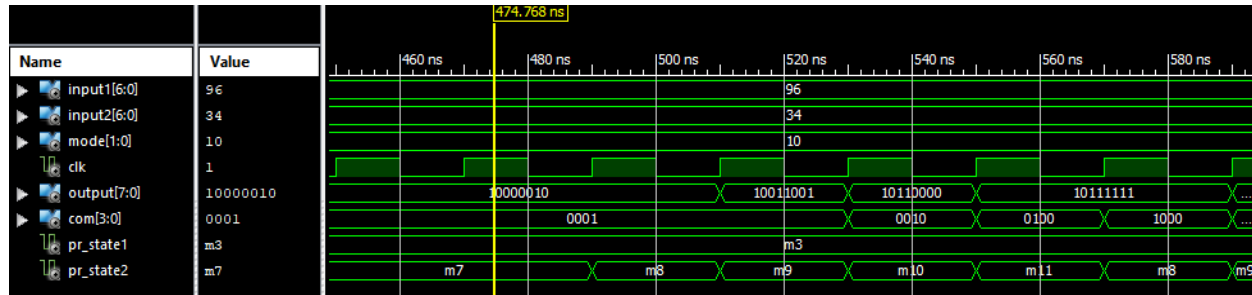
clk <= not(clk) after 10 ns;
mode <= "01" after 2 ns, "10" after 400 ns;
input1 <= "1100000" after 1 ns;
input2 <= "0100010" after 1 ns;

```

نتیجه تست بنچ (بزرگنمایی برای "01" mode =)



نتیجه تست بنچ (بزرگنمایی برای "10" = mode)



پاسخ مرحله 3-1

کد UCF:

```

net "input1[0]" loc = p34;
net "input1[1]" loc = p33;
net "input1[2]" loc = p31;
net "input1[3]" loc = p29;
net "input1[4]" loc = p28;
net "input1[5]" loc = p27;
net "input1[6]" loc = p26;

net "input2[0]" loc = p12;
net "input2[1]" loc = p11;
net "input2[2]" loc = p10;
net "input2[3]" loc = p9;
net "input2[4]" loc = p7;
net "input2[5]" loc = p5;
net "input2[6]" loc = p4;

net "mode[0]" loc = p24;
net "mode[1]" loc = p3;

net "clk" loc = p80;

net "output[0]" loc = p140;
net "output[1]" loc = p139;
net "output[2]" loc = p138;
net "output[3]" loc = p137;
net "output[4]" loc = p135;
net "output[5]" loc = p133;
net "output[6]" loc = p132;
net "output[7]" loc = p131;

net "com[0]" loc = p125;
net "com[1]" loc = p126;
net "com[2]" loc = p128;
net "com[3]" loc = p130;

```

ورودی های [input1[0-6] به D0-6 از DIP Switch 1 و بیت mode مربوط به آن به D7 از DIP Switch 1 وصل شده است.

ورودی های [input2[0-6] به D0-6 از DIP Switch 2 و بیت mode مربوط به آن به D7 از DIP Switch 2 وصل شده است.

Clk به کلاک داخلی FPGA وصل شده است.

خروجی ها به 7SEG متصل شده اند. و com به پایه های Common.

پاسخ مرحله 1-2

```
library IEEE;
USE IEEE.std_logic_1164.ALL;
USE IEEE.std_logic_arith.ALL;
USE IEEE.std_logic_unsigned.ALL;
```

اضافه کردن کتابخانه های موردنظر

```
entity input_control is port( input1, input2: in std_logic_vector(6 downto 0);
                             apply: in std_logic;
                             Green, Yellow, Red: out std_logic_vector(6 downto 0));
end input_control;
```

معرفی پورت های ورودی و خروجی

```
architecture behavioral of input_control is
begin
process(apply)
begin
    if rising_edge(apply) then
        if (input1 < "1100011" and input2 < "1100011" and input1 > "0000000" and input2 > "0000000" and input1 /= input2) then
            if (input1 > input2) then
                Red <= input1;
                Green <= input2;
                Yellow <= input1 - input2;
            elsif (input1 < input2) then
                Red <= input2;
                Green <= input1;
                Yellow <= input2 - input1;
            end if;
        end if;
    end if;
end process;
end behavioral;
```

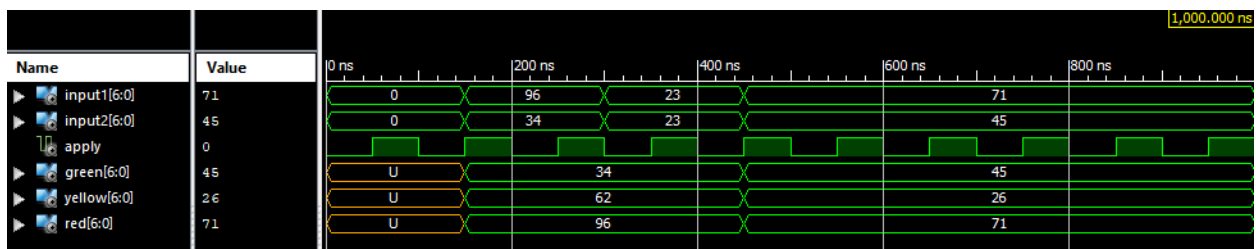
apply در پروسه گذاشته میشود. با هر لبه بالارونده آن، دو ورودی (با شرایط مذکور) با هم مقایسه میشوند، و تفاضل آنها در Yellow، عدد بزرگتر در Red و عدد کوچکتر در Green ریخته میشود. در صورتیکه شرایط برقرار نبودند، خروجی ها تغییر نمیکنند (در تست بنچ مشخص است)

پاسخ مرحله 2-2

کد تست بنچ:

```
input1 <= "1100000" after 150 ns, "0010111" after 300 ns, "1000111" after 450 ns;
input2 <= "0100010" after 150 ns, "0010111" after 300 ns, "0101101" after 450 ns;
apply <= not(apply) after 50 ns;
```

نتیجه تست بنچ:



پاسخ مرحله 2-3

کد UCF:

```
net "input1[0]" loc = p34;
net "input1[1]" loc = p33;
net "input1[2]" loc = p31;
net "input1[3]" loc = p29;
net "input1[4]" loc = p28;
net "input1[5]" loc = p27;
net "input1[6]" loc = p26;

net "input2[0]" loc = p12;
net "input2[1]" loc = p11;
net "input2[2]" loc = p10;
net "input2[3]" loc = p9;
net "input2[4]" loc = p7;
net "input2[5]" loc = p5;
net "input2[6]" loc = p4;

net "clk" loc = p181;
```

چون ماهیت ورودی (دیپ سویچ) و خروجی (سون سگمنت یا LED) برای پیاده سازی سخت افزاری متفاوت است، باید با سیم به هم (توسط خود FPGA با دستور add component) متصل شوند.