آنائیس گل‌بوداغیانس ۴۰۱۲۲۱۱۳

تمرین سری اول FPGA

سوال ۱)

در تشخیص زوج یا فرد بودن تعداد یک‌ها، آمدیم پرارزش‌ترین بیت را با بیت کناری XOR کردیم و نتیجه را با بیت بعدی، به XOR کردن ادامه دادیم. اگر خروجی یک می‌شد یعنی تعداد یک‌ها فرد است و اگر صفر می‌شد، یعنی تعداد زوج است.

کد ماژول:

```
library IEEE;
USE ieee.std_logic_1164.all ;
USE ieee.std_logic_unsigned.all ;
use IEEE.NUMERIC_STD.ALL;

entity evenodd is
Port (a:in STD_LOGIC_VECTOR (7 downto 0);
b: out STD_LOGIC_VECTOR (7 downto 0));
end evenodd;

architecture Behavioral of evenodd is
signal q: std_logic;
signal y, z, w, s, two: std_logic_vector(7 downto 0);
```

```
begin

q <=((((((((a(7) xor a(6))xor a(5))xor a(4))xor a(3))xor a(2))xor a(1))xor a(0)));

y <= a + x"2F";

z <= a-x"25";

w <= (y(0)&y(7)&y(6)&y(5)&y(4)&y(3)&y(2)&y(1));

s <= (z(6)&z(5)&z(4)&z(3)&z(2)&z(1)&z(0)&z(7));

with q select

b <= (s) when ('1'),

 (w) when ('0'),

 (x"00") when others;



end Behavioral;
```

برای ضرب‌در ۲ یا تقسیم بر ۲ کردن از شیفت دادن بیت‌ها استفاده کردیم.

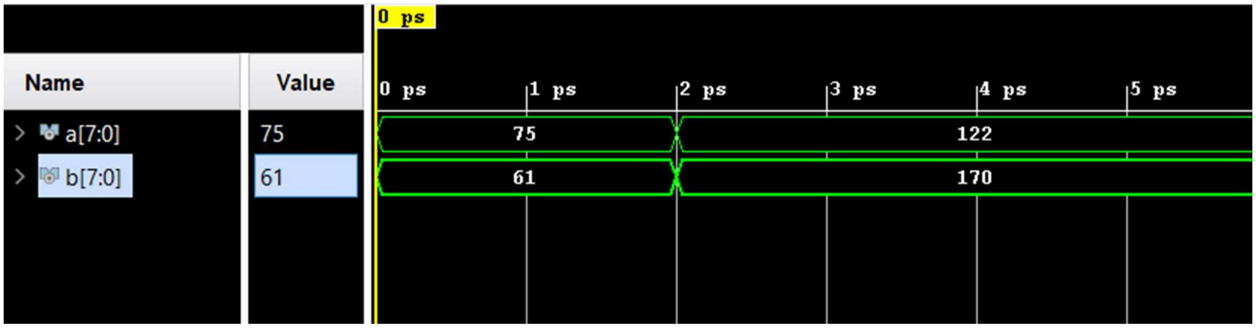
کد تست‌بنچ:

```
library IEEE;

use IEEE.STD_LOGIC_1164.ALL;

use IEEE.NUMERIC_STD.ALL;
```

```
entity evenodd_tb is
end evenodd_tb;

architecture Behavioral of evenodd_tb is
component evenodd
PORT(
a: in std_logic_vector(7 downto 0);
b: out STD_LOGIC_VECTOR (8 downto 0)
);
end component;
signal a: std_logic_vector(7 downto 0):= (others => '0');
signal b: std_logic_vector(7 downto 0):= (others => '0');
begin
process
begin

a <= x"4B";
wait for 2 ps;
a <= x"7A";
```

```
wait;
end process;

U: entity work.evenodd(behavioral) PORT MAP(a=>a, b=>b);

end Behavioral;
```

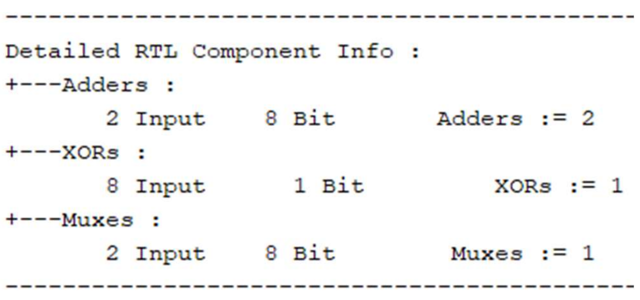
نتیجه شبیه‌سازی:



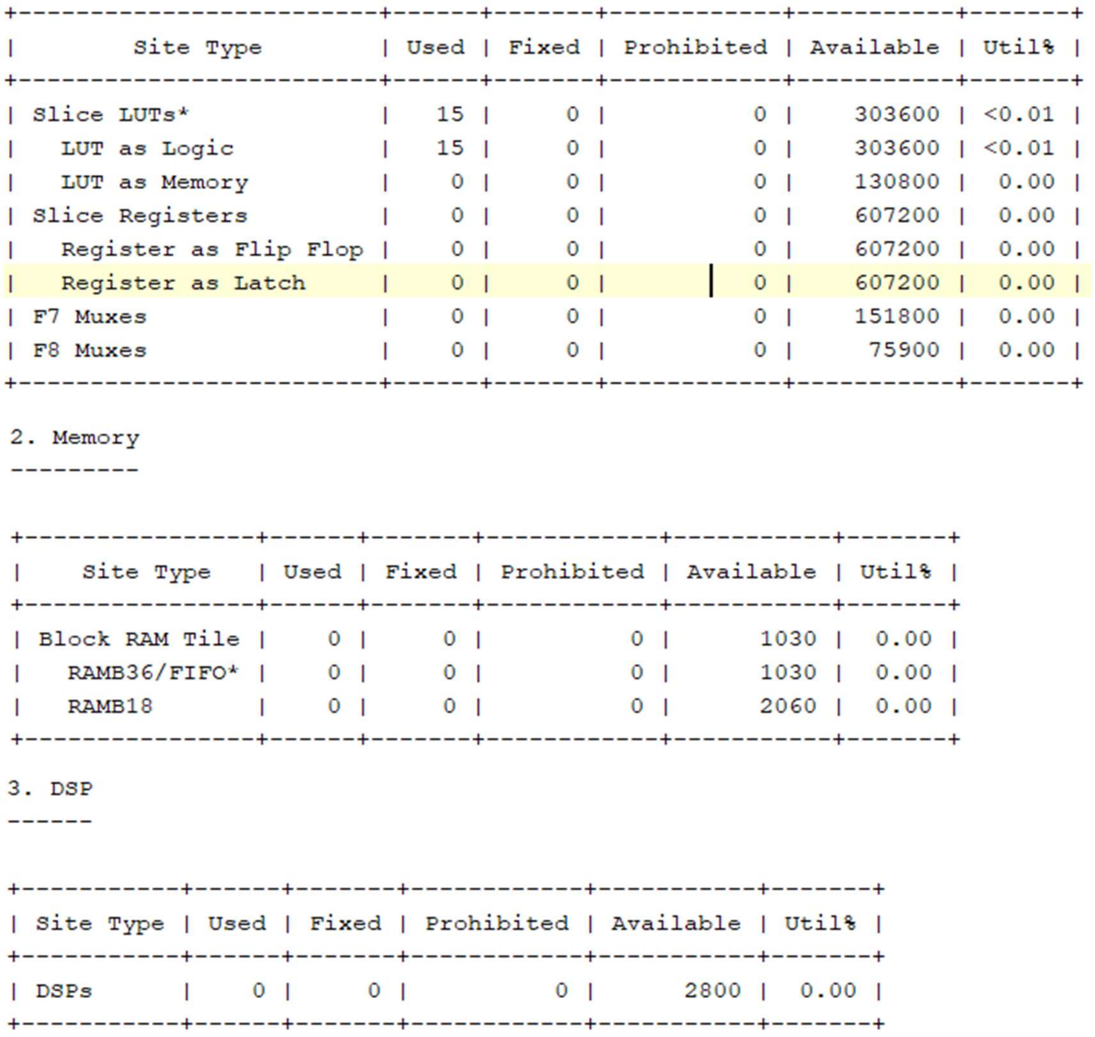
این یک شبیه‌سازی محدود بوده و اگر می‌خواستیم تمامی جوانب این سوال را بررسی کنیم، باید محاسبات floating point یا اعشاری نیز انجام می‌دادیم. از عملگر تقسیم نمی‌توانستیم استفاده کنیم زیرا قابل سنتز نیست.

در صورت سرریز کردن می‌توانستیم بیت پورت b را بیش‌تر کنیم.

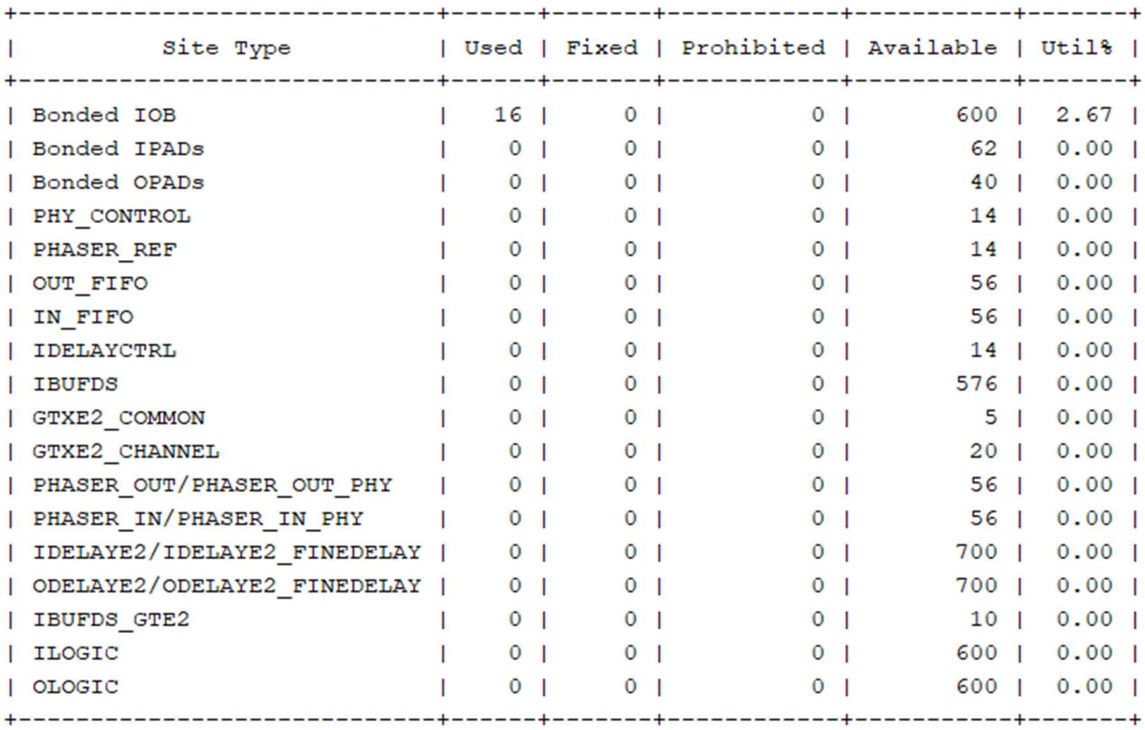
بخش‌های مورد نظر گزارش سنتز:



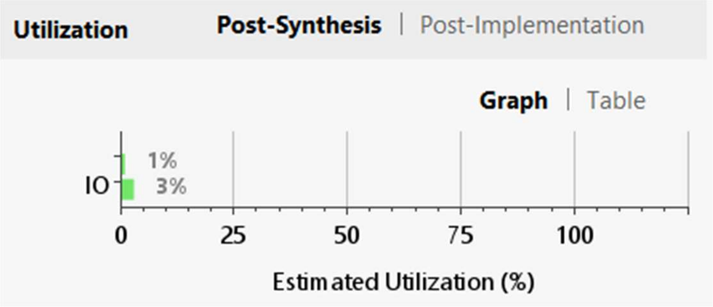
```
---------------------------------------------
Detailed RTL Component Info :
+---Adders :
       2 Input    8 Bit       Adders := 2
+---XORs :
       8 Input      1 Bit         XORs := 1
+---Muxes :
       2 Input    8 Bit        Muxes := 1
---------------------------------------------
```



```
+-------------------------+------+-------+------------+-----------+-------+
|        Site Type        | Used | Fixed | Prohibited | Available | Util% |
+-------------------------+------+-------+------------+-----------+-------+
| Slice LUTs*             |   15 |     0 |          0 |    303600 | <0.01 |
|   LUT as Logic          |   15 |     0 |          0 |    303600 | <0.01 |
|   LUT as Memory         |    0 |     0 |          0 |    130800 |  0.00 |
| Slice Registers         |    0 |     0 |          0 |    607200 |  0.00 |
|   Register as Flip Flop |    0 |     0 |          0 |    607200 |  0.00 |
|   Register as Latch     |    0 |     0 |          0 |    607200 |  0.00 |
| F7 Muxes                |    0 |     0 |          0 |    151800 |  0.00 |
| F8 Muxes                |    0 |     0 |          0 |     75900 |  0.00 |
+-------------------------+------+-------+------------+-----------+-------+

2. Memory
---------

+----------------+------+-------+------------+-----------+-------+
|    Site Type   | Used | Fixed | Prohibited | Available | Util% |
+----------------+------+-------+------------+-----------+-------+
| Block RAM Tile |    0 |     0 |          0 |      1030 |  0.00 |
|   RAMB36/FIFO* |    0 |     0 |          0 |      1030 |  0.00 |
|   RAMB18       |    0 |     0 |          0 |      2060 |  0.00 |
+----------------+------+-------+------------+-----------+-------+

3. DSP
------

+-----------+------+-------+------------+-----------+-------+
| Site Type | Used | Fixed | Prohibited | Available | Util% |
+-----------+------+-------+------------+-----------+-------+
| DSPs      |    0 |     0 |          0 |      2800 |  0.00 |
+-----------+------+-------+------------+-----------+-------+
```



```
+-----------------------------+------+-------+------------+-----------+-------+
|          Site Type          | Used | Fixed | Prohibited | Available | Util% |
+-----------------------------+------+-------+------------+-----------+-------+
| Bonded IOB                  |   16 |     0 |          0 |       600 |  2.67 |
| Bonded IPADs                |    0 |     0 |          0 |        62 |  0.00 |
| Bonded OPADs                |    0 |     0 |          0 |        40 |  0.00 |
| PHY_CONTROL                 |    0 |     0 |          0 |        14 |  0.00 |
| PHASER_REF                  |    0 |     0 |          0 |        14 |  0.00 |
| OUT_FIFO                    |    0 |     0 |          0 |        56 |  0.00 |
| IN_FIFO                     |    0 |     0 |          0 |        56 |  0.00 |
| IDELAYCTRL                  |    0 |     0 |          0 |        14 |  0.00 |
| IBUFDS                      |    0 |     0 |          0 |       576 |  0.00 |
| GTXE2_COMMON                |    0 |     0 |          0 |         5 |  0.00 |
| GTXE2_CHANNEL               |    0 |     0 |          0 |        20 |  0.00 |
| PHASER_OUT/PHASER_OUT_PHY   |    0 |     0 |          0 |        56 |  0.00 |
| PHASER_IN/PHASER_IN_PHY     |    0 |     0 |          0 |        56 |  0.00 |
| IDELAYE2/IDELAYE2_FINEDELAY |    0 |     0 |          0 |       700 |  0.00 |
| ODELAYE2/ODELAYE2_FINEDELAY |    0 |     0 |          0 |       700 |  0.00 |
| IBUFDS_GTE2                 |    0 |     0 |          0 |        10 |  0.00 |
| ILOGIC                      |    0 |     0 |          0 |       600 |  0.00 |
| OLOGIC                      |    0 |     0 |          0 |       600 |  0.00 |
+-----------------------------+------+-------+------------+-----------+-------+
```



سوال دوم)

طراحی stopwatch

کد اصلی ماژول:

```
library IEEE;

use IEEE.STD_LOGIC_1164.ALL;

use ieee.std_logic_unsigned.all;
```

```
entity stopwatch is
    Port (
        clk     : in STD_LOGIC;
        rst     : in STD_LOGIC;
        start   : in STD_LOGIC;
        sec_100 : out STD_LOGIC_VECTOR (7 downto 0);
        sec     : out STD_LOGIC_VECTOR (7 downto 0);
        min     : out STD_LOGIC_VECTOR (7 downto 0)
    );
end stopwatch;

architecture Behavioral of stopwatch is
    signal s_100 : STD_LOGIC_VECTOR (7 downto 0) := (others => '0');
    signal s      : STD_LOGIC_VECTOR (7 downto 0) := (others => '0');
    signal m      : STD_LOGIC_VECTOR (7 downto 0) := (others => '0');
begin

process(clk, rst)
begin
    if rst = '1' then
        s_100 <= (others => '0');
        s     <= (others => '0');
        m      <= (others => '0');
    elsif rising_edge(clk) then
        if start = '1' then
            s_100 <= s_100 + 1;
            if s_100 = 99 then
                s_100 <= (others => '0');
                s <= s + 1;
                if s = 59 then
                    s <= (others => '0');
                    m <= m + 1;
                    if m = 59 then
                        m <= (others => '0');
                    end if;
                end if;
            end if;
        end if;
    end if;
end process;
```

```
sec_100 <= s_100;

sec     <= s;

min     <= m;


end Behavioral;
```

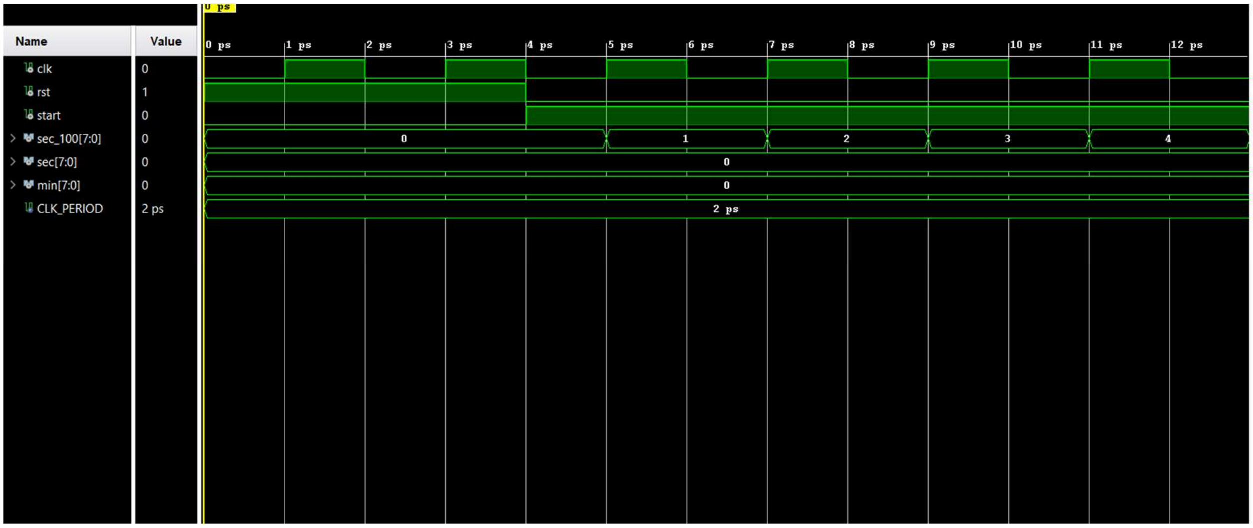
کد تست‌بنچ:

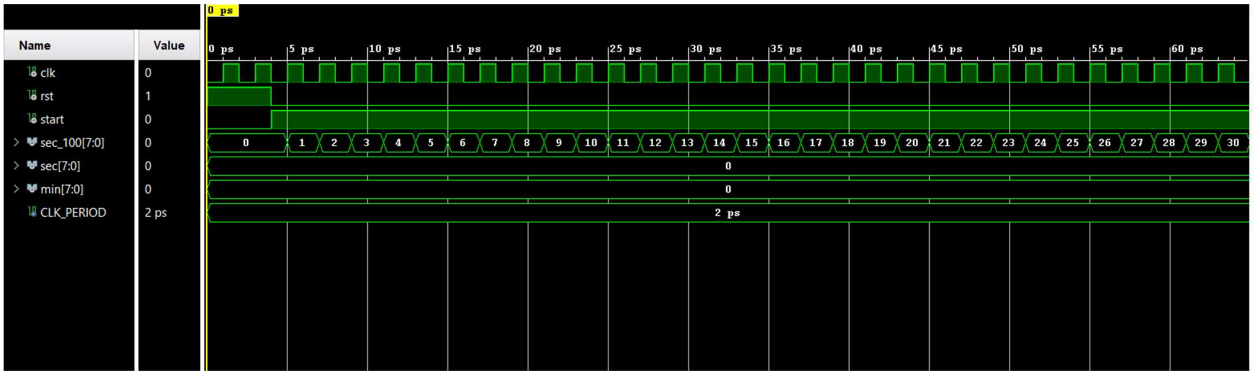
```
library IEEE;

use IEEE.STD_LOGIC_1164.ALL;

use IEEE.NUMERIC_STD.ALL;

use ieee.std_logic_arith.all;

use ieee.std_logic_unsigned.all;


entity stopwatch_tb is

end stopwatch_tb;


architecture Behavioral of stopwatch_tb is

component stopwatch

    Port ( clk : in STD_LOGIC;

           rst : in STD_LOGIC;

           start : in STD_LOGIC;
```

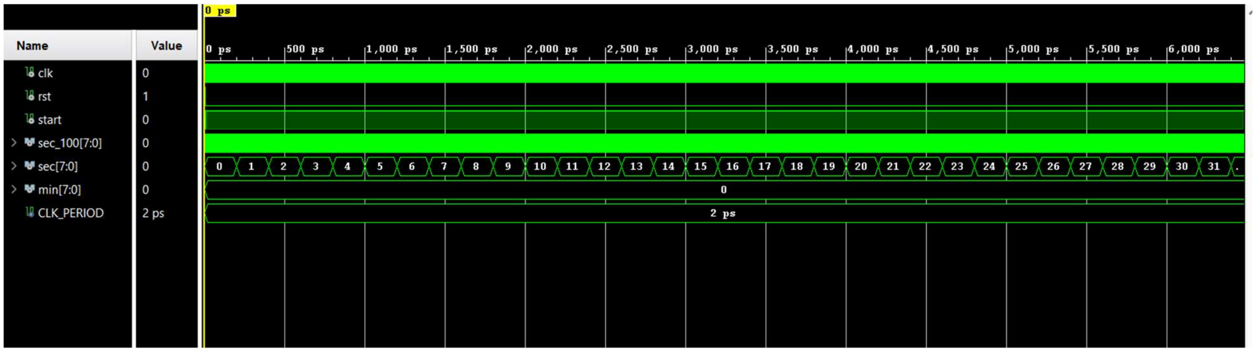
```
         sec_100 : out STD_LOGIC_VECTOR (7 downto 0);
         sec : out STD_LOGIC_VECTOR (7 downto 0);
         min : out STD_LOGIC_VECTOR (7 downto 0));
end component;
signal clk :STD_LOGIC:= '0';
signal rst, start :STD_LOGIC:= '0';
signal sec_100: std_logic_vector(7 downto 0);

signal sec: std_logic_vector(7 downto 0);
signal min: std_logic_vector(7 downto 0);

constant CLK_PERIOD : time := 2 ps;
begin




   -- Clock generation
   clk <= not clk after CLK_PERIOD/2;

process
begin
```

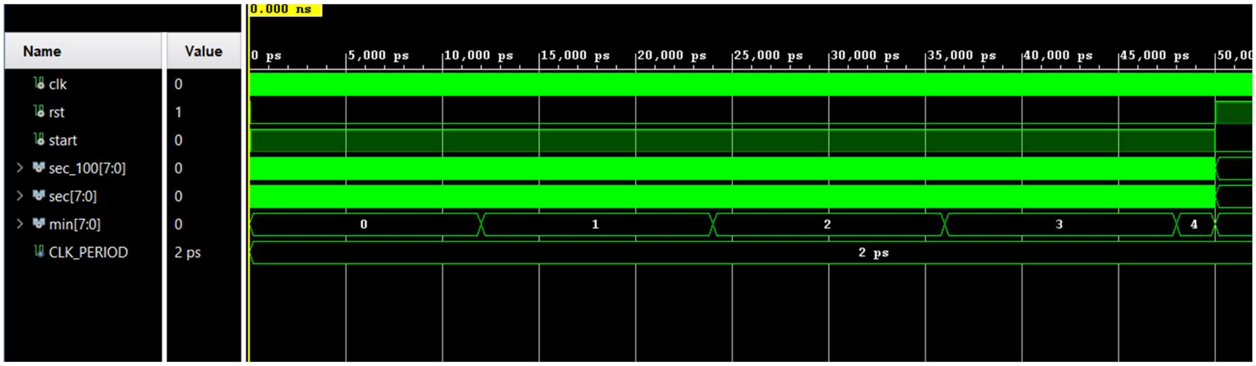
```
rst <= '1';

wait for 4 ps;

rst <= '0';

start <= '1';

wait for 50000 ps;

rst <='1';

start <= '0';



wait;

end process;


U: entity work.stopwatch(behavioral) PORT MAP( clk=>clk,

            rst =>rst,

            start => start,

            sec_100 => sec_100,

            sec => sec,

            min => min);


end Behavioral;
```

نتایج شبیه‌سازی:

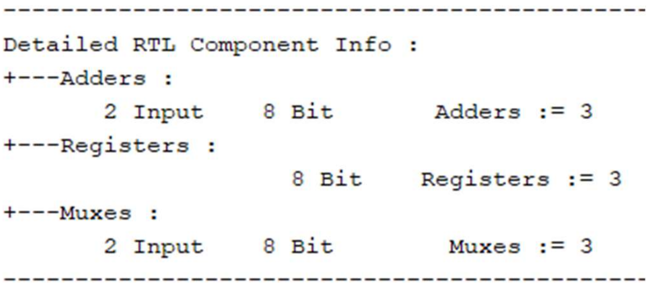




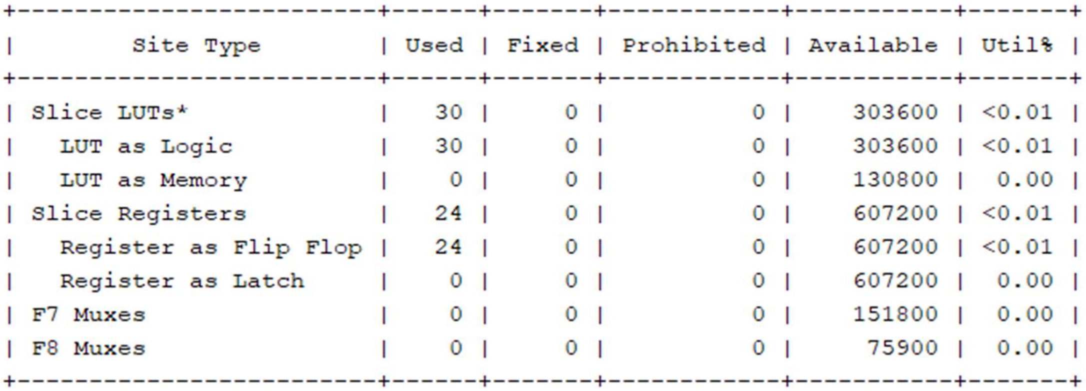




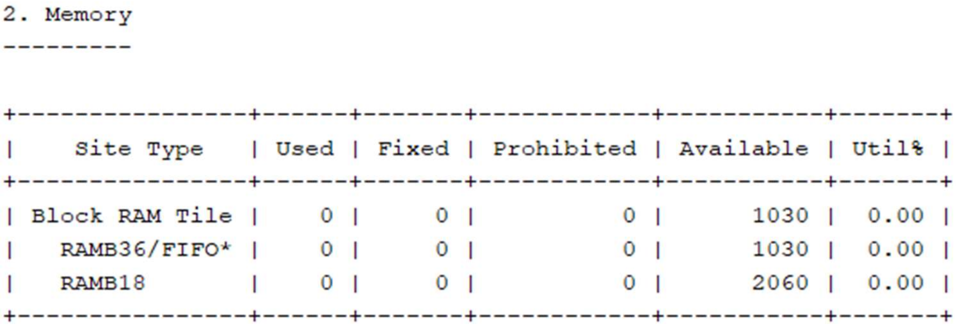
نتایج مورد نظر سنتز:



```
---------------------------------------------
Detailed RTL Component Info :
+---Adders :
       2 Input      8 Bit       Adders := 3
+---Registers :
                    8 Bit     Registers := 3
+---Muxes :
       2 Input      8 Bit        Muxes := 3
---------------------------------------------
```



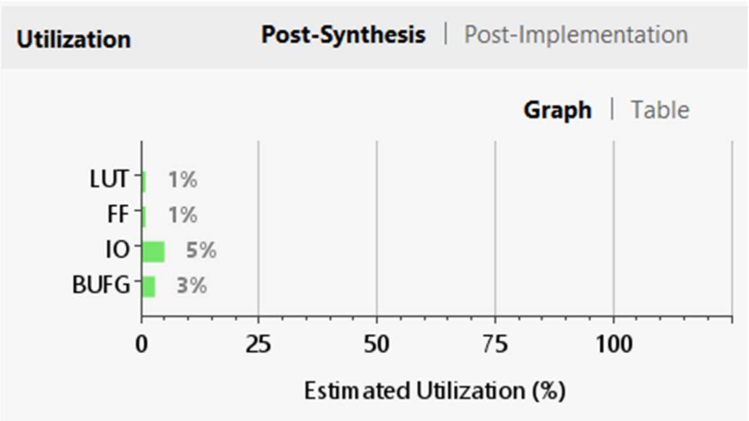
```
+-------------------------+------+-------+------------+-----------+-------+
|        Site Type        | Used | Fixed | Prohibited | Available | Util% |
+-------------------------+------+-------+------------+-----------+-------+
| Slice LUTs*             |   30 |     0 |          0 |    303600 | <0.01 |
|   LUT as Logic          |   30 |     0 |          0 |    303600 | <0.01 |
|   LUT as Memory         |    0 |     0 |          0 |    130800 |  0.00 |
| Slice Registers         |   24 |     0 |          0 |    607200 | <0.01 |
|   Register as Flip Flop |   24 |     0 |          0 |    607200 | <0.01 |
|   Register as Latch     |    0 |     0 |          0 |    607200 |  0.00 |
| F7 Muxes                |    0 |     0 |          0 |    151800 |  0.00 |
| F8 Muxes                |    0 |     0 |          0 |     75900 |  0.00 |
+-------------------------+------+-------+------------+-----------+-------+
```



```
2. Memory
---------

+-----------------+------+-------+------------+-----------+-------+
|    Site Type    | Used | Fixed | Prohibited | Available | Util% |
+-----------------+------+-------+------------+-----------+-------+
| Block RAM Tile  |    0 |     0 |          0 |      1030 |  0.00 |
|   RAMB36/FIFO*  |    0 |     0 |          0 |      1030 |  0.00 |
|   RAMB18        |    0 |     0 |          0 |      2060 |  0.00 |
+-----------------+------+-------+------------+-----------+-------+
+-----------------------------+------+-------+------------+-----------+-------+
|          Site Type          | Used | Fixed | Prohibited | Available | Util% |
+-----------------------------+------+-------+------------+-----------+-------+
| Bonded IOB                  |   27 |     0 |          0 |       600 |  4.50 |
| Bonded IPADs                |    0 |     0 |          0 |        62 |  0.00 |
| Bonded OPADs                |    0 |     0 |          0 |        40 |  0.00 |
| PHY_CONTROL                 |    0 |     0 |          0 |        14 |  0.00 |
| PHASER_REF                  |    0 |     0 |          0 |        14 |  0.00 |
| OUT_FIFO                    |    0 |     0 |          0 |        56 |  0.00 |
| IN_FIFO                     |    0 |     0 |          0 |        56 |  0.00 |
| IDELAYCTRL                  |    0 |     0 |          0 |        14 |  0.00 |
| IBUFDS                      |    0 |     0 |          0 |       576 |  0.00 |
| GTXE2_COMMON                |    0 |     0 |          0 |         5 |  0.00 |
| GTXE2_CHANNEL               |    0 |     0 |          0 |        20 |  0.00 |
| PHASER_OUT/PHASER_OUT_PHY   |    0 |     0 |          0 |        56 |  0.00 |
| PHASER_IN/PHASER_IN_PHY     |    0 |     0 |          0 |        56 |  0.00 |
| IDELAYE2/IDELAYE2_FINEDELAY |    0 |     0 |          0 |       700 |  0.00 |
| ODELAYE2/ODELAYE2_FINEDELAY |    0 |     0 |          0 |       700 |  0.00 |
| IBUFDS_GTE2                 |    0 |     0 |          0 |        10 |  0.00 |
| ILOGIC                      |    0 |     0 |          0 |       600 |  0.00 |
| OLOGIC                      |    0 |     0 |          0 |       600 |  0.00 |
+-----------------------------+------+-------+------------+-----------+-------+
```

```
5. Clocking
-----------

+------------+------+-------+------------+-----------+-------+
|  Site Type | Used | Fixed | Prohibited | Available | Util% |
+------------+------+-------+------------+-----------+-------+
| BUFGCTRL   |    1 |     0 |          0 |        32 |  3.13 |
| BUFIO      |    0 |     0 |          0 |        56 |  0.00 |
| MMCME2_ADV |    0 |     0 |          0 |        14 |  0.00 |
| PLLE2_ADV  |    0 |     0 |          0 |        14 |  0.00 |
| BUFMRCE    |    0 |     0 |          0 |        28 |  0.00 |
| BUFHCE     |    0 |     0 |          0 |       168 |  0.00 |
| BUFR       |    0 |     0 |          0 |        56 |  0.00 |
+------------+------+-------+------------+-----------+-------+
```

```
+----------+------+---------------------+
| Ref Name | Used | Functional Category |
+----------+------+---------------------+
| OBUF     |   24 |                  IO |
| FDCE     |   24 |        Flop & Latch |
| LUT6     |   19 |                 LUT |
| LUT4     |    5 |                 LUT |
| LUT2     |    5 |                 LUT |
| LUT5     |    4 |                 LUT |
| LUT3     |    3 |                 LUT |
| IBUF     |    3 |                  IO |
| LUT1     |    2 |                 LUT |
| BUFG     |    1 |               Clock |
+----------+------+---------------------+
```



سوال سوم)

تقسیم‌کننده فرکانسی با duty cycle مشخص

(پورت‌های خروجی tenc و count برای کنترل شمارش است و می‌توانسیم کلاً آن‌ها را حذف کنیم و تنها به‌عنوان سیگنال داشته باشیم.)

کد اصلی ماژول:

```
library IEEE;

use IEEE.STD_LOGIC_1164.ALL;

use IEEE.std_logic_unsigned.all;

use IEEE.std_logic_arith.all;

--use IEEE.NUMERIC_STD.ALL;



entity duty is

    Port ( clk : in STD_LOGIC;

           c : in STD_LOGIC_VECTOR (3 downto 0);
```

```
        wave : out STD_LOGIC;
        count, tenc : out STD_LOGIC_VECTOR (7 downto 0));
end duty;

architecture Behavioral of duty is
signal cc, b, ten : std_logic_vector (7 downto 0) := x"00";
signal ww : std_logic :='0';
begin


process(clk)
begin
if clk='0' then
   if ten /=99 then
      if c/=0 then
         if cc/=((c)*x"A") then
            cc <= cc+'1';
            if c/=0 then
               ww <= '1';
            else
               ww<='0';
```

```
                end if;
            else
                ww <= '0';
            end if;
        else
            ww <= '0';
        end if;
        ten <= ten +'1';
    else
        ten <= x"00";
        cc <= x"00";
    end if;
end if;
end process;
count <= cc;
wave <= ww;
tenc <= ten;
end Behavioral;
```

کد تست‌بنچ:

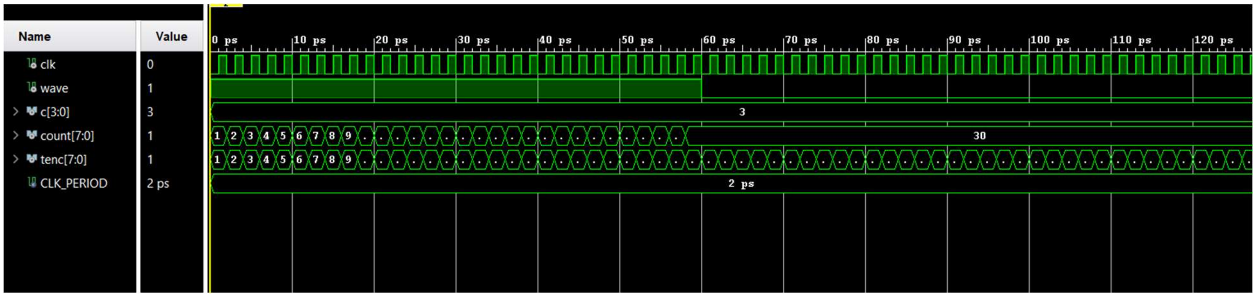
```
library IEEE;
```

```
use IEEE.STD_LOGIC_1164.ALL;

use IEEE.NUMERIC_STD.ALL;

use ieee.std_logic_arith.all;

use ieee.std_logic_unsigned.all;


entity duty_tb is

end duty_tb;


architecture Behavioral of duty_tb is

component duty

    Port ( clk : in STD_LOGIC;

           c : in STD_LOGIC_VECTOR (3 downto 0);

           wave : out STD_LOGIC;

           count, tenc : out STD_LOGIC_VECTOR (7 downto 0));

end component;

signal clk :STD_LOGIC:= '0';

signal wave :STD_LOGIC:= '0';

signal c: std_logic_vector(3 downto 0);

signal count, tenc :  STD_LOGIC_VECTOR (7 downto 0);
```

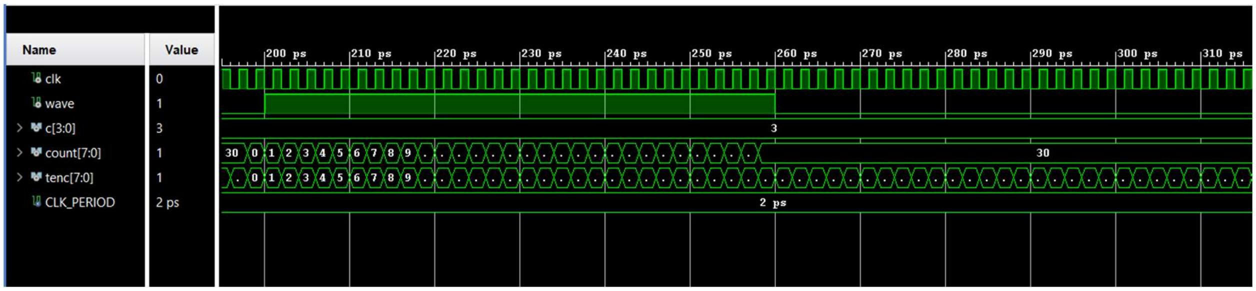
```
constant CLK_PERIOD : time := 2 ps;

begin



    -- Clock generation

    clk <= not clk after CLK_PERIOD/2;


process

begin

c <= x"3";

wait for 400 ps;

c <= x"1";

wait for 400 ps;

c <= x"7";

wait for 400 ps;

c<= x"A";



wait;

end process;
```

```
U: entity work.duty(behavioral) PORT MAP( clk=>clk,

          c =>c,

          wave => wave,

          tenc => tenc,

          count => count);


end Behavioral;
```

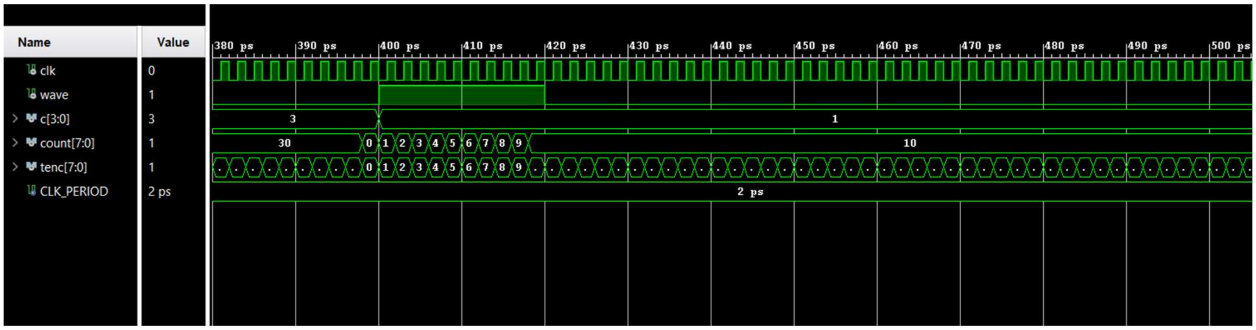
نتایج شبیه‌سازی:

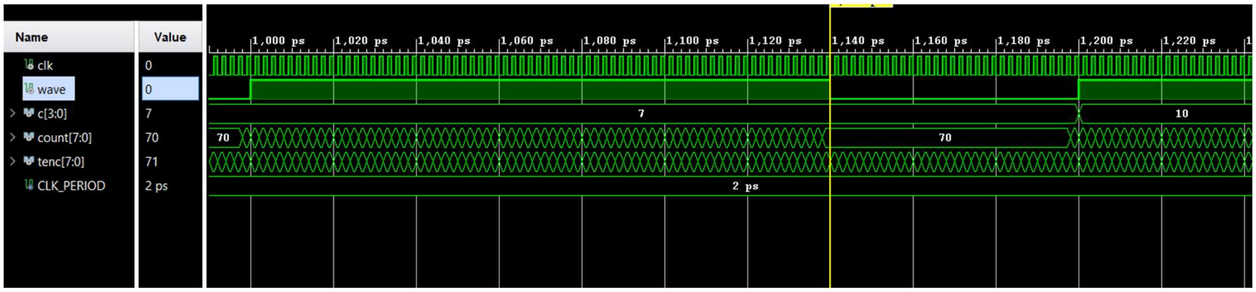


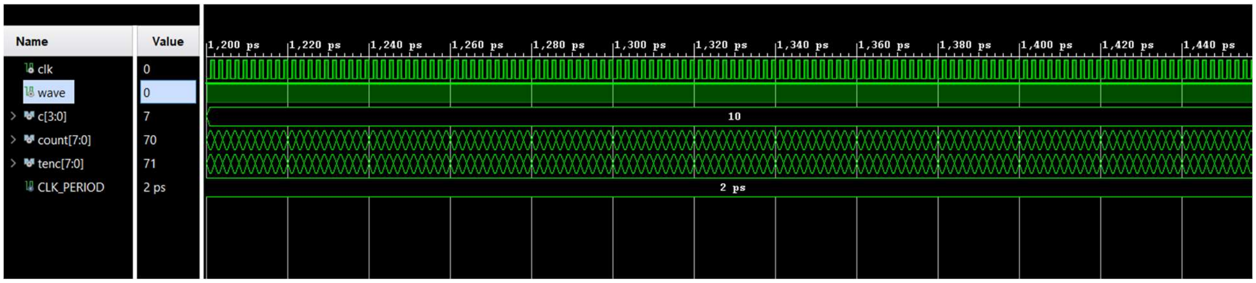
Name	Value
clk	0
wave	1
c[3:0]	3
count[7:0]	1
tenc[7:0]	1
CLK_PERIOD	2 ps



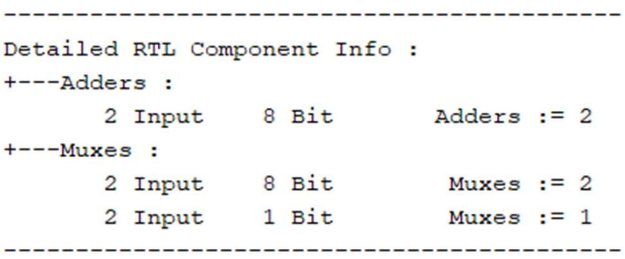
Name	Value
clk	0
wave	1
c[3:0]	3
count[7:0]	1
tenc[7:0]	1
CLK_PERIOD	2 ps







نتایج مورد نظر سنتز:



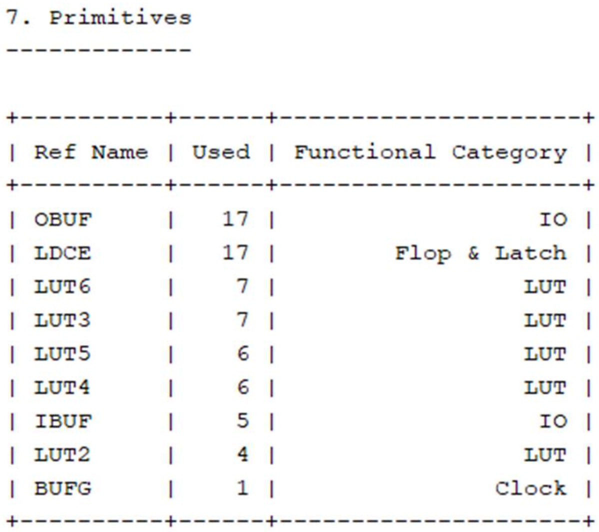
```
---------------------------------------------
Detailed RTL Component Info :
+---Adders :
	   2 Input      8 Bit       Adders := 2
+---Muxes :
	   2 Input      8 Bit        Muxes := 2
	   2 Input      1 Bit        Muxes := 1
---------------------------------------------
```

```
+-------------------------+------+-------+------------+-----------+-------+
|        Site Type        | Used | Fixed | Prohibited | Available | Util% |
+-------------------------+------+-------+------------+-----------+-------+
| Slice LUTs*             |   21 |     0 |          0 |    303600 | <0.01 |
|   LUT as Logic          |   21 |     0 |          0 |    303600 | <0.01 |
|   LUT as Memory         |    0 |     0 |          0 |    130800 |  0.00 |
| Slice Registers         |   17 |     0 |          0 |    607200 | <0.01 |
|   Register as Flip Flop |    0 |     0 |          0 |    607200 |  0.00 |
|   Register as Latch     |   17 |     0 |          0 |    607200 | <0.01 |
| F7 Muxes                |    0 |     0 |          0 |    151800 |  0.00 |
| F8 Muxes                |    0 |     0 |          0 |     75900 |  0.00 |
+-------------------------+------+-------+------------+-----------+-------+
```

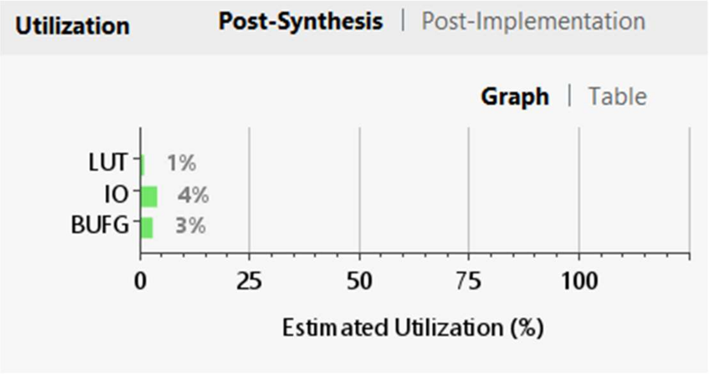
```
1.1 Summary of Registers by Type
--------------------------------

+-------+--------------+-------------+--------------+
| Total | Clock Enable | Synchronous | Asynchronous |
+-------+--------------+-------------+--------------+
| 0     |            _ |           - |            - |
| 0     |            _ |           - |          Set |
| 0     |            _ |           - |        Reset |
| 0     |            _ |         Set |            - |
| 0     |            _ |       Reset |            - |
| 0     |          Yes |           - |            - |
| 0     |          Yes |           - |          Set |
| 17    |          Yes |           - |        Reset |
| 0     |          Yes |         Set |            - |
| 0     |          Yes |       Reset |            - |
+-------+--------------+-------------+--------------+
```

```
2. Memory
---------

+-----------------+------+-------+------------+-----------+-------+
|    Site Type    | Used | Fixed | Prohibited | Available | Util% |
+-----------------+------+-------+------------+-----------+-------+
| Block RAM Tile  |    0 |     0 |          0 |      1030 |  0.00 |
|   RAMB36/FIFO*  |    0 |     0 |          0 |      1030 |  0.00 |
|   RAMB18        |    0 |     0 |          0 |      2060 |  0.00 |
+-----------------+------+-------+------------+-----------+-------+
+-----------------------------+------+-------+------------+-----------+-------+
|          Site Type          | Used | Fixed | Prohibited | Available | Util% |
+-----------------------------+------+-------+------------+-----------+-------+
| Bonded IOB                  |   22 |     0 |          0 |       600 |  3.67 |
| Bonded IPADs                |    0 |     0 |          0 |        62 |  0.00 |
| Bonded OPADs                |    0 |     0 |          0 |        40 |  0.00 |
| PHY_CONTROL                 |    0 |     0 |          0 |        14 |  0.00 |
| PHASER_REF                  |    0 |     0 |          0 |        14 |  0.00 |
| OUT_FIFO                    |    0 |     0 |          0 |        56 |  0.00 |
| IN_FIFO                     |    0 |     0 |          0 |        56 |  0.00 |
| IDELAYCTRL                  |    0 |     0 |          0 |        14 |  0.00 |
| IBUFDS                      |    0 |     0 |          0 |       576 |  0.00 |
| GTXE2_COMMON                |    0 |     0 |          0 |         5 |  0.00 |
| GTXE2_CHANNEL               |    0 |     0 |          0 |        20 |  0.00 |
| PHASER_OUT/PHASER_OUT_PHY   |    0 |     0 |          0 |        56 |  0.00 |
| PHASER_IN/PHASER_IN_PHY     |    0 |     0 |          0 |        56 |  0.00 |
| IDELAYE2/IDELAYE2_FINEDELAY |    0 |     0 |          0 |       700 |  0.00 |
| ODELAYE2/ODELAYE2_FINEDELAY |    0 |     0 |          0 |       700 |  0.00 |
| IBUFDS_GTE2                 |    0 |     0 |          0 |        10 |  0.00 |
| ILOGIC                      |    0 |     0 |          0 |       600 |  0.00 |
| OLOGIC                      |    0 |     0 |          0 |       600 |  0.00 |
+-----------------------------+------+-------+------------+-----------+-------+
```

```
5. Clocking
-----------

+------------+------+-------+------------+-----------+-------+
|  Site Type | Used | Fixed | Prohibited | Available | Util% |
+------------+------+-------+------------+-----------+-------+
| BUFGCTRL   |    1 |     0 |          0 |        32 |  3.13 |
| BUFIO      |    0 |     0 |          0 |        56 |  0.00 |
| MMCME2_ADV |    0 |     0 |          0 |        14 |  0.00 |
| PLLE2_ADV  |    0 |     0 |          0 |        14 |  0.00 |
| BUFMRCE    |    0 |     0 |          0 |        28 |  0.00 |
| BUFHCE     |    0 |     0 |          0 |       168 |  0.00 |
| BUFR       |    0 |     0 |          0 |        56 |  0.00 |
+------------+------+-------+------------+-----------+-------+
```



```
7. Primitives
-------------

+----------+------+---------------------+
| Ref Name | Used | Functional Category |
+----------+------+---------------------+
| OBUF     |   17 |                  IO |
| LDCE     |   17 |         Flop & Latch |
| LUT6     |    7 |                 LUT |
| LUT3     |    7 |                 LUT |
| LUT5     |    6 |                 LUT |
| LUT4     |    6 |                 LUT |
| IBUF     |    5 |                  IO |
| LUT2     |    4 |                 LUT |
| BUFG     |    1 |               Clock |
+----------+------+---------------------+
```


Utilization
Post-Synthesis | Post-Implementation
Graph | Table
LUT
1%
IO
4%
BUFG
3%
0
25
50
75
100
Estimated Utilization (%)