

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«Владимирский государственный университет
имени Александра Григорьевича и Николая Григорьевича Столетовых»
(ВлГУ)

Кафедра «Вычислительная техника»

Лабораторная работа № 2

по дисциплине

«САПР ПЛИС и ИМС»

Выполнил:
ст. гр. ВТм - 112
А.Х. Муна

Принял:
С.С. Гладько

Владимир 2012

Разработка ядра цифрового блока на базе языка VHDL

Цель работы: изучение правил создания цифровых блоков на VHDL, разработка ядра цифрового блока.

Вариант 10.

1. **Задание:** Счетчик со сбросом, загрузкой, сигналом разрешения – устройство подсчитывает количество изменений входных данных с последнего сигнала «сброс». Счет выполняется только при наличии сигнала «разрешение счета».

Вход: входные 32-х разрядные данные;
сброс;
разрешение счета.

Выход: выходные данные.

1. Спецификация устройства:

Устройство подсчитывает количество изменений входных данных с последнего сигнала «сброс». Вход: двоичное 32-х разрядное число

Выход: двоичное число

Интерфейс устройства описывается следующим образом:

entity counter is

port(

CLK: in std_logic;

RST: in std_logic;

EN : in std_logic;

DI :in std_logic_vector(31 downto 0);

CNT: out std_logic_vector(31 downto 0)

);

DI Входными сигналами являются сигналы CLK-синхросигнал, RST-сброса шина
EN шина разрешение. Выходной сигнал CNT- шина последовательная числа.

В ходе работы созданы два ядра - netlist файлы для реализаций устройства в базе FPGA ПЛИС Virtex4 XC4VFX12 и Kintex7 XC7K70T.

Файлы имеют обозначения counter.ngc.

VHDL

```
library IEEE;
```

```
use IEEE.STD_LOGIC_1164.ALL;
```

```
use IEEE.STD_LOGIC_ARITH.ALL;
```

```
use IEEE.STD_LOGIC_UNSIGNED.ALL;
```

```

entity counter is
port(
    CLK: in std_logic;
    RST: in std_logic;
    EN : in std_logic;
    DI :in std_logic_vector(31 downto 0);
    CNT: out std_logic_vector(31 downto 0)

);
end counter;

architecture Behavioral of counter is
    signal temp: std_logic_vector(31 downto 0);
begin
    CounterProcess: process(DI, RST, CLK, EN)
    begin
        if( RST ='1') then
            temp <= (others =>'0');
        else
            if(rising_edge(CLK) and EN = '1') then
                temp <= temp+1;
            end if;
        end if;
    end process;
    CNT <= temp;
end Behavioral;

```

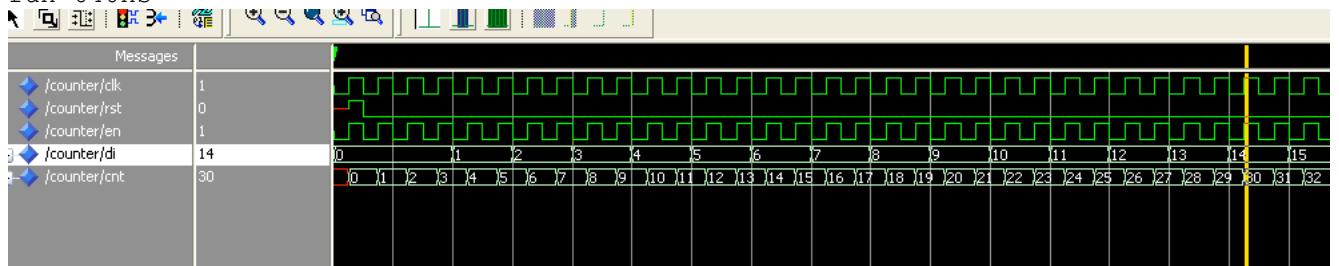
2. Тестирование

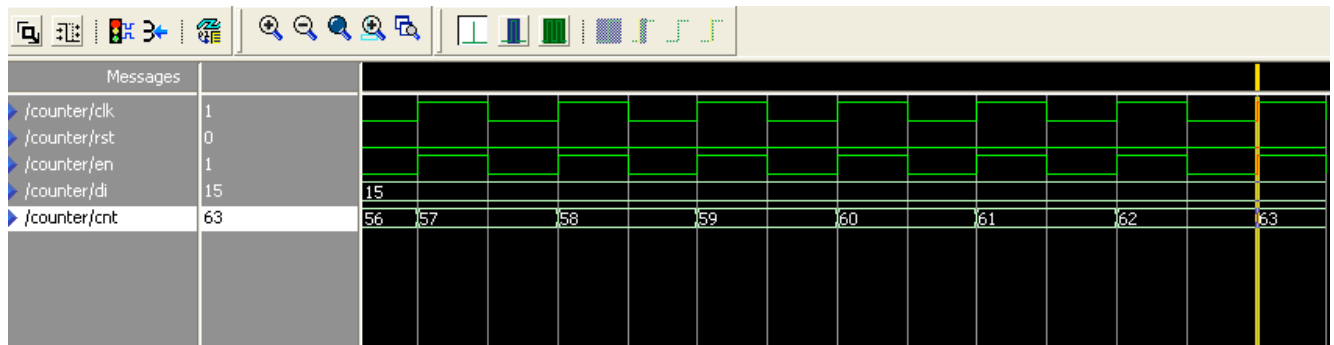
-modelsim:

```

force sim:/counter/clk 0 0,1 5ns -r 10ns
force sim:/counter/en 1 10ns, 0 320ns
force sim:/counter/rst 1 5ns, 0 10ns
force sim:/counter/DI 00000000000000000000000000000000
0ns,000000000000000000000000000000000001 40ns,000000000000000000000000000000000010
60ns,000000000000000000000000000000000011 80ns,0000000000000000000000000000000000100
100ns,0000000000000000000000000000000000101 120ns,0000000000000000000000000000000000110
140ns, 0000000000000000000000000000000000111 160ns, 00000000000000000000000000000000001000
180ns, 00000000000000000000000000000000001001 200ns, 00000000000000000000000000000000001010
220ns, 00000000000000000000000000000000001011 240ns, 00000000000000000000000000000000001100
260ns, 00000000000000000000000000000000001101 280ns, 00000000000000000000000000000000001110
300ns, 00000000000000000000000000000000001111 320ns
run 640ns

```





-Xilinx:

```

LIBRARY ieee;
USE ieee.std_logic_1164.ALL;

-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
--USE ieee.numeric_std.ALL;

ENTITY counter_test_bench IS
END counter_test_bench;

ARCHITECTURE behavior OF counter_test_bench IS

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

    COMPONENT counter
    PORT (
        CLK : IN  std_logic;
        RST : IN  std_logic;
        EN  : IN  std_logic;
        DI  : IN  std_logic_vector(31 downto 0);
        CNT : OUT std_logic_vector(31 downto 0)
    );
    END COMPONENT;

    --Inputs
    signal CLK : std_logic := '0';
    signal RST : std_logic := '0';
    signal EN  : std_logic := '0';
    signal DI  : std_logic_vector(31 downto 0) := (others => '0');

    --Outputs
    signal CNT : std_logic_vector(31 downto 0);

    -- Clock period definitions
    constant CLK_period : time := 10 ns;

BEGIN

    -- Instantiate the Unit Under Test (UUT)
    uut: counter PORT MAP (
        CLK => CLK,
        RST => RST,
        EN  => EN,
        DI  => DI,
        CNT => CNT
    );

    -- Clock process definitions
    CLK_process :process
begin

```

```

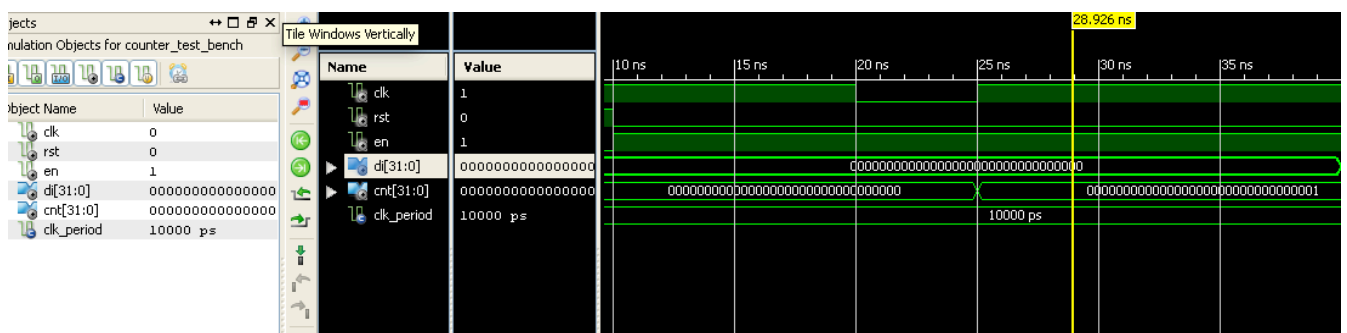
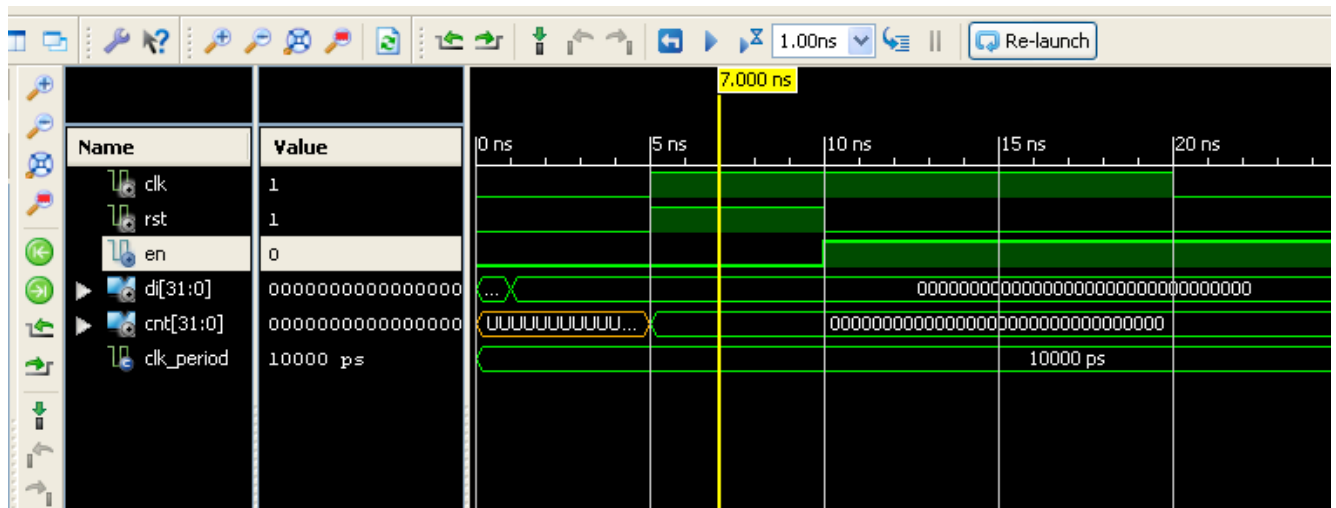
CLK <= '0';
wait for CLK_period/2;
CLK <= '1';
wait for CLK_period/2;
EN <= '1'; wait for CLK_period;
end process;

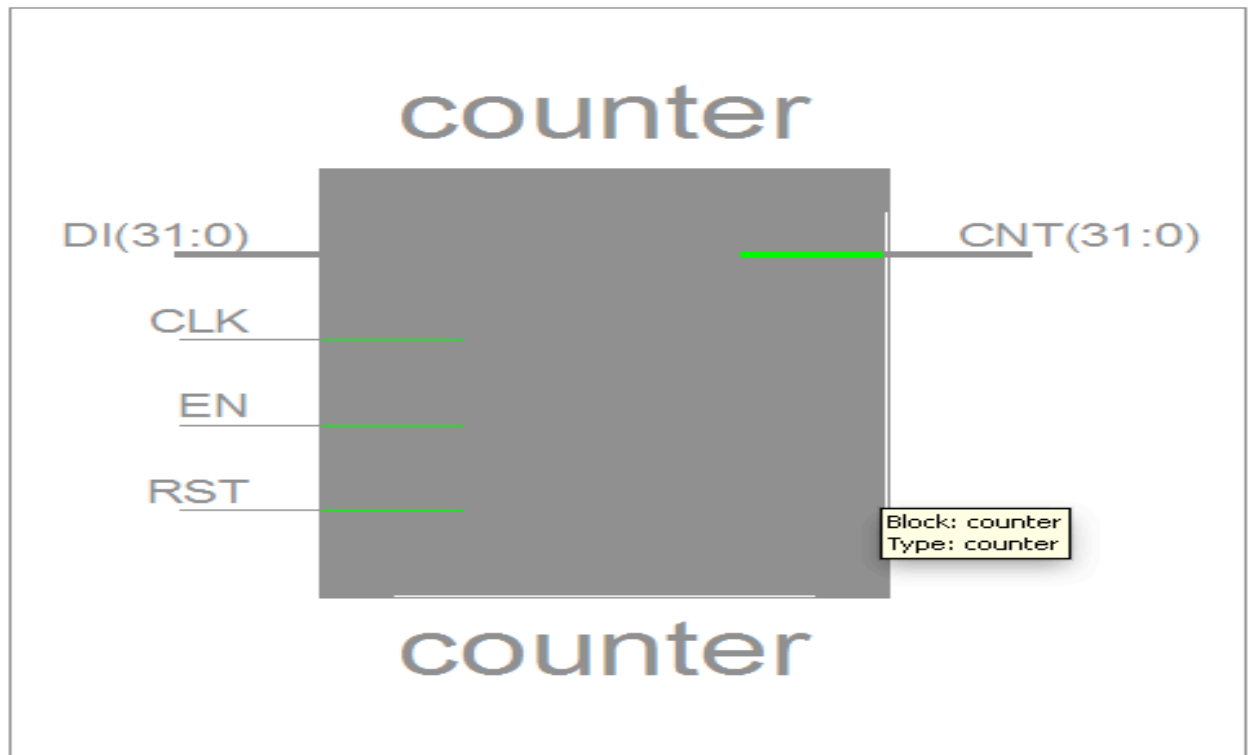
RST <= '0', '1' after 5ns, '0' after 10ns;

DI <= (others => '1'),
      (0 => '0', others => '0') after 1ns,
      (0 => '1', others => '0') after 40ns,
      (1 => '1', 0 => '0', others => '0') after 60ns,
      (1 => '1', 0 => '1', others => '0') after 80ns,
      (2 => '1', 1 => '0', 0 => '0', others => '0') after 100ns,
      (2 => '1', 1 => '0', 0 => '1', others => '0') after 120ns,
      (2 => '1', 1 => '1', 0 => '0', others => '0') after 140ns,
      (2 => '1', 1 => '1', 0 => '1', others => '0') after 160ns,
      (3 => '1', 2 => '0', 1 => '0', 0 => '0', others => '0') after 180ns,
      (3 => '1', 2 => '0', 1 => '0', 0 => '1', others => '0') after
200ns,
      (3 => '1', 2 => '0', 1 => '1', 0 => '0', others => '0') after 220ns,
      (3 => '1', 2 => '0', 1 => '1', 0 => '1', others => '0') after 240ns,
      (3 => '1', 2 => '1', 1 => '0', 0 => '0', others => '0') after
260ns,
      (3 => '1', 2 => '1', 1 => '0', 0 => '1', others => '0') after
280ns,
      (3 => '1', 2 => '1', 1 => '1', 0 => '0', others => '0') after
300ns,
      (3 => '1', 2 => '1', 1 => '1', 0 => '1', others => '0') after
320ns;

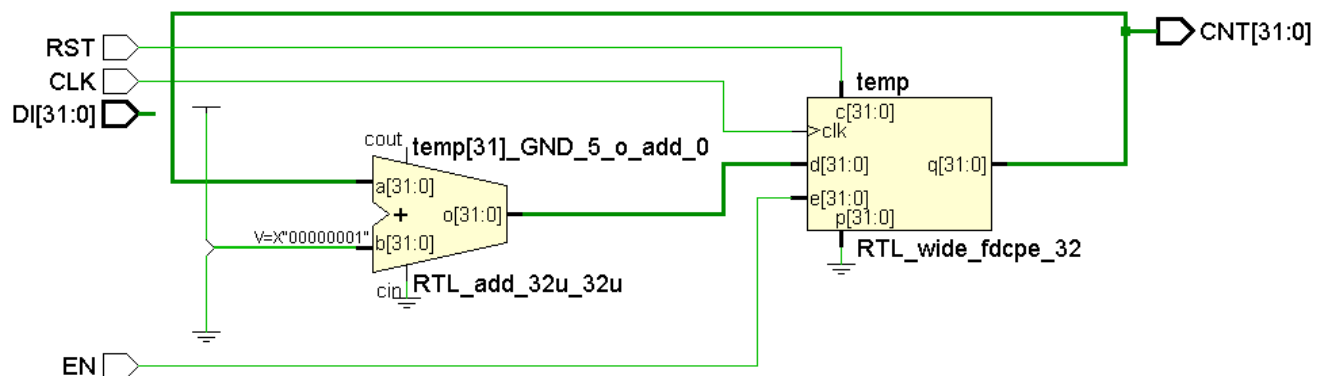
END;

```

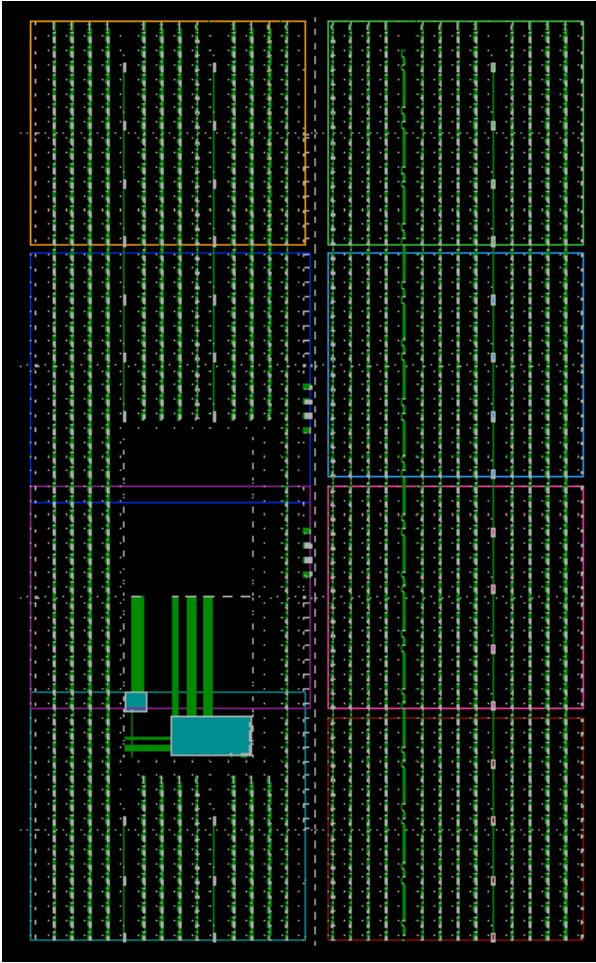


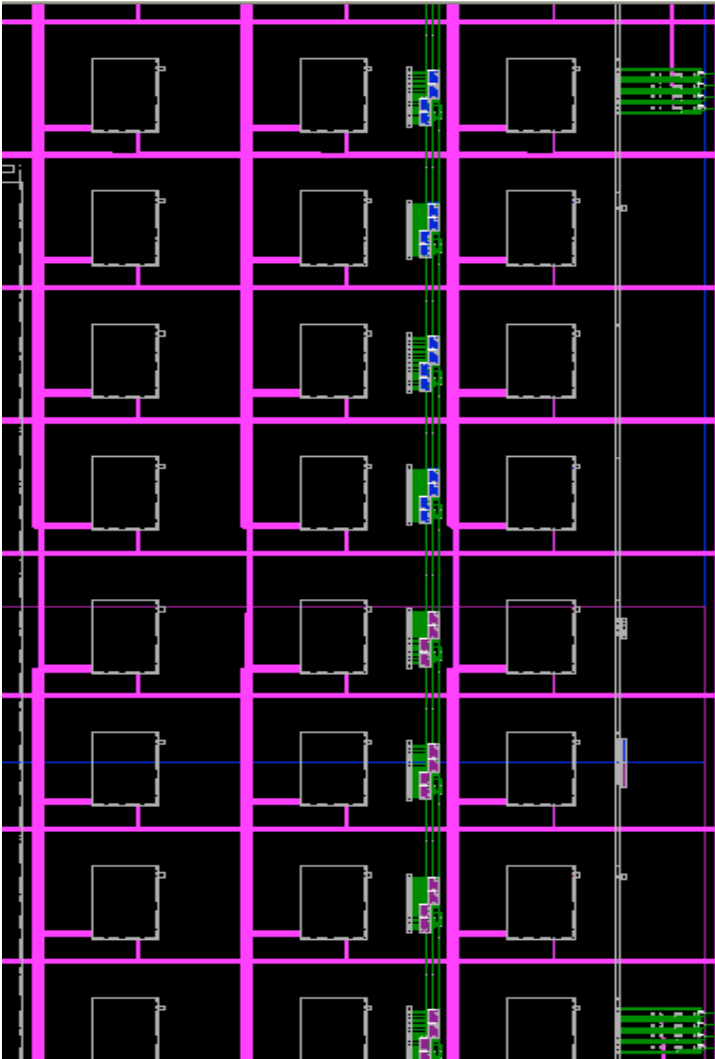


Белый ящик:

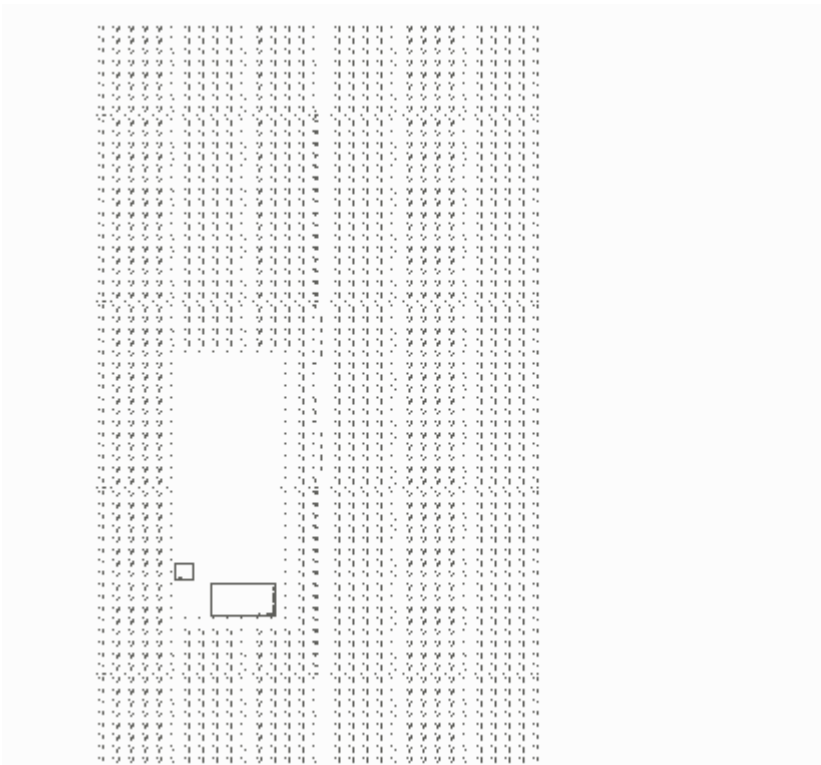


Размещение(Virtex4 XC4VFX12):





Трассировка(Virtex4 XC4VFX12):

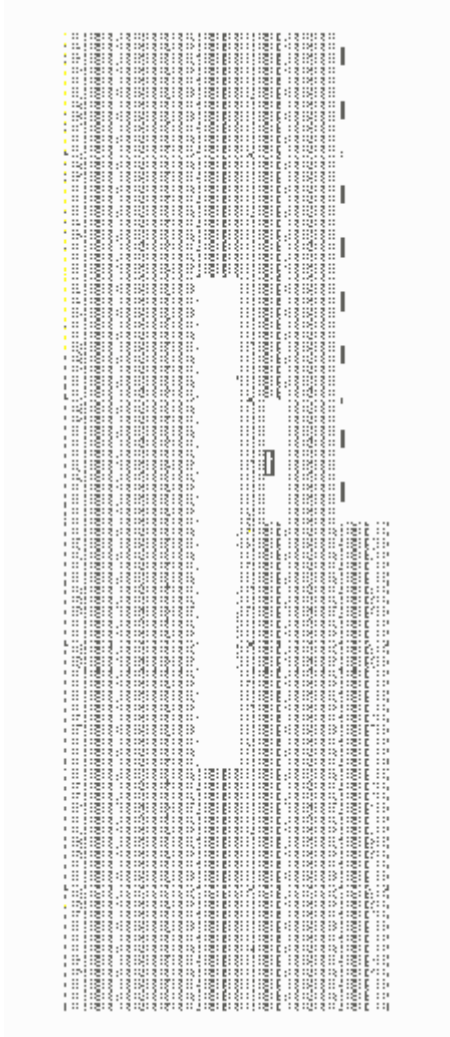


RTL-схема(Kintex7 XC7K70T):

Размещение(Kintex7 XC7K70T):



Трассировка(Kintex7 XC7K70T):



Вывод: изучены правила создания цифровых блоков на VHDL и разработки ядра цифрового блока.