

# گزارشکار جلسه چهارم آزمایشگاه معماری

آرمین افضلی 400521054

محمد صالح پزند 400521171

در این آزمایش قصد داریم یک ثانیه و دقیقه شمار با قابلیت START و STOP و RESET و ALARM درست کنیم که عدد BCD هر 7SEG را در LED های برد نمایش دهد. این ویژگی ها به این صورت پیاده سازی میشوند:

START , STOP : برای هر کدام یک کلید فشاری در نظر میگیریم . این کلید ها به طور پیش فرض مقدارشان 1 است پس برای شروع باید مقدار ان صفر شود .

RESET: با فشردن این کلید فشاری ساعت از صفر شروع به کار میکند.

ALARM: میتوان برای ساعت الارم در نظر گرفت به این صورت که بعد از رسیدن به یک مقدار دلخواه یک BUZZER فعال میشود و فقط با ریست شدن متوقف میشود.

از آنجایی که این آزمایش در ادامه آزمایش قبل است فقط تغییرات اعمال شده مورد بررسی قرار میگیرد.

بیس این آزمایش همان شمارنده آزمایش قبل است با این تفاوت که دو رقم اول از 00 تا 59 میروند و دو رقم بعدی از 00 تا 24 .

```

entity bcd_7seg is
  Port ( gclock : in  STD_LOGIC;
        stop : in std_logic;
        start : in std_logic;
        reset : in std_logic;
        alarm : out std_logic;
        selseg : out std_logic_vector (4 downto 0);
        output : out  STD_LOGIC_VECTOR (6 downto 0);
        binary_out1 : out std_logic_vector (3 downto 0);
        binary_out2 : out std_logic_vector (3 downto 0);
        binary_out3 : out std_logic_vector (3 downto 0);
        binary_out4 : out std_logic_vector (3 downto 0));
end bcd_7seg;

```

در ابتدا باید اینپوت های START , STOP , RESET را مشخص کنیم. همچنین یک خروجی ALARM که به BUZZER مپ میشود اضافه میکنیم. اندازه selseg را هم یکی افزایش میدهیم و آخرین مقدار آن یعنی selseg[4] را به دو نقطه چشمک زن وسط ساعت اختصاص میدهیم. برای نمایش دادن BCD هر عدد هم یک خروجی 4 بیتی تعریف میکنیم که به LED ها مپ می شود.

```

signal counter1, counter2, counter3, counter4 : std_logic_vector(3 downto 0) := "0000";
signal data1, data2, data3, data4 : std_logic_vector (6 downto 0) := "00000000";
signal data5 : std_logic_vector(2 downto 0) := "000";
signal cont : std_logic := '0';

```

یک سیگنال cont تعریف میکنیم که حالت STOP یا START بودن ساعت را مشخص میکند. همچنین یک سیگنال 3 بیتی برای دو نقطه در نظر میگیریم .

```

variable refresh : integer range 0 to 5 := 0;
begin
if (rising_edge(clock10ms)) then
if (refresh < 5) then
refresh := refresh + 1;
else
refresh := 0;
end if;

case refresh is
when 0 =>
selseg(4) <= '0';
selseg(0) <= '1';
output <= data1;
when 1 =>
selseg(0) <= '0';
selseg(1) <= '1';
output <= data2;
when 2 =>
selseg(1) <= '0';
selseg(2) <= '1';
output <= data3;
when 3 =>
selseg(2) <= '0';
selseg(3) <= '1';
output <= data4;
when 4 =>
selseg(3) <= '0';
selseg(4) <= '1';
output(2 downto 0) <= data5;
when others => null;
end case;
end if;
end;

```

در این PROCESS که رفرش شدن هر کدام از 7SEG ها مشخص میکند یک حالت پنجم اضافه میکنیم که مربوط به روشن شدن دو نقطه است.

```

process(clock1s, reset)

variable value1, value2, value3, value4 : std_logic_vector(3 downto 0) := "0000";

begin
if (reset = '0') then
value1 := "0000";
value2 := "0000";
value3 := "0000";
value4 := "0000";

counter1 <= "0000";
counter2 <= "0000";
counter3 <= "0000";
counter4 <= "0000";
alarm <= '0';
elsif (rising_edge(clock1s) and cont = '1') then
if (value1 < "1001") then
value1 := value1 + 1;
else
value1 := "0000";
if (value2 < "0101") then
value2 := value2 + 1;
else
value2 := "0000";
if (value3 < "1001") then
value3:= value3 + 1;
else
value3 := "0000";

```

در این PROCESS ریست را دریافت میکنیم و چک میکنیم اگر صفر شد مقادیر اعداد ساعت را صفر کنیم و اگر الارم روشن بود آن را خاموش میکنیم.

در ادامه یک شرط به شمارنده اضافه میکنیم که هر موقع **cont** فعال بود به شمارش ادامه بدهد همچنین از انجایی که در ساعت 7SEG دوم و چهارم از 0 تا 5 میروند ، مقادیر BCD را هم در این دو شرط تغییر میدهیم .

```

counter4 <= value4;
counter3 <= value3;
counter2 <= value2;
counter1 <= value1;
binary_out1 <= value1;
binary_out2 <= value2;
binary_out3 <= value3;
binary_out4 <= value4;

if (value1(0) = '0') then
    data5 <= "011";
else
    data5 <= "000";
end if;
if (value2 = "0011") then
    alarm <= '1';
end if;

```

این قسمت که ادامه process قبل است مقادیر BCD اعداد را وارد دیتای LED ها میکنیم . همچنین برای چراغ چشمک زن یک شرط میگذاریم که هر زمان مقدار پر ارزش ترین رقم BCD ثانیه شمار 0 بود چراغ روشن و اگر 1 بود خاموش شود و چون این مقدار هر ثانیه عوض میشد باعث چشمک زن شدن چراغ میشود. مقدار این 7SEG سه بیت است . دو بیت برای دو نقطه وسط ساعت و یک بیت برای درجه که این یک بیت در تمامی حالات باید صفر باشد .

در اخر هم یک مقدار دلخواه که در اینجا 30 ثانیه در نظر گرفته شده برای الارم تعیین میکنیم که بعد از ان الارم روشن میشود و تا وقتی که ریست نشود روشن میماند.

```
process(start, stop)
begin

    if (start = '0' and stop = '1') then
        cont <= '1';
    elsif (start = '1' and stop = '0') then
        cont <= '0';
    end if;
end process;
```

این **process** برای روشن و خاموش کردن است که با فشردن هر کدام از کلیدها مقدار **cont** تغییر میکند و ساعت متوقف یا به کار خود ادامه میدهد. چون مقدار اولیه کلید 1 است پس برای شروع به کار کردن باید 0 شود و برعکس.

```

NET "gclock" CLOCK_DEDICATED_ROUTE = FALSE;
NET "gclock" LOC = P184;
NET "output[0]" LOC = P10;
NET "output[1]" LOC = P7;
NET "output[2]" LOC = P11;
NET "output[3]" LOC = P5;
NET "output[4]" LOC = P4;
NET "output[5]" LOC = P12;
NET "output[6]" LOC = P9;
NET "selseg[0]" LOC = P15;
NET "selseg[1]" LOC = P20;
NET "selseg[2]" LOC = P19;
NET "selseg[3]" LOC = P18;
NET "selseg[4]" LOC = P16;
NET "alarm" LOC = P13;
NET "start" LOC = P190;
NET "stop" CLOCK_DEDICATED_ROUTE = FALSE;
NET "stop" LOC = P189;
NET "reset" LOC = P187;

```

```

---
NET "binary_out1[0]" LOC = P42;
NET "binary_out1[1]" LOC = P43;
NET "binary_out1[2]" LOC = P44;
NET "binary_out1[3]" LOC = P45;
NET "binary_out2[0]" LOC = P46;
NET "binary_out2[1]" LOC = P48;
NET "binary_out2[2]" LOC = P50;
NET "binary_out2[3]" LOC = P51;
NET "binary_out3[0]" LOC = P61;
NET "binary_out3[1]" LOC = P62;
NET "binary_out3[2]" LOC = P63;
NET "binary_out3[3]" LOC = P64;
NET "binary_out4[0]" LOC = P65;
NET "binary_out4[1]" LOC = P67;
NET "binary_out4[2]" LOC = P68;
NET "binary_out4[3]" LOC = P71;

```

در انتها ورودی و خروجی های جدید را مپ میکنیم. سه ورودی START, STOP,

RESET به سه کلید و الارم به BUZZER داده میشود. چراغ چشمک زن هم مپ

میشود و هرکدام از مقادیر BDC اعداد ساعت هم به LED های متناظر متصل میشود.

