

به نام خدا

گزارش کار آزمایشگاه معماری کامپیوتر

آزمایش جلسه پنجم

عنوان آزمایش:

UART Serial Communication

نام استاد:

استاد علی جوادی

اعضای گروه:

غزل عربعلی - بهاره کاوسی نژاد

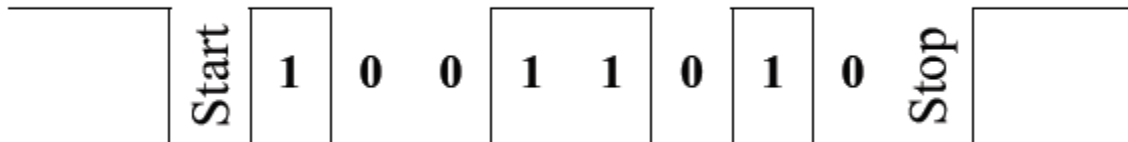
آزمایش: UART Serial Communication

هدف آزمایش: نمایش بیت‌های ارسالی از طریق IDC Telecom Connector

تئوری آزمایش:

در این آزمایش می‌خواهیم بیت‌هایی که روی Dip Switch ها مشخص می‌شوند را از طریق IDC Telecom Connector در برنامه Docklight نمایش دهیم.

روش کار پروتکل به این صورت است که برای مشخص ساختن بیت اول ابتدا یک بیت 0 و برای مشخص ساختن پایان نیز 01 ارسال می‌کند.



Bauds	Bit duration	Speed
9600 bauds	104.167 μ s	1200 bytes/s
19200 bauds	52.083 μ s	2400 bytes/s
28800 bauds	34.722 μ s	3600 bytes/s
38400 bauds	26.042 μ s	4800 bytes/s
57600 bauds	17.361 μ s	7200 bytes/s
76800 bauds	13.021 μ s	9600 bytes/s
115200 bauds	8.681 μ s	14400 bytes/s

فرکانس مورد برابر با 40 مگاهرتز است پس:

$$40,000,000 / 115200 = 347.2 \text{ Hz}$$

$$347.2 / 2 = 173.61 \sim 174 \text{ Hz}$$

در نتیجه باید یک کلاک با فرکانس 174 هرتز بسازیم.

روش و چگونگی انجام آزمایش:

در ابتدا ورودی‌ها و خروجی را مشخص می‌کنیم:

- GCLK: ورودی
- TX: خروجی برای نمایش توسط IDC Telecom Connector
- DIP: یک باس 8 بیتی ورودی که همان بیت‌های Dip Switch ها هستند.

```
entity RS232SerialCommunication is
    Port ( GCLK : in  STD_LOGIC;
          TX : out  STD_LOGIC;
          DIP : in  STD_LOGIC_VECTOR (7 downto 0));
end RS232SerialCommunication;
```

در مرحله بعد CLOCK174 را می‌سازیم:

```
architecture Behavioral of RS232SerialCommunication is
    signal CLOCK174 : STD_LOGIC := '0';
begin

    process(GCLK)
        variable CounterClock174 : integer range 0 to 100000 := 0;
    begin
        if(rising_edge(GCLK)) then
            if(CounterClock174 < 174) then
                CounterClock174 := CounterClock174 + 1;
            else
                CounterClock174 := 0;
                CLOCK174 <= not CLOCK174;
            end if;
        end if;
    end process;
```

سه متغیر تعریف می‌کنیم:

- DipSwitchingBits: بیت‌های dip switch ها
- BitCounter: به منظور شمارش تعداد بیت‌ها است و مقداری از 0 تا 11 دارد.
- ClockCycle

به ازای هر rising edge کلاک 174، فرکانس را با استفاده از ClockCycle مشخص کرده و هر بار چک می‌کنیم که آیا BitCounter، صفر (شروع) و یا 9 یا 10 (دو بیت 01) است یا خیر. اگر هیچکدام از این شروط برقرار نبود، DipSwitchingBits، BitCounter و ClockCycle را یک واحد افزایش می‌دهیم.

```

process (CLOCK174)
variable DipSwitchingBits : integer range 0 to 7 := 0;
-- 1 start(=0) + 8 bit + 2 stop(=01) => 11 bits
variable BitCounter : integer range 0 to 10 := 1;
-- For each 11 bits reading
variable ClockCycle : integer range 0 to 80000 := 0;
begin
    if (rising_edge(CLOCK174)) then
        if(ClockCycle = 80000) then |
            -- first bit = start
            if (BitCounter = 0) then
                TX <= '0';
            -- One before last bit
            elsif (BitCounter = 9) then
                TX <= '0';
            -- Last bit
            elsif (BitCounter = 10) then
                TX <= '1';
                ClockCycle := 0;
            -- Data
            else
                TX <= DIP(DipSwitchingBits);
                -- Increment the data index
                if(DipSwitchingBits < 8) then
                    DipSwitchingBits := DipSwitchingBits + 1;
                -- Data finished
                else
                    DipSwitchingBits := 0;
                end if;
            end if;
            -- Checking the state of 11 bits
            if (BitCounter < 11) then
                BitCounter := BitCounter + 1;
            else
                BitCounter := 0;
            end if;
        else
            ClockCycle := ClockCycle + 1;
        end if;
    end if;
end process;

end Behavioral;

```

بحث و نتیجه گیری:

در انتها با استفاده از Docklight، نتیجه را مشاهده می‌کنیم.