Pre-Report

FPGA Lab Eperiment 2

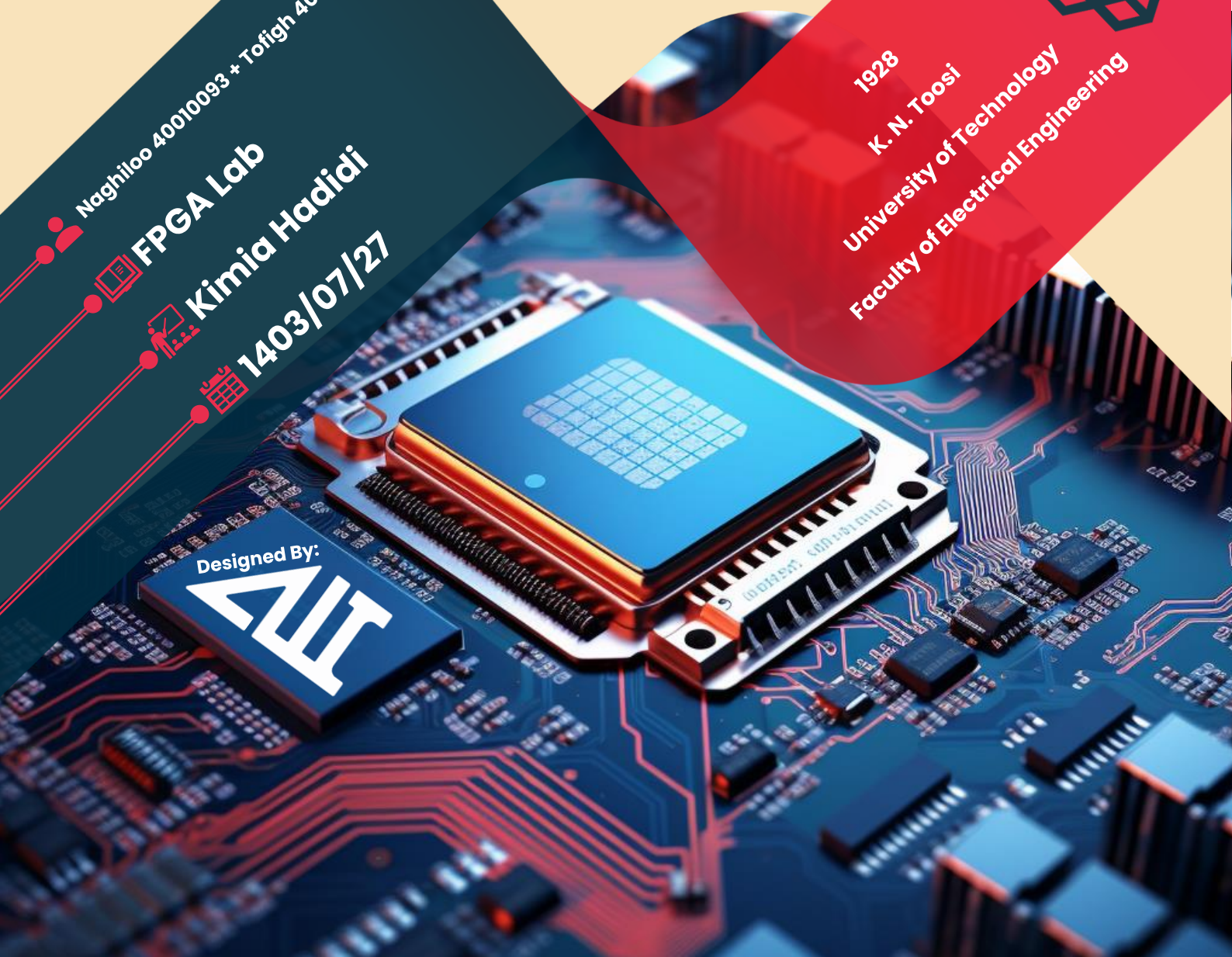
Naghiloo 40010093 + Tofigh 40003913

FPGA Lab

Kimia Hadidi

1403/07/27

1928
K. N. Toosi
University of Technology
Faculty of Electrical Engineering



بخش اول: مقسم فرکانس

مرحله ۱-۱:

۱- با فرض اینکه کلاک ورودی همان اسیلاتور ۵۰ مگاهرتزیِ برد آزمایشگاه است، برای تولید کلاک ۱ هرتز مقدار Div چه خواهد شد؟

$$\frac{f_{CLK_{in}}}{2 \times Div} = \frac{50\ MHz}{2 \times Div} = 1\ Hz \rightarrow Div = 25\ M$$

۲- کد مربوط به این آزمایش را نوشته و جزئیات آن را توضیح دهید.

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

پکیج STD_LOGIC_1164 از کتابخانه IEEE به منظور استفاده از تایپ STD_LOGIC

بقیه پکیج ها برای انجام محاسبات

```
entity experiment_2_1 is
  generic ( n : integer := 25 * (10**6) );
    Port ( CLK_IN : in  STD_LOGIC;
           CLK_OUT : inout  STD_LOGIC:= '0' );
end experiment_2_1;
```

سیگنال های ورودی و خروجی

```
architecture Behavioral of experiment_2_1 is
    signal count : INTEGER := 0;
  signal CLK : STD_LOGIC := '0';
    constant Div : INTEGER := n;
begin
    process(Clk_in)
    begin
        if rising_edge(Clk_in) then
            count <= count + 1;
            if count = (Div-1) then
                    CLK <= NOT CLK;
            count <= 0;
            end if;
        end if;
    end process;
  CLK_OUT <= CLK;

end Behavioral;
```

سیگنال ها و ثابت های واسطه مورد نیاز در بدنه اصلی را اینجا تعریف میکنیم

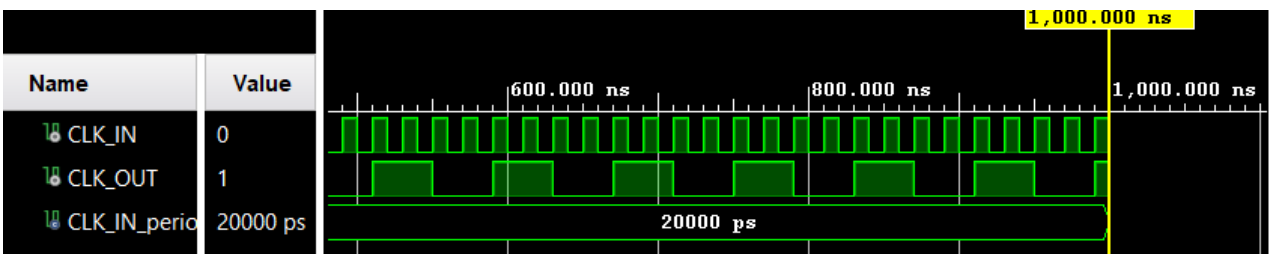
هر لبه بالارونده کلاک ورودی توسط سیگنال count شمرده میشود و با مقدار (Div-1) مقایسه میشود و درصورت برابری سیگنال CLK ما not میشود.

مرحله ۱-۲:

با نوشتن یک Test bench در نرم‌افزار ISE در حالتی که فرکانس کلاک ورودی دلخواه باشد، با انتخاب Div مناسب، کلاک خروجی را با فرکانسی برابر با ۱/۴ فرکانس ورودی تولید کنیم. پاسخ شبیه‌سازی خود را نمایش دهید

```
 -- Instantiate the Unit Under Test (UUT)
uut: entity work.experiment_2_1 generic map ( n => 2 ) PORT MAP ( CLK_IN => CLK_IN, CLK_OUT => CLK_OUT);

    CLK_IN <= not(CLK_IN) after CLK_IN_period/2;
```



مرحله۱-۳:

نتیجه ی گزارش سنتز را برای این طراحی ارائه دهید و دلیلی برای استفاده شدن این مقدار از منابعFPGA را گزارش کنید.

```
=========================================================================
*                      Advanced HDL Synthesis                           *
=========================================================================

=========================================================================
Advanced HDL Synthesis Report

Macro Statistics
# Counters                                             : 1
 32-bit up counter                                     : 1
# Registers                                            : 1
 Flip-Flops                                            : 1
```

یک شمارنده برای شمارش لبه بالارونده (تا ۲۵ میلیون لبه)

برای نگهداری کلاک خروجی تا اتمام شمارش

```
=========================================================================
*                         Final Report                                  *
=========================================================================
Final Results
RTL Top Level Output File Name     : experiment_2_1.ngr
Top Level Output File Name         : experiment_2_1
Output Format                      : NGC
Optimization Goal                  : Speed
Keep Hierarchy                     : No

Design Statistics
# IOs                              : 2

Cell Usage :
# BELS                             : 114
#      GND                         : 1
#      INV                         : 2
#      LUT1                        : 31
#      LUT4                        : 8
#      MUXCY                       : 39
#      VCC                         : 1
#      XORCY                       : 32
# FlipFlops/Latches                : 33
#      FDE                         : 1
#      FDR                         : 32
# Clock Buffers                    : 1
#      BUFGP                       : 1
# IO Buffers                       : 1
#      OBUF                        : 1
```

تعداد ورودی و خروجی ها (مجموعا)

یک بافر برای کلاک

مرحله۱-۴:

کد UCF این طراحی را جهت پیاده سازی سخت افزاری بر روی برد آزمایشگاه بنویسید.

```
NET "CLK_IN" LOC = p80 ;
NET "CLK_OUT" LOC = p93 ;
```

بخش دوم: راه انداز سون سگمنت

مرحله ۲-۱:

۱- تحقیق کنید تا چه فرکانسی تغییرات آن با چشم قابل تشخیص نمی‌باشد و دلیل آن را بیان کنید.

تغییرات فرکانس با چشم قابل تشخیص زمانی نیست که فرکانس به حدی بالا باشد که فریم‌های تصویر از توانایی پردازش چشم و مغز انسان فراتر برود. برای بیشتر افراد، این حدود ۶۰ هرتز است. بالاتر از این فرکانس، تصاویر به نظر پیوسته می‌آیند و تغییرات به‌سختی قابل تشخیص می‌باشند.

چشم انسان به‌طور طبیعی می‌تواند فرکانس‌های پایین‌تر را تشخیص دهد، اما در فرکانس‌های بالاتر، پردازش اطلاعات با تأخیر روبه‌رو می‌شود و تصاویر مداوم به نظر می‌رسند.

۲- کد مربوط به این آزمایش را نوشته و جزئیات آن را توضیح دهید.

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

پکیج STD_LOGIC_1164 از کتابخانه IEEE به منظور استفاده از تایپ STD_LOGIC

بقیه پکیج ها برای انجام محاسبات

```
entity experiment_2_2 is
    Port ( CLK : in  STD_LOGIC;
           DATA_IN : in  STD_LOGIC_VECTOR (7 downto 0);
           SEG_SEL : out  STD_LOGIC_VECTOR (3 downto 0);
           SEG_OUT : out  STD_LOGIC_VECTOR (7 downto 0));
end experiment_2_2;
```

سیگنال های ورودی و خروجی

```
architecture Behavioral of experiment_2_2 is

  signal digit_select : STD_LOGIC := '0';
    signal digit_ones : STD_LOGIC_VECTOR(3 downto 0);
    signal digit_tens : STD_LOGIC_VECTOR(3 downto 0);
    signal digit_out : STD_LOGIC_VECTOR(3 downto 0);
begin
    digit_ones <= DATA_IN(3 downto 0);
    digit_tens <= DATA_IN(7 downto 4);
```

تعریف سیگنال هایی به منظور جدا کردن یکان و دهگان عدد ورودی و نیز یک سیگنال برای سوییچ کردن بین دو سون سگمنت

```
    process(CLK)
    begin
        if rising_edge(CLK) then
            if digit_select = '0' then
                SEG_SEL <= "0001";
                digit_out <= digit_ones;
                digit_select <= '1';
            else
                SEG_SEL <= "0010";
                digit_out<= digit_tens;
                digit_select <= '0';
            end if;
        end if;
    end process;
```

با هر لبه کلاک بین نمایش یکان و دهگان عدد مد نظر سوییچ میکنیم

```
  WITH digit_out SELECT
  SEG_OUT <= "11000000" WHEN "0000" ,
                      "11111001" WHEN "0001" ,
                      "10100100" WHEN "0010" ,
                      "10110000" WHEN "0011" ,
                      "10011001" WHEN "0100" ,
                      "10010010" WHEN "0101" ,
                      "10000010" WHEN "0110" ,
                      "11111000" WHEN "0111" ,
                      "10000000" WHEN "1000" ,
                      "10010000" WHEN "1001" ,
                      "10001000" WHEN "1010" ,
```

این قسمت از کد عینا مشابه آزمایش اول می باشدو لذا توضیحی داده نمی شود.

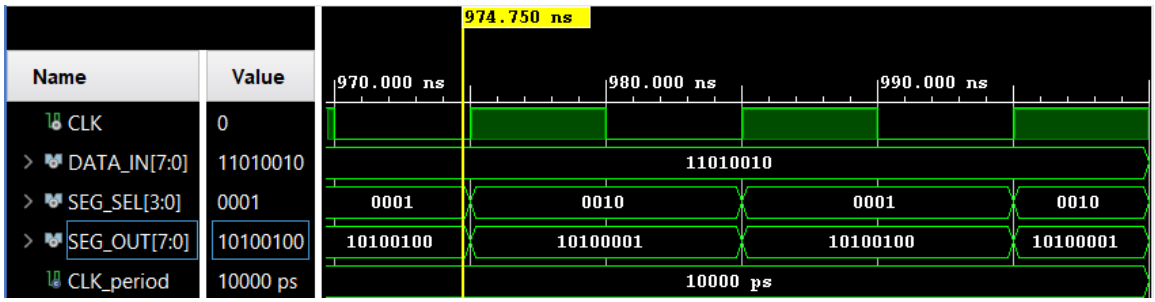
```
                "10000011" WHEN "1011" ,
                "11100110" WHEN "1100" ,
                "10100001" WHEN "1101" ,
                "10000110" WHEN "1110" ,
                "10001110" WHEN "1111" ,
                "00000000" WHEN others ;

end Behavioral;
```

مرحله۲-۲:

با نوشتن یک Test bench در نرم افزار ISE پاسخ شبیه سازی را در حالتی که عدد ورودی 11010010 باشد، نمایش دهید.

```
        DATA_IN <= "00000000" , "11010001" AFTER 40 ns , "11010010" AFTER 80 ns ;
-- Clock process definitions
CLK_process :process
begin
        CLK <= '0';
        wait for CLK_period/2;
        CLK <= '1';
        wait for CLK_period/2;
end process;
```



مرحله۲-۳:

نتیجه ی گزارش سنتز را برای این طراحی ارائه دهید و دلیلی برای استفاده شدن این مقدار از منابع FPGA را گزارش کنید.

```
=========================================================================
Advanced HDL Synthesis Report

Macro Statistics
# ROMs                                                 : 1
 16x8-bit ROM                                          : 1
# Registers                                            : 9
 Flip-Flops                                            : 9
=========================================================================
*                        Final Report                                   *
=========================================================================
Final Results
RTL Top Level Output File Name     : experiment_2_2.ngr
Top Level Output File Name         : experiment_2_2
Output Format                      : NGC
Optimization Goal                  : Speed
Keep Hierarchy                     : No

Design Statistics
# IOs                              : 21

Cell Usage :
# BELS                             : 13
#      GND                         : 1
#      LUT3                        : 4
#      LUT4                        : 7
#      VCC                         : 1
# FlipFlops/Latches                : 6
#      FD                          : 5
#      FDR                         : 1
# Clock Buffers                    : 1
#      BUFGP                       : 1
# IO Buffers                       : 20
#      IBUF                        : 8
#      OBUF                        : 12
=========================================================================
```

یک رام به دلیل نیاز به ذخیره سازی مقادیر باینری معادل برای روشن شدن سون سگمنت و نمایش اعداد

ما در قسمت پراسس، به نحوی سیگنال ها را assign کردیم که ۹ بیت سر راهشان فلیپ فلاپ قرار گرفت که در زیر این سیگنال ها مشاهده میکنیم:

```
SEG_SEL
digit_out
digit_select
```

مجموعا ۲۱ ورودی/خروجی

مرحله۲-۴:

کد UCFاین طراحی را جهت پیاده سازی سخت افزاری بر روی برد آزمایشگاه بنویسید.

```
NET "CLK" LOC = p181 ;

NET "DATA_IN[7]" LOC = p24 ;
NET "DATA_IN[6]" LOC = p26 ;
NET "DATA_IN[5]" LOC = p27 ;
NET "DATA_IN[4]" LOC = p28 ;
NET "DATA_IN[3]" LOC = p29 ;
NET "DATA_IN[2]" LOC = p31 ;
NET "DATA_IN[1]" LOC = p33 ;
NET "DATA_IN[0]" LOC = p34 ;

NET "SEG_SEL[3]" LOC = p130 ;
NET "SEG_SEL[2]" LOC = p128 ;
NET "SEG_SEL[1]" LOC = p126 ;
NET "SEG_SEL[0]" LOC = p125 ;

NET "SEG_OUT[7]" LOC = p131 ;
NET "SEG_OUT[6]" LOC = p132 ;
NET "SEG_OUT[5]" LOC = p133 ;
NET "SEG_OUT[4]" LOC = p135 ;
NET "SEG_OUT[3]" LOC = p137 ;
NET "SEG_OUT[2]" LOC = p138 ;
NET "SEG_OUT[1]" LOC = p139 ;
NET "SEG_OUT[0]" LOC = p140 ;
```

بخش سوم: بالا/پایین شمار دو رقمی مبنای ۱۶

مرحله۳-۱:

۱- با توجه به ورودیها چند حالت ممکن پیش بیاید، آنها را ذکر کنید.

اگر en صفر باشد، شمارشی انجام نمی‌شود.

اگر en و updown هر دو ۱ باشند، شمارش به سمت بالا انجام می‌شود.

اگر en یک ولی updown صفر باشد، شمارش به سمت پایین انجام می شود.

۲- کد مربوط به این آزمایش را نوشته و جزئیات آن را توضیح دهید.

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

پکیج STD_LOGIC_1164 از کتابخانه IEEE به منظور استفاده از تایپ STD_LOGIC

بقیه پکیج ها برای انجام محاسبات

```
entity experiment_2_3 is
    Port ( CLK : in  STD_LOGIC;
           ENABLE : in  STD_LOGIC;
           UP_DOWN : in  STD_LOGIC;
           COUNTER_OUT : out  STD_LOGIC_VECTOR (7 downto 0));
end experiment_2_3;
```

سیگنال های ورودی و خروجی

```
architecture Behavioral of experiment_2_3 is

    signal counter : STD_LOGIC_VECTOR(7 downto 0) := "00000000";

begin

    process(CLK)
    begin
        if rising_edge(CLK) then
            if ENABLE = '1' then
                if UP_DOWN = '1' then counter <= counter + 1;
                else counter <= counter - 1;
                end if;
            end if;
        end if;
    end process;

    COUNTER_OUT <= counter;
end Behavioral;
```

بسیار ساده است، به طوری که با هر لبه کلاک، درصورتی که En فعال باشد، اگر در حالت بالا شمار باشیم، یک واحد افزایش وگرنه یک واحد کاهش میدهیم.

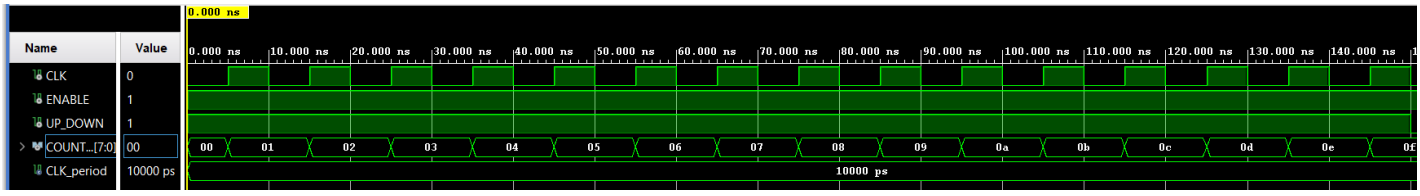
مرحله۳-۲:

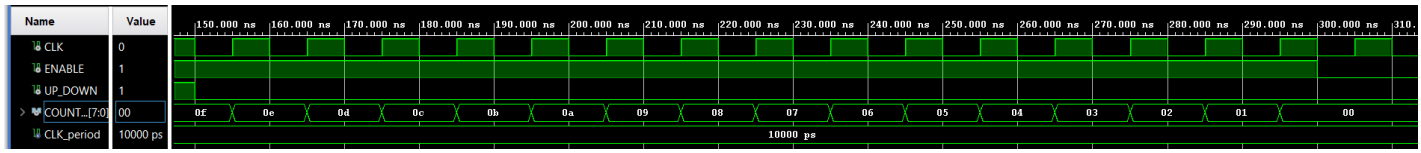
بانوشتن یکTest bench در نرم افزارISE پاسخ شبیه سازی را با توجه به شرایط زیر نمایش دهید.

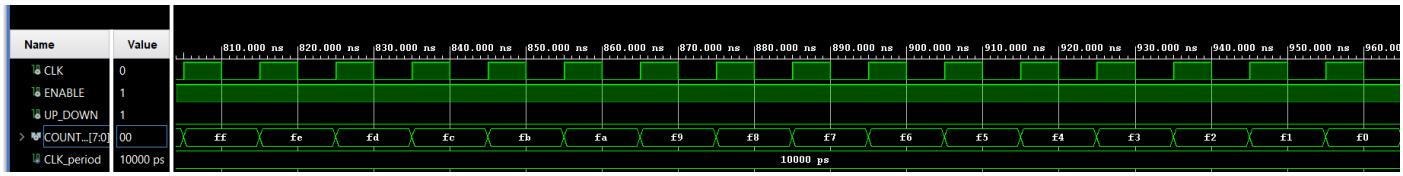
به ازای کلاک ورودی با فرکانس دلخواه، ابتدا شمارنده در حالت بالا شمار باشد و از ۰۰تا FFشروع به شمارش کند و سپس با تغییر ورودیUp/Downدر حالت پایین شمار قرار گرفته و تا F0بشمارد.

```
-- Clock process definitions
CLK_process :process
begin
		CLK <= '0';
		wait for CLK_period/2;
		CLK <= '1';
		wait for CLK_period/2;
end process;


-- Stimulus process
stim_proc: process
begin
     wait for 150 ns;
     UP_DOWN <= '0';
     wait for 150 ns;
     ENABLE <= '0';
     wait for 500 ns;
     ENABLE <= '1';
     wait;
end process;
```







مرحله۳-۳:

گزارش سنتز با دلیل.

```
=========================================================================
Advanced HDL Synthesis Report

Macro Statistics
# Counters                                             : 1
 8-bit updown counter                                  : 1
```

اصلا هدف کد یک شمارنده بالا/پایین هست که خب اینجا هم یک دونه ۸ بیتی از آن استخراج شده که طبیعی است

```
Design Statistics
# IOs                              : 11

Cell Usage :
# BELS                             : 24
#      INV                         : 1
#      LUT2                        : 8
#      MUXCY                       : 7
#      XORCY                       : 8
# FlipFlops/Latches                : 8
#      FDE                         : 8
# Clock Buffers                    : 1
#      BUFGP                       : 1
# IO Buffers                       : 10
#      IBUF                        : 2
#      OBUF                        : 8
=========================================================================
```

مجموعا ۱۱ خروجی و ورودی داریم

که ۲ ورودی

۸ خروجی

۱ کلاک

مرحله۳-۴:

کد UCF

```
NET "CLK" LOC = p181 ;

NET "ENABLE" LOC = p12 ;
NET "UP_DOWN" LOC = p11 ;

NET "COUNTER_OUT[7]" LOC = p102 ;
NET "COUNTER_OUT[6]" LOC = p101 ;
NET "COUNTER_OUT[5]" LOC = p100 ;
NET "COUNTER_OUT[4]" LOC = p97 ;
NET "COUNTER_OUT[3]" LOC = p96 ;
NET "COUNTER_OUT[2]" LOC = p95 ;
NET "COUNTER_OUT[1]" LOC = p94 ;
NET "COUNTER_OUT[0]" LOC = p93 ;
```