```
آنائیس گلبوداغیانس ۱۲۲۱۱۳•۴
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 \le a + x''2F'';
z \le a-x^25;
w \le (y(0)&y(7)&y(6)&y(5)&y(4)&y(3)&y(2)&y(1));
s \le (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 ضرب در 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 \le x''4B'';
wait for 2 ps;
a \le x"7A";
```

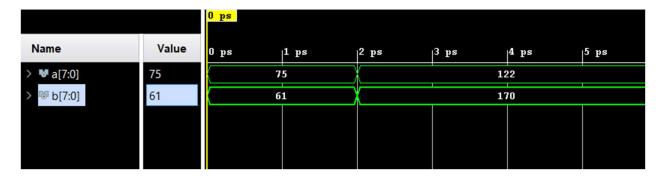
wait;

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 | | 15 | | LUT as Logic 0 | 0 | 303600 | <0.01 | | 0 | 0 | 0 | 130800 | 0.00 | | LUT as Memory 0 | 607200 | 0.00 | Register as Flip Flop | 0 | 0 | 0 | 607200 | 0.00 | | Register as Latch | 0 | 0 | 0 | 607200 | 0.00 | 0 | 151800 | 0.00 | F7 Muxes | 0 | 0 | 0 | 0 0 | 75900 | 0.00 | | F8 Muxes I +----+

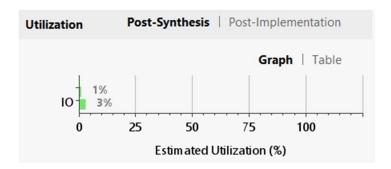
Memory

+-		+		+-		+		+		+-		+
I	Site Type	I	Used	I	Fixed	I	Prohibited	١	Available	I	Util%	I
+-		+		+-		+		+		+-		+
I	Block RAM Tile	I	0	I	0	I	0	I	1030	I	0.00	I
I	RAMB36/FIFO*	I	0	1	0	I	0	1	1030	1	0.00	I
I	RAMB18	I	0	I	0	I	0	1	2060	I	0.00	I
+-		-+-		+-		+		+		+-		+

3. DSP

+	-+	+	+	-+	-++
Site Type	Used	Fixed	Prohibited	Available	Util%
+	+	+	+	-+	-++
DSPs	1 0	0	0	2800	0.00

+	Site Type	+-	Used	+	Fixed	+	Prohibited	+	 Available	+	Util%	-+
+												
1	Bonded IOB	I	16	I	0	I	0	I	600	I	2.67	1
1	Bonded IPADs	I	0	١	0	I	0	١	62	1	0.00	I
1	Bonded OPADs		0	I	0	I	0	I	40	I	0.00	I
1	PHY_CONTROL	I	0	١	0	I	0	I	14	1	0.00	1
1	PHASER_REF	I	0	١	0	I	0	I	14	I	0.00	I
1	OUT_FIFO		0	١	0	I	0	I	56	I	0.00	I
1	IN_FIFO	1	0	I	0	I	0	1	56	1	0.00	1
1	IDELAYCTRL	I	0	I	0	I	0	I	14	1	0.00	1
1	IBUFDS		0	١	0	1	0	1	576	I	0.00	1
1	GTXE2_COMMON		0	-	0	I	0	1	5	1	0.00	I
1	GTXE2_CHANNEL	I	0	I	0	I	0	1	20	1	0.00	1
1	PHASER_OUT/PHASER_OUT_PHY	I	0	I	0	I	0	I	56	1	0.00	I
1	PHASER_IN/PHASER_IN_PHY		0	I	0	I	0	-	56	1	0.00	I
1	IDELAYE2/IDELAYE2_FINEDELAY	1	0	1	0	1	0	1	700	1	0.00	1
1	ODELAYE2/ODELAYE2_FINEDELAY	I	0	I	0	I	0	I	700	1	0.00	1
1	IBUFDS_GTE2		0	١	0	I	0	1	10	I	0.00	I
1	ILOGIC	I	0		0	I	0	1	600	1	0.00	
1	OLOGIC	I	0	I	0	I	0	1	600	1	0.00	I
+-		+-		+		+-		+		+-		-+



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;
          : in STD_LOGIC;
     rst
     start : in STD_LOGIC;
     sec_100 : out STD_LOGIC_VECTOR (7 downto 0);
           : out STD_LOGIC_VECTOR (7 downto 0);
     sec
           : out STD_LOGIC_VECTOR (7 downto 0)
     min
  );
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');
             : STD_LOGIC_VECTOR (7 downto 0) := (others => '0');
  signal m
begin
process(clk, rst)
begin
  if rst = '1' then
     s_100 <= (others => '0');
```

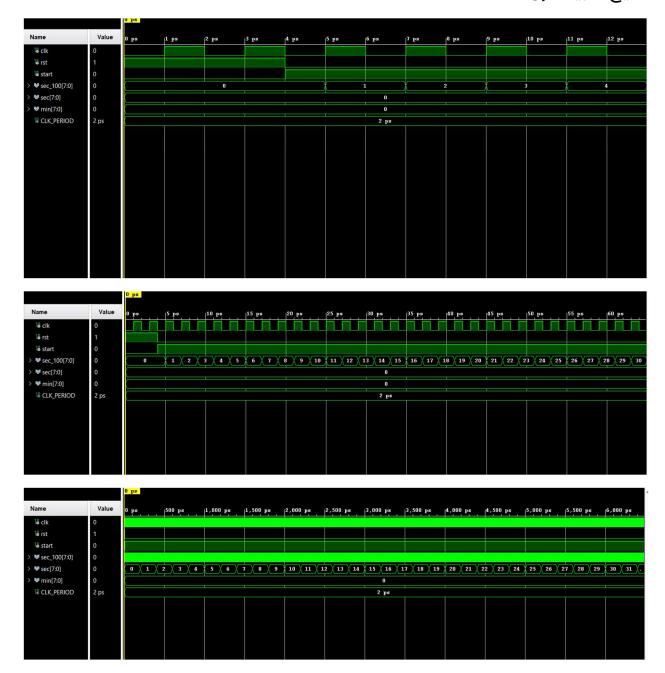
```
s <= (others => '0');
          <= (others => '0');
     m
  elsif rising_edge(clk) then
     if start = '1' then
        s_100 <= s_100 + 1;
        if s_{100} = 99 then
           s_100 <= (others => '0');
           s <= s + 1;
           if s = 59 then
              s <= (others => '0');
              m \le m + 1;
              if m = 59 then
                m <= (others => '0');
              end if;
           end if;
        end if;
     end if;
  end if;
end process;
```

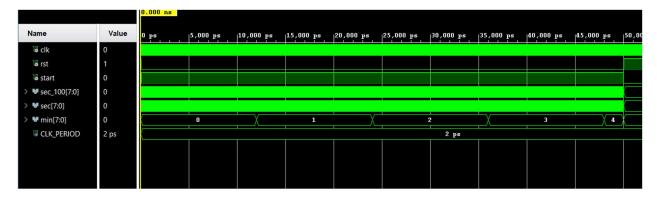
```
sec_100 <= s_100;
sec
      <= s;
min <= m;
end Behavioral;
                                                             کد تستبنچ:
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.NUMERIC_STD.ALL;
use ieee.std_logic_arith.all;
use ieee.std_logic_unsigned.all;
entity stopwatch_tb is
end stopwatch_tb;
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:

+----+ | Slice LUTs* 30 | 0 | 303600 | <0.01 | 30 | LUT as Logic 0 | 0 | 303600 | <0.01 | LUT as Memory 0 | 0 | 130800 | 0.00 | 0 | | 24 | | Slice Registers 0 | 0 | 607200 | <0.01 | Register as Flip Flop | 24 | 0 | 0 | 607200 | <0.01 | 607200 | 0.00 | Register as Latch 0 | 0 | 0 | | F7 Muxes 1 0 | 0 | 151800 | 0.00 | 0 | 75900 | 0.00 | | F8 Muxes 0 | 0 | 0 | 1

2. Memory

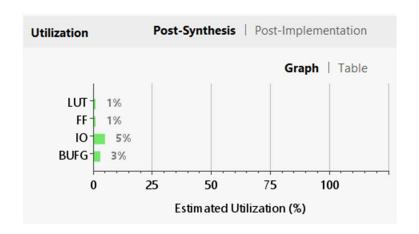
1	Site Type	I	Used	I	Fixed	1	Prohibited	I	Available	I	Util%	I
	Block RAM Tile	Ī	0	l	0	1	0	I	1030	1	0.00	I
1	RAMB36/FIFO* RAMB18	1	0		0	:	0	Ċ	2060	•	0.00	•
+-	KAMDIO	- -+		•		•	0	•		•		•

+		+		+		+-		+		+		-+
1	Site Type	•		•			Prohibited	•		•		Ī
I	Bonded IOB	ı	27		0		0					Ī
1	Bonded IPADs	1	0	1	0	1	0	I	62	I	0.00	1
I	Bonded OPADs	I	0	I	0	I	0	I	40	I	0.00	I
1	PHY_CONTROL	١	0	١	0	I	0	I	14	ı	0.00	I
1	PHASER_REF	1	0	١	0	I	0	I	14	I	0.00	I
1	OUT_FIFO	1	0	I	0	I	0	١	56	١	0.00	I
1	IN_FIFO	١	0	I	0	I	0	I	56	I	0.00	I
1	IDELAYCTRL	I	0	I	0	1	0	I	14	I	0.00	I
1	IBUFDS	I	0	1	0	1	0	I	576	I	0.00	1
1	GTXE2_COMMON	١	0	I	0	I	0	I	5	I	0.00	I
1	GTXE2_CHANNEL	١	0	I	0	I	0	I	20	I	0.00	I
1	PHASER_OUT/PHASER_OUT_PHY	I	0	I	0	1	0	I	56	I	0.00	I
1	PHASER_IN/PHASER_IN_PHY	١	0	I	0	1	0	١	56	I	0.00	1
1	IDELAYE2/IDELAYE2_FINEDELAY	I	0	I	0	I	0	I	700	I	0.00	I
1	ODELAYE2/ODELAYE2_FINEDELAY	١	0	I	0	I	0	I	700	I	0.00	I
I	IBUFDS_GTE2	1	0	1	0	1	0	I	10	I	0.00	I
1	ILOGIC	١	0	I	0	I	0	I	600	I	0.00	I
I	OLOGIC	١	0	I	0	I	0	I	600	I	0.00	I
+		+		+		+-		+		+		-+

5. Clocking

1		ĺ	Used	l	Fixed	ĺ	Prohibited	I	Available	I	Util%	I
ī	BUFGCTRL	ı	1		0	ı	0	ı	32		3.13	
1	BUFIO	I	0	ı	0	I	0	I	56	I	0.00	1
1	MMCME2_ADV	I	0	I	0	1	0	١	14	I	0.00	1
1	PLLE2_ADV	I	0	I	0	1	0	1	14	I	0.00	1
1	BUFMRCE	I	0	I	0	I	0	I	28	I	0.00	١
1	BUFHCE	I	0	I	0	1	0	١	168	I	0.00	1
1	BUFR	I	0	I	0	I	0	١	56	I	0.00	1
+-		4.		+-		4.		-+		-+		-+

+	+	++
Ref Name	Used	Functional Category
+	+	++
OBUF	24	10
FDCE	24	Flop & Latch
LUT6	19	LUT
LUT4	5	LUT
LUT2	5	LUT
LUT5	4	LUT
LUT3	3	LUT
IBUF	3	10
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;
```

```
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 \le x"7";
wait for 400 ps;
c <= x''A'';
wait;
end process;
```

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

c =>c,

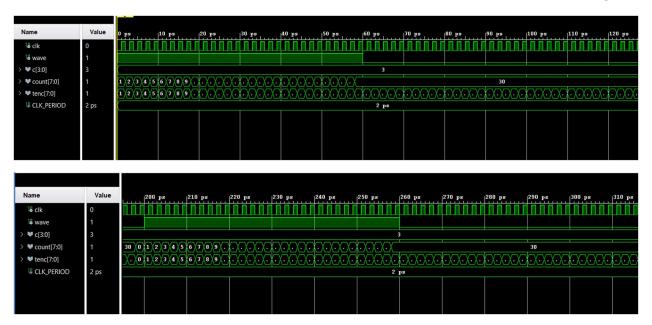
wave => wave,

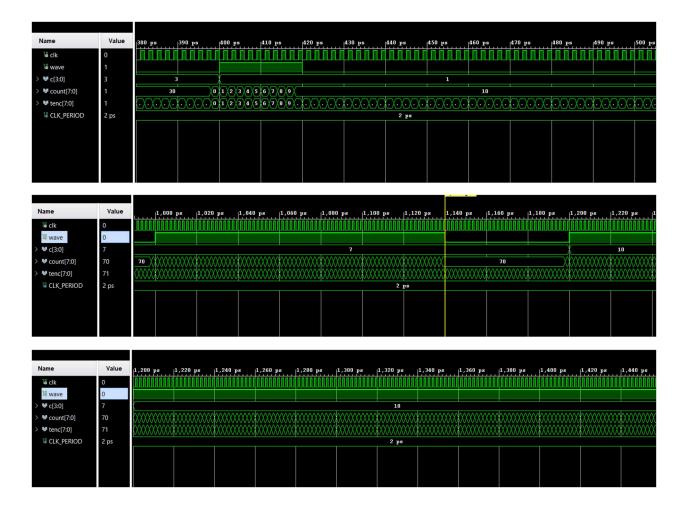
tenc => tenc,

count => count);

end Behavioral;

نتایج شبیهسازی:





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
```

+		+-		+-		+		+		+-		-+
I	Site Type	•		•		•	Prohibited	•		•		•
I	Slice LUTs*	Ī	21		0	•	0	•	303600			•
1	LUT as Logic	I	21	I	0	I	0	I	303600	I	<0.01	I
1	LUT as Memory	1	0	I	0	I	0	I	130800	I	0.00	1
1	Slice Registers	1	17	I	0	I	0	I	607200	1	<0.01	I
1	Register as Flip Flop	I	0	I	0	I	0	I	607200	1	0.00	I
1	Register as Latch	I	17	I	0	I	0	I	607200	I	<0.01	I
1	F7 Muxes	1	0		0	I	0	I	151800	I	0.00	I
1	F8 Muxes	1	0	1	0	I	0	I	75900	1	0.00	I
+		+-		+-		+		+		4.		-+

1.1 Summary of Registers by Type

| Total | Clock Enable | Synchronous | Asynchronous | +----+ 0 - 1

_ 1 Set | _ 1 1 0 Reset | 0 Set | 0 Reset - | 1 0 Yes 1 0 Yes - 1 Set | I 1 17 1 Yes Reset | 0 Yes Set | - 1 - 1 1 0 I Yes Reset

2. Memory

Ī	Site Type	1	Used	1	Fixed	I	Prohibited	I	Available	I	Util%	I
	Block RAM Tile						0				0.00	
1	RAMB36/FIFO*	١	0	I	0	I	0	I	1030	I	0.00	I
1	RAMB18	١	0	1	0	I	0	I	2060	I	0.00	1
+		-+-				4		4		4.		_

+		+-		+		+		+		-+		+-
1	Site Type	I	Used	I	Fixed	I	Prohibited	١	Available	1	Util%	I
+		+-		+		+		+		-+		-+
1	Bonded IOB	I	22	١	0	١	0	١	600	I	3.67	I
1	Bonded IPADs	I	0	I	0	1	0	١	62	1	0.00	I
1	Bonded OPADs	1	0	I	0	1	0	١	40	1	0.00	I
I	PHY_CONTROL	١	0	I	0	١	0	١	14	١	0.00	I
1	PHASER_REF	I	0	I	0	1	0	١	14	1	0.00	I
I	OUT_FIFO	1	0	I	0	1	0	١	56	1	0.00	1
1	IN_FIFO	1	0	I	0	I	0	١	56	1	0.00	I
1	IDELAYCTRL	١	0	١	0	١	0	١	14	1	0.00	I
1	IBUFDS	I	0	1	0	١	0	1	576	1	0.00	I
1	GTXE2_COMMON	I	0	I	0	I	0	I	5	I	0.00	I
1	GTXE2_CHANNEL	I	0	I	0	١	0	١	20	I	0.00	I
1	PHASER_OUT/PHASER_OUT_PHY	I	0	I	0	I	0	I	56	1	0.00	I
1	PHASER_IN/PHASER_IN_PHY	1	0	I	0	1	0	١	56	1	0.00	I
1	IDELAYE2/IDELAYE2_FINEDELAY	1	0	I	0	I	0	I	700	1	0.00	I
1	ODELAYE2/ODELAYE2_FINEDELAY	I	0	١	0	١	0	١	700	1	0.00	I
1	IBUFDS_GTE2	I	0	I	0	I	0	1	10	1	0.00	I
1	ILOGIC	I	0	I	0	I	0	I	600	1	0.00	I
I	OLOGIC	I	0	I	0	I	0	١	600	١	0.00	I
+		+-		+		+		+		-+		-+

5. Clocking

+-		•					Prohibited					
+-		+		+-		+-		+		+-		+
1	BUFGCTRL	I	1	١	0	I	0	I	32	I	3.13	Ī
1	BUFIO	I	0	1	0	I	0	I	56	I	0.00	I
1	MMCME2_ADV	١	0	I	0	١	0	I	14	١	0.00	I
1	PLLE2_ADV		0	I	0	I	0	I	14	I	0.00	1
1	BUFMRCE	1	0	١	0	I	0	1	28	I	0.00	I
1	BUFHCE	١	0	١	0	I	0	I	168	I	0.00	I
1	BUFR		0	1	0	I	0	I	56	I	0.00	I
+-		+		+-		+-		-+		+-		+

7. Primitives

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

