

گزارشکار آزمایشگاه معماری جلسه 5

آرمین افضلی 400521054

محمد صالح پزند 400521171

در این آزمایش قصد داریم به وسیله یک پورت سریال یک داده 8 بیتی را انتقال بدهیم و در سیستم آن را بخوانیم.

برای این کار از دو پراسس استفاده میکنیم یکی برای ساختن کلاک مناسب هماهنگ با استاندارد ها و یکی برای پیاده سازی برنامه اصلی. ما در اینجا با 115200 bauds کار میکنیم. پس باید کلاک اصلی دستگاه را 357 بار کند بکنیم؛ پس هر 173 سیکل کلاک مد نظر را not میکنیم.

```
signal clock : std_logic := '0';

begin

process (gclk)
variable count_div : integer range 0 to 173 := 0;
begin
    if (rising_edge(gclk)) then
        if (count_div < 173) then
            count_div := count_div + 1;
        else
            count_div := 0;
            clock <= not clock;
        end if;
    end if;
end process;
```

```
entity rs232 is
    Port ( gclk : in  STD_LOGIC;
          TX : out  STD_LOGIC;
          DIP : in  STD_LOGIC_VECTOR (7 downto 0));
end rs232;

architecture Behavioral of rs232 is
```

برای شروع کلاک برد (gclock) را به عنوان ورودی تعریف میکنیم. خروجی TX هم اطلاعات را به صورت سریال انتقال میدهد. ورودی 8 بیتی هم از dip switch گرفته میشود.

```

process (clock)
variable counter : integer range 1 to 12 := 1;
variable dip_c : integer range 0 to 8 := 0;
variable flag_c : integer range 0 to 80000 := 0;
begin

    if (rising_edge(clock)) then
        if (flag_c = 80000) then
            if (counter = 1) then
                TX <= '0';
            elsif (counter = 10) then
                TX <= '0';
            elsif (counter = 11) then
                TX <= '1';
                flag_c := 0;
            else
                TX <= DIP(dip_c);

                if (dip_c < 8) then
                    dip_c := dip_c + 1;
                else
                    dip_c := 0;
                end if;
            end if;

            if (counter < 12) then
                counter := counter + 1;
            else
                counter := 1;
            end if;
        else
            flag_c := flag_c + 1;
        end if;
    end if;
end if;

```

برای فرستادن اطلاعات باید ابتدا یک بیت 0 بفرستیم به نشانه استارت شدن . سپس 8 بیت دیتای dip switch ها را وارد میکنیم. در انتها باید یک بیت 0 دیگر برای stop کردن میفرستیم و بعد از آن 1 فرستاده میشود که همان استیت ابتدایی برنامه است، در مجموع 11 بیت؛ پس یک شمارنده از 1 تا 11 در نظر میگیریم و با خواندن هر بیت به آن یکی اضافه میکنیم. یک شمارنده dip_c (از 0 تا 7) برای انتخاب بیت ورودی و ذخیره آن در خروجی است. علاوه بر این برای اینکه خروجی هر یک ثانیه یکبار فرستاده شود یک شمارنده قرار میدهیم که هر 80000 سیکل کلاک یکبار خروجی را ارسال کند که در نتیجه خروجی تقریباً هر یک ثانیه یک بار ارسال میشود

```
NET "DIP[0]" LOC = P183;  
NET "DIP[1]" LOC = P182;  
NET "DIP[2]" LOC = P181;  
NET "DIP[3]" LOC = P180;  
NET "DIP[4]" LOC = P178;  
NET "DIP[5]" LOC = P176;  
NET "DIP[6]" LOC = P175;  
NET "DIP[7]" LOC = P172;  
NET "gclk" CLOCK_DEDICATED_ROUTE = FALSE;  
NET "gclk" LOC = P184;  
NET "TX" LOC = P40;
```

در انتها ورودی ها را به dip switch ها و خروجی را به پایه خروجی پورت سریال
مپ میکنیم.