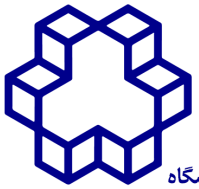
بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ



دانشگاه
خواجه نصیرالدین طوسی

نیمسال اول تحصیلی

1402-1403

تهیه کننده: رامین توکلی

درس: FPGA

شماره دانشجویی: ۹۹۲۵۰۶۳

استاد درس: دکتر حسینی‌نژآد

تمرین شماره یک

دانشکده مهندسی برق

سوال ۱) مداری طراحی نمایید تا یک عدد۸ بیتی را به عنوان ورودی دریافت نماید و مشخص نماید که تعداد یک های آن زوج یا فرد است. اگر تعداد یک ها زوج باشد عدد دریافتی را با ۴۷جمع نماید و نصف مجموع را در پورت خروجی قرار دهد. در غیر این دو برابر تفاضل عدد دریافتی و عدد ۳۷ را به پورت خروجی ارسال نماید.

راهنمایی: برای تعیین زوج یا فرد بودن تعداد یک ها می توانید از گیت xor استفاده نمایید.

کد بخش سنتز مدار:

```
10  library IEEE;
11  use IEEE.STD_LOGIC_1164.ALL;
12  use IEEE.numeric_std.all;
13
14  entity Even_Odd is
15      Port ( input : in  STD_LOGIC_VECTOR (7 downto 0);
16             output : out  SIGNED (8 downto 0));
17  end Even_Odd;
18
19  architecture Behavioral of Even_Odd is
20
21  signal tmp: STD_LOGIC;
22  signal s1: STD_LOGIC_VECTOR(8 downto 0);
23  signal s2: STD_LOGIC_VECTOR(8 downto 0);
24
25  begin
26  tmp <= input(0)xor input(1)xor input(2)xor
27  input(3)xor input(4)xor input(5)xor input(6)xor input(7);
28
29  s1 <= std_logic_vector((unsigned('0' & input) + 47)/2);
30  s2 <= std_logic_vector((signed('0' & input) - 37) + (signed('0' & input) - 37));
31
32
33  output <= signed(s1) when tmp ='0' else
34  signed(s2);
35
36  end Behavioral;
```

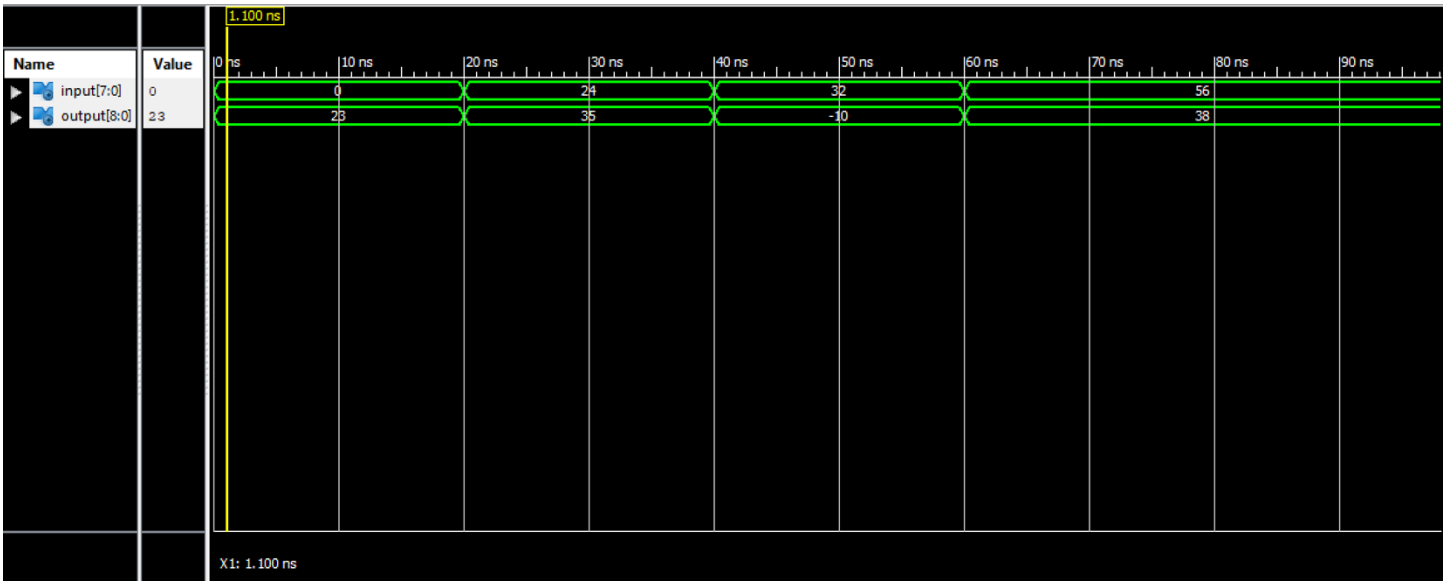
اندازه پورت‌های ورودی و خروجی به ترتیب ۸ بیت و ۹ بیت در نظر گرفته شده است. به دلیل اینکه خروجی مدار ممکن است از ۸ بیت یعنی ۲۵۵ بیشتر شود.

نتیجه و بررسی سنتز مدار و المان‌های استخراج شده:

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

Synthesizing Unit <Even_Odd>.
    Related source file is "C:\xilinix\ise-project\FPGA_class\HW1\EX1\FPGA
    Found 9-bit adder for signal <s1> created at line 1241.
    Found 9-bit subtractor for signal <n0024> created at line 0.
    Summary:
        inferred   2 Adder/Subtractor(s).
        inferred   1 Multiplexer(s).
Unit <Even_Odd> synthesized.

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

Macro Statistics
# Adders/Subtractors                                   : 2
 9-bit adder                                           : 1
 9-bit subtractor                                      : 1
# Multiplexers                                         : 1
 9-bit 2-to-1 multiplexer                              : 1
# Xors                                                 : 1
 1-bit xor8                                            : 1

=========================================================================

=========================================================================
*                            Design Summary                             *
=========================================================================

Top Level Output File Name         : Even_Odd.ngc

Primitive and Black Box Usage:
------------------------------
# BELS                             : 16
#      LUT2                        : 1
#      LUT3                        : 3
#      LUT4                        : 2
#      LUT5                        : 3
#      LUT6                        : 7
# IO Buffers                       : 17
#      IBUF                        : 8
#      OBUF                        : 9

Device utilization summary:
---------------------------

Selected Device : 6slx4tqg144-3


Slice Logic Utilization:
 Number of Slice LUTs:                  16  out of   2400     0%
    Number used as Logic:               16  out of   2400     0%

Slice Logic Distribution:
 Number of LUT Flip Flop pairs used:    16
   Number with an unused Flip Flop:     16  out of     16   100%
   Number with an unused LUT:            0  out of     16     0%
   Number of fully used LUT-FF pairs:    0  out of     16     0%
   Number of unique control sets:        0

IO Utilization:
 Number of IOs:                         17
 Number of bonded IOBs:                 17  out of    102    16%  Activat
```

کد بخش تست بنچ و شبیه سازی مدار:

```
1   LIBRARY ieee;
2   USE ieee.std_logic_1164.ALL;
3   use IEEE.numeric_std.all;
4
5   ENTITY TB_Even_Odd IS
6   END TB_Even_Odd;
7
8   ARCHITECTURE behavior OF TB_Even_Odd IS
9
10
11
12      COMPONENT Even_Odd
13      PORT(
14           input : IN  std_logic_vector(7 downto 0);
15           output : OUT  SIGNED(8 downto 0)
16          );
17      END COMPONENT;
18
19     --Inputs
20     signal input : std_logic_vector(7 downto 0) := (others => '0');
21
22     --Outputs
23     signal output : SIGNED(8 downto 0);
24
25  BEGIN
26
27     -- Instantiate the Unit Under Test (UUT)
28     uut: Even_Odd PORT MAP (
29             input => input,
30             output => output
31           );
32
33  input<= "00011000" after 20 ns, "00100000" after 40 ns, "00111000" after 60ns;
34
35  END;
36
```



همانطور که مشاهده می‌شود به ازای ورودی ۰ خروجی باید ۲۳ شود، به ازای ورودی ۲۴ خروجی خروجی باید ۳۵ شود، به ازای ورودی ۳۲ خروجی باید ۱۰- شود و به ازای ورودی ۵۶ خروجی باید ۳۸ شود که نتایج شبیه سازی همین نتایج را نشان می‌دهد که نتیجه می‌گیریم طراحی صحیح بوده و مدار درست کار می‌کند.

سوال ۲) یک stopwatch طراحی و شبیه سازی کنید.

این طراحی دارای یک ورودی کلاک، یک ورودی ریست آسنکرون برای صفر کردن زمان، یک ورودی استارت برای شروع زمان گیری، سه خروجی صدم ثانیه، ثانیه و دقیقه خواهد بود. اندازه پریود کلاک در شبیه سازی می تواند برای سرعت بخشیدن به کار دلخواه و کوچک باشد.

کد بخش سنتز مدار:

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_arith.ALL;
use IEEE.STD_LOGIC_unsigned.ALL;

entity Stopwatch is
    Port ( clk : in  STD_LOGIC;
           reset : in  STD_LOGIC;
           start : in  STD_LOGIC;
           min : out  STD_LOGIC_VECTOR (6 downto 0);
           sec : out STD_LOGIC_VECTOR (6 downto 0);
           Hsec : out  STD_LOGIC_VECTOR (6 downto 0));
end Stopwatch;

architecture Behavioral of Stopwatch is

signal count1: std_logic_vector(6 downto 0):="0000000";
signal count2: std_logic_vector(6 downto 0):="0000000";
signal count3: std_logic_vector(6 downto 0):="0000000";

begin

process(clk, reset)
begin
if reset='1' then count1 <= "0000000"; count2 <= "0000000"; count3 <= "0000000";
elsif (clk' event and clk='1' and start = '1') then
   count1 <= count1 + "0000001";
   if (count1 = "1100011")then
   if (count2 = "0111011")then
   if (count3 = "0111011")then
      count1 <= "0000000";
      count2 <= "0000000";
      count3 <= "0000000";
   else
      count3 <= count3 + "0000001";
      count2 <= "0000000";
      count1 <= "0000000";
```

```
38      end if;
39      else
40         count1 <= "0000000";
41         count2 <= count2 + "0000001";
42      end if;
43      end if;
44
45
46    end if;
47    end process;
48    min <= count3;
49    sec <= count2;
50    Hsec <= count1;
51
52    end behavioral;
```

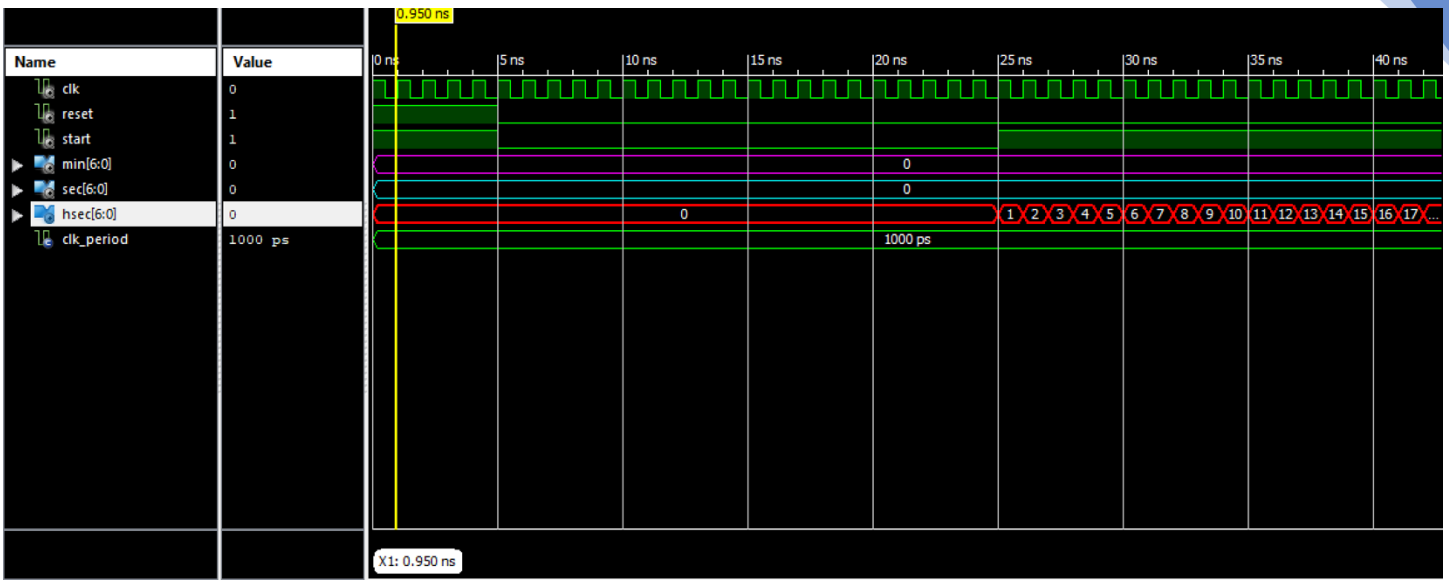
نتیجه و بررسی سنتز مدار و المان‌های استخراج شده:

```
=========================================================================

Synthesizing Unit <Stopwatch>.
    Related source file is "C:\xilinix\ise-project\FPGA_class\HW1\EX1\FPGA_EX1\Stopwatch.vhd"
    Found 7-bit register for signal <count2>.
    Found 7-bit register for signal <count3>.
    Found 7-bit register for signal <count1>.
    Found 7-bit adder for signal <count1[6]_GND_5_o_add_0_OUT> created at line 27.
    Found 7-bit adder for signal <count3[6]_GND_5_o_add_4_OUT> created at line 35.
    Found 7-bit adder for signal <count2[6]_GND_5_o_add_6_OUT> created at line 41.
    Summary:
        inferred   3 Adder/Subtractor(s).
        inferred  21 D-type flip-flop(s).
        inferred   3 Multiplexer(s).
Unit <Stopwatch> synthesized.

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

Macro Statistics
# Adders/Subtractors                                   : 3
 7-bit adder                                           : 3
# Registers                                            : 3
 7-bit register                                        : 3
# Multiplexers                                         : 3
 7-bit 2-to-1 multiplexer                              : 3

=========================================================================
```

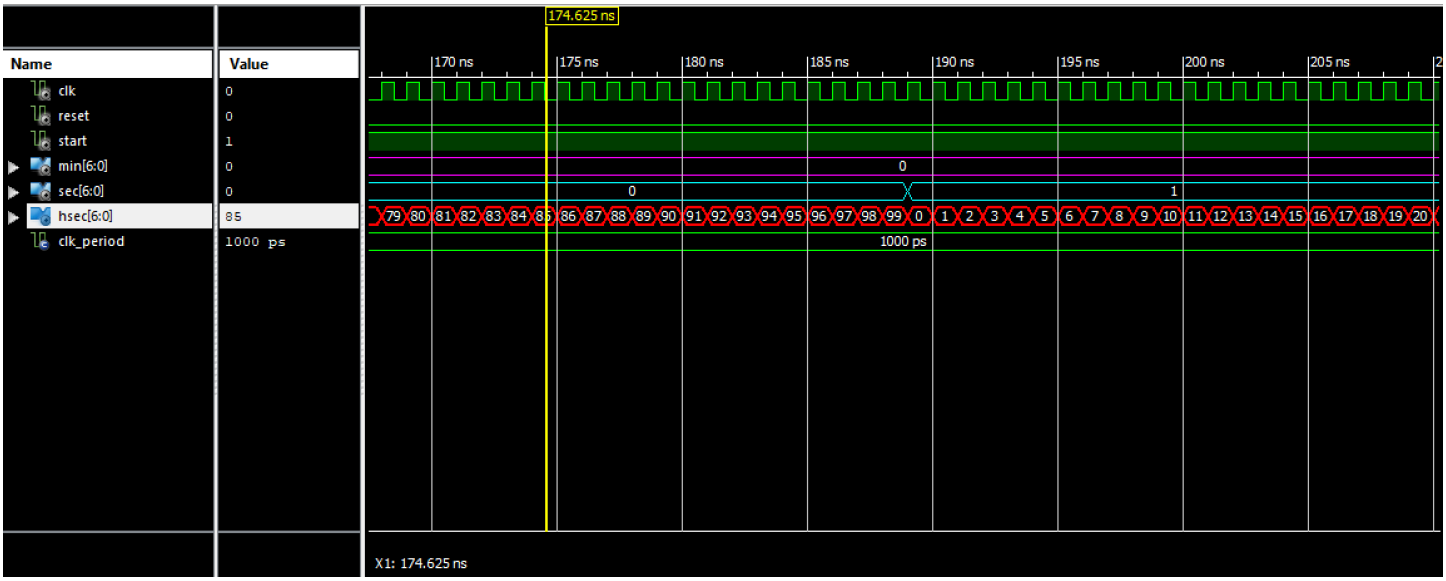
```
Top Level Output File Name         : Stopwatch.ngc

Primitive and Black Box Usage:
------------------------------
# BELS                             : 41
#      GND                         : 1
#      INV                         : 1
#      LUT2                        : 6
#      LUT3                        : 4
#      LUT4                        : 4
#      LUT5                        : 5
#      LUT6                        : 19
#      MUXF7                       : 1
# FlipFlops/Latches                : 19
#      FDCE                        : 19
# Clock Buffers                    : 1
#      BUFGP                       : 1
# IO Buffers                       : 23
#      IBUF                        : 2
#      OBUF                        : 21

Device utilization summary:
---------------------------

Selected Device : 6slx4tqg144-3


Slice Logic Utilization:
 Number of Slice Registers:              19  out of   4800      0%
 Number of Slice LUTs:                   39  out of   2400      1%
    Number used as Logic:                39  out of   2400      1%

Slice Logic Distribution:
 Number of LUT Flip Flop pairs used:     39
   Number with an unused Flip Flop:      20  out of     39     51%
   Number with an unused LUT:             0  out of     39      0%
   Number of fully used LUT-FF pairs:    19  out of     39     48%

   Number of unique control sets:         5

IO Utilization:
 Number of IOs:                          24
 Number of bonded IOBs:                  24  out of    102     23%

Specific Feature Utilization:
 Number of BUFG/BUFGCTRLs:               1  out of     16      6%
```

کد بخش تست بنچ و شبیه سازی مدار:

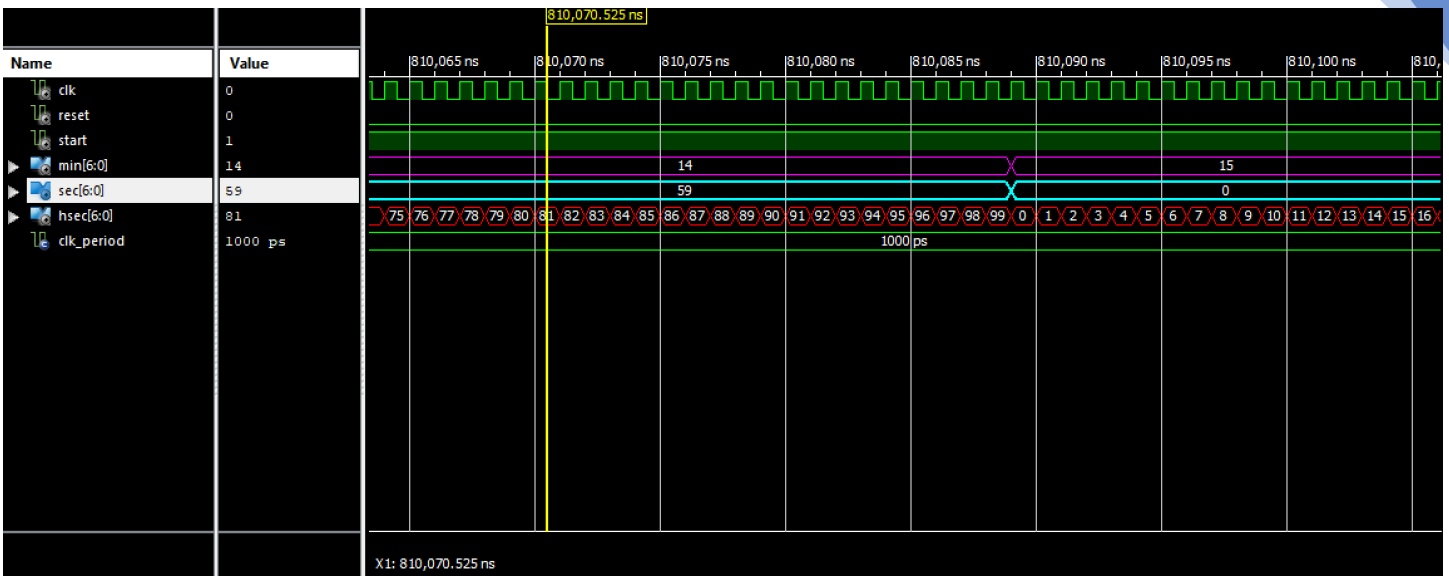
```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
use IEEE.STD_LOGIC_arith.ALL;
use IEEE.STD_LOGIC_unsigned.ALL;

ENTITY TB_Stopwatch IS
generic (
            clk_period : time := 1 ns -- per le tempistiche
        );
END TB_Stopwatch;

ARCHITECTURE behavior OF TB_Stopwatch IS

    -- Component Declaration for the Unit Under Test (UUT)

    COMPONENT Stopwatch
    PORT(
         clk : IN  std_logic;
         reset : IN  std_logic;
         start : IN  std_logic;
         min : OUT  std_logic_vector(6 downto 0);
         sec : OUT  std_logic_vector(6 downto 0);
         Hsec : OUT  std_logic_vector(6 downto 0)
        );
    END COMPONENT;

   --Inputs
   signal clk : std_logic := '1';
   signal reset : std_logic := '1';
   signal start : std_logic := '1';

   --Outputs
   signal min : std_logic_vector(6 downto 0);
   signal sec : std_logic_vector(6 downto 0);
   signal Hsec : std_logic_vector(6 downto 0);


BEGIN

   -- Instantiate the Unit Under Test (UUT)
   uut: Stopwatch PORT MAP (
          clk => clk,
          reset => reset,
          start => start,
          min => min,
          sec => sec,
          Hsec => Hsec
        );

clk <= NOT clk after clk_period/2;
start <= '0' after 5ns , '1' after 25 ns;
reset <= '0' after 5ns , '1' after 60 ns,'0' after 90 ns;


END;
```



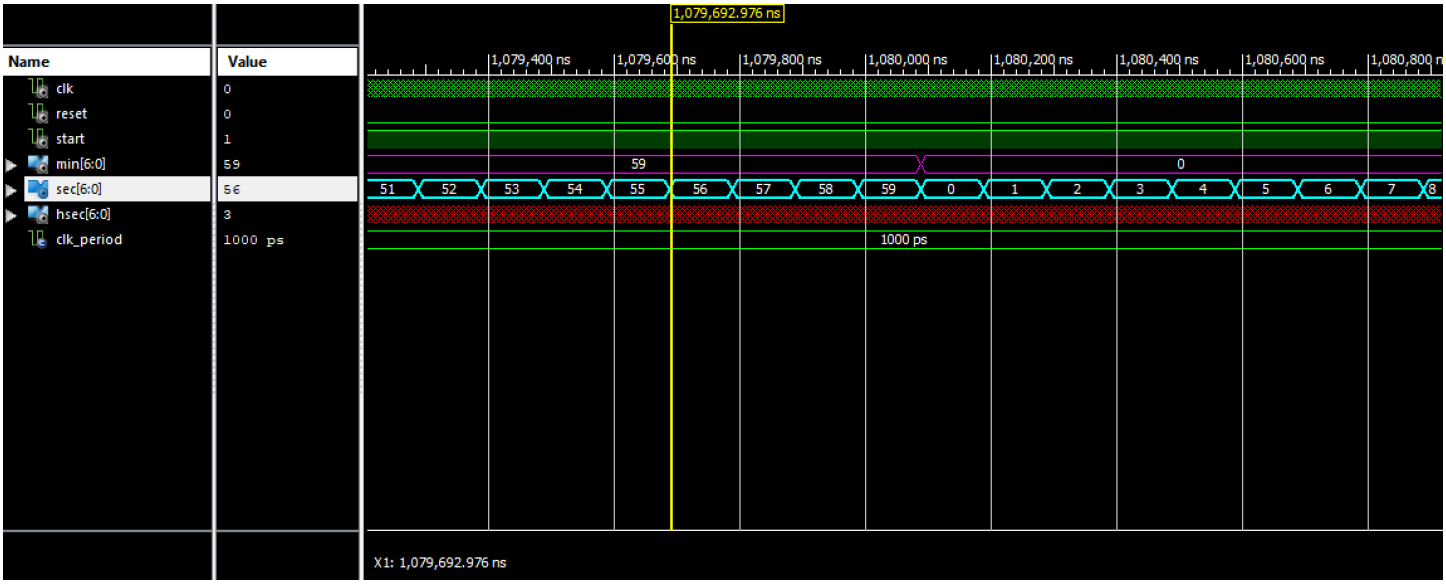
بعد از بررسی تست بنچ به سراغ نمودار زمانی ورودی و خروجی‌های مدار می‌رسیم که در بالا مشاهده میکنیم سیگنال start تا قبل از ۵ نانوثانیه فعال است ولی سیگنال reset هم فعال است برای همین خروجی صفر باقی می‌ماند. در زمان ۲۵ نانوثانیه سیگنال reset غیر فعال است و سیگنال start فعال است پس تایمر شروع به شمارش می‌کند.



برای بررسی صحت عملکر تایمر باید شمارش ثانیه شمار را بعد از شمارش صدم ثانیه شمار را نیز بررسی بکنیم. مشاهده می‌شود که در تصویر بالا بعد از اینکه صدم ثانیه شمار به مقدار ۹۹ رسید خروجی ثانیه شمار افزایش می‌یابد.



برای بررسی صحت عملکر تایمر باید شمارش دقیقه شمار را بعد از شمارش ثانیه شمار را نیز بررسی بکنیم. مشاهده می‌شود که در تصویر بالا بعد از اینکه ثانیه شمار به مقدار ۵۹ رسید خروجی دقیقه شمار افزایش می‌یابد.



همچنین برای بررسی صحت عملکر تایمر باید شمارش تایمر را بعد از اینکه شمارش کامل شد نیز بررسی بکنیم پس بعد از اینکه صدم ثانیه، ثانیه و دقیقه کامل شدند باید شمارش تایمر ما دوباره از مقدار صفر شروع شود.

مشاهده می‌شود که در تصویر بالا بعد از اینکه دقیقه شمار به مقدار ۵۹ رسید خروجی صفر می‌شود.

سوال ۳) یک تقسیم کننده فرکانسی باduty cycle مشخص طراحی و شبیه سازی کنید.

این طراحی شامل یک ورودی کلاک ، یک ورودی کنترل ۴ بیتی duty cycle و یک خروجی با فرکانس تقسیم شده است. در این طراحی ورودی کنترل ۴ بیتی می تواند مقادیری بین ۰ تا ۱۰ را دریافت کند. مقدار دریافتی در ۱۰ ضرب شده و بزرگی duty cycleما را تعیین می کند. برای مثال اگر به ورودی کنترل عدد ۴ داده شود مدت زمان ۱بودن خروجی ۴۰ درصد از طول پریود خواهد بود و ۶۰ درصد دیگر ۰ خواهد شد.

در این تمرین فرکانس کلاک ورودی ۱۰۰ برابر فرکانس کلاک خروجی خواهد بود.

کد بخش سنتز مدار:

```
1  library IEEE;
2  use IEEE.STD_LOGIC_1164.ALL;
3  use IEEE.STD_LOGIC_arith.ALL;
4  use IEEE.STD_LOGIC_unsigned.ALL;
5
6  entity Freq_div is
7      Port ( clk : in  STD_LOGIC;
8             duty_cycle : in  STD_LOGIC_VECTOR (3 downto 0);
9             out_clk : out  STD_LOGIC);
10 end Freq_div;
11
12 architecture Behavioral of Freq_div is
13
14 signal dcycle: STD_LOGIC_VECTOR (7 downto 0);
15 signal count: integer range 0 to 255:= 0;
16 signal output: STD_LOGIC:= '0';
17
18 begin
19 dcycle <= ( duty_cycle) * "1010" when (duty_cycle < "1010");
20 --count <= 100 - (conv_integer(dcycle))
21
22 process(clk)
23 begin
24 if (clk' event and clk='0') then
25    count <= count + 1;
26    if( count = 99 and output = '1' ) then
27       count <= 0;
28       output <= '0';
29    elsif((99 - CONV_INTEGER(dcycle)) <= count ) then
30       output <= '1';
31
32    end if;
33 end if;
34
35 end process;
36 out_clk <= output;
37 end Behavioral;
```

نتیجه و بررسی سنتز مدار و المان‌های استخراج شده:

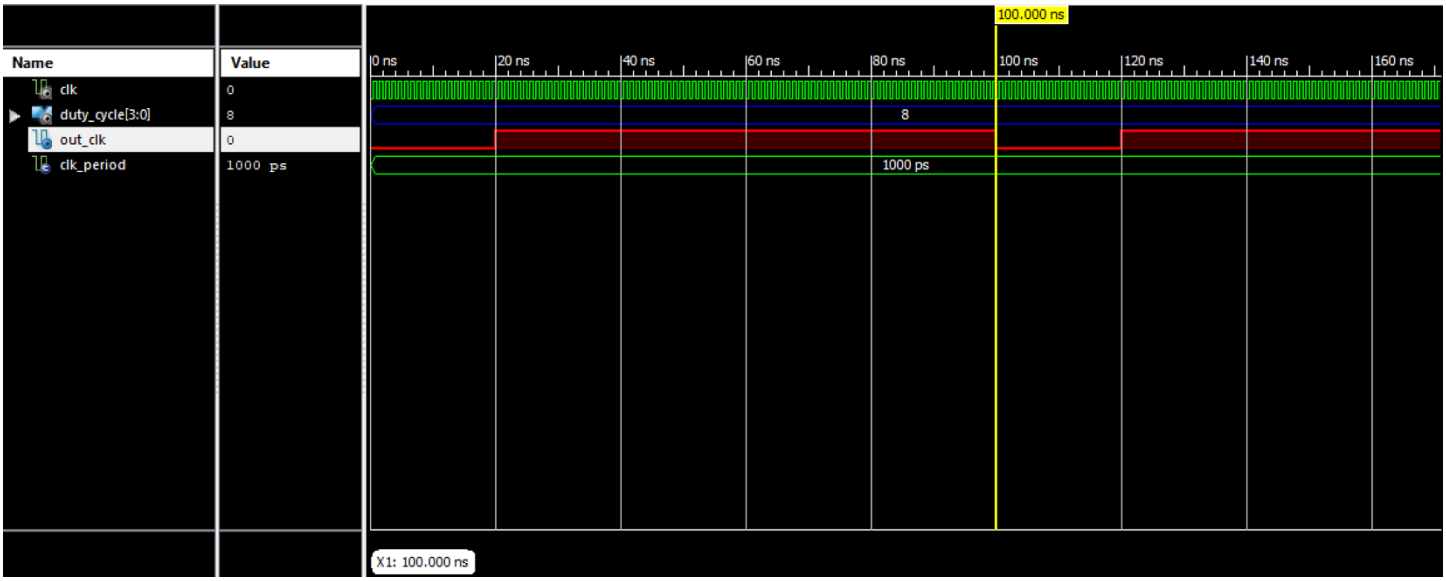
```
HDL Synthesis Report

Macro Statistics
# Multipliers                                      : 1
 4x4-bit multiplier                                : 1
# Adders/Subtractors                               : 2
 8-bit adder                                       : 1
 9-bit subtractor                                  : 1
# Registers                                        : 2
 1-bit register                                    : 1
 8-bit register                                    : 1
# Latches                                          : 8
 1-bit latch                                       : 8
# Comparators                                      : 2
 4-bit comparator greater                          : 1
 9-bit comparator greater                          : 1

*                            Design Summary                                  *
================================================================================

Top Level Output File Name         : Freq_div.ngc

Primitive and Black Box Usage:
------------------------------
# BELS                             : 52
#      GND                         : 1
#      INV                         : 2
#      LUT1                        : 7
#      LUT2                        : 11
#      LUT3                        : 1
#      LUT4                        : 7
#      LUT5                        : 3
#      LUT6                        : 4
#      MUXCY                       : 7
#      VCC                         : 1
#      XORCY                       : 8
# FlipFlops/Latches                : 16
#      FD                          : 8
#      FD_1                        : 1
#      LD                          : 7
# Clock Buffers                    : 1
#      BUFGP                       : 1
# IO Buffers                       : 5
#      IBUF                        : 4
#      OBUF                        : 1
```

```
Slice Logic Utilization:
 Number of Slice Registers:              16  out of   4800     0%
 Number of Slice LUTs:                   35  out of   2400     1%
    Number used as Logic:                35  out of   2400     1%

Slice Logic Distribution:
 Number of LUT Flip Flop pairs used:     37
   Number with an unused Flip Flop:      21  out of     37    56%
   Number with an unused LUT:             2  out of     37     5%
   Number of fully used LUT-FF pairs:    14  out of     37    37%
   Number of unique control sets:         3

IO Utilization:
 Number of IOs:                           6
 Number of bonded IOBs:                   6  out of    102     5%

Specific Feature Utilization:
 Number of BUFG/BUFGCTRLs:                1  out of     16     6%
```

کد بخش تست بنچ و شبیه سازی مدار:

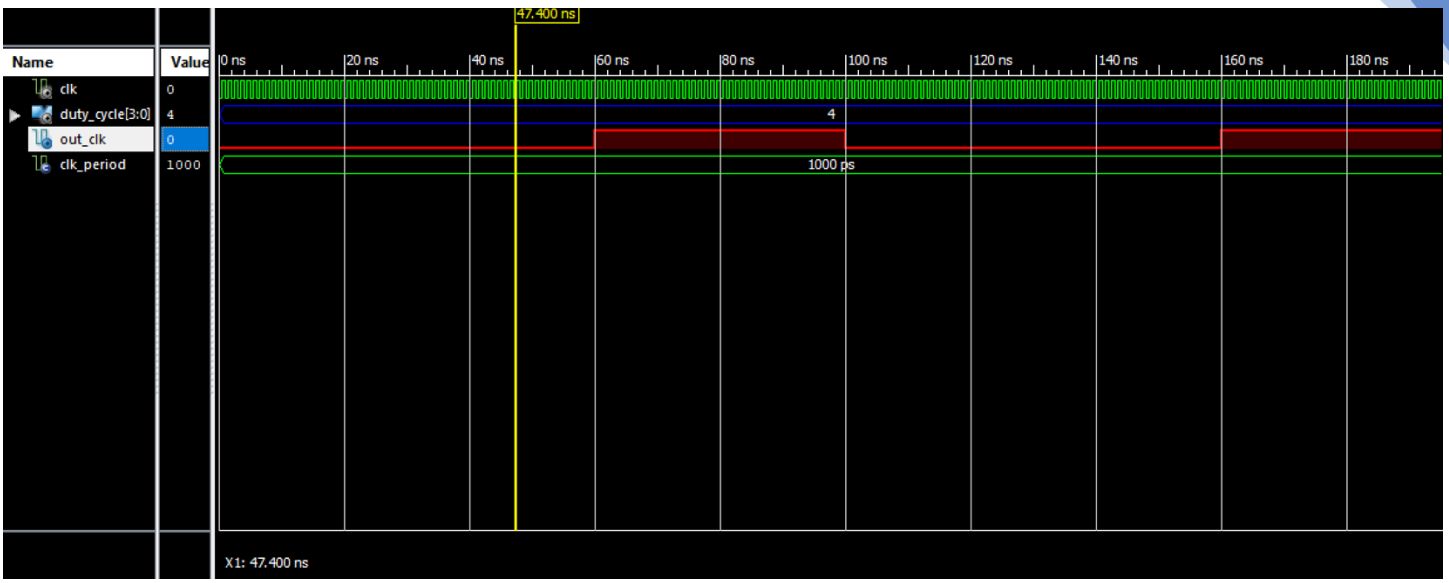
```
1  LIBRARY ieee;
2  USE ieee.std_logic_1164.ALL;
3  use IEEE.STD_LOGIC_arith.ALL;
4  use IEEE.STD_LOGIC_unsigned.ALL;
5
6  ENTITY TB_Freq_div IS
7  generic (
8              clk_period : time := 1 ns -- per le tempistiche
9          );
10 END TB_Freq_div;
11
12 ARCHITECTURE behavior OF TB_Freq_div IS
13
14     -- Component Declaration for the Unit Under Test (UUT)
15
16     COMPONENT Freq_div
17     PORT(
18          clk : IN  std_logic;
19          duty_cycle : IN  std_logic_vector(3 downto 0);
20          out_clk : OUT  std_logic
21         );
22     END COMPONENT;
23
24
25    --Inputs
26    signal clk : std_logic := '0';
27    signal duty_cycle : std_logic_vector(3 downto 0) := (others => '0');
28
29    --Outputs
30    signal out_clk : std_logic;
31
32 BEGIN
33
```

```
34     -- Instantiate the Unit Under Test (UUT)
35     uut: Freq_div PORT MAP (
36            clk => clk,
37            duty_cycle => duty_cycle,
38            out_clk => out_clk
39          );
40
41
42  clk <= NOT clk after clk_period/2;
43  duty_cycle <= "1000"  ;
44
45  END;
46
```



همانطور که مشاهده می‌شود کلاک ورودی فرکانسی ۱۰۰ برابر فرکانس خروجی دارد و هر با وارد کردن عدد ۸، کلاک خروجی ۸۰ نانوثانیه یک می‌ماند و ۲۰ ثانیه صفر. در تصویر زیر تقسیم کننده دیگری را بررسی می‌کنیم.

```
clk <= NOT clk after clk_period/2;
duty_cycle <= "0100"  ;
```



همانطور که مشاهده می‌شود کلاک ورودی فرکانسی ۱۰۰ برابر فرکانس خروجی دارد و هر با وارد کردن عدد ۴، کلاک خروجی ۴۰ نانوثانیه یک می‌ماند و ۶۰ ثانیه صفر.