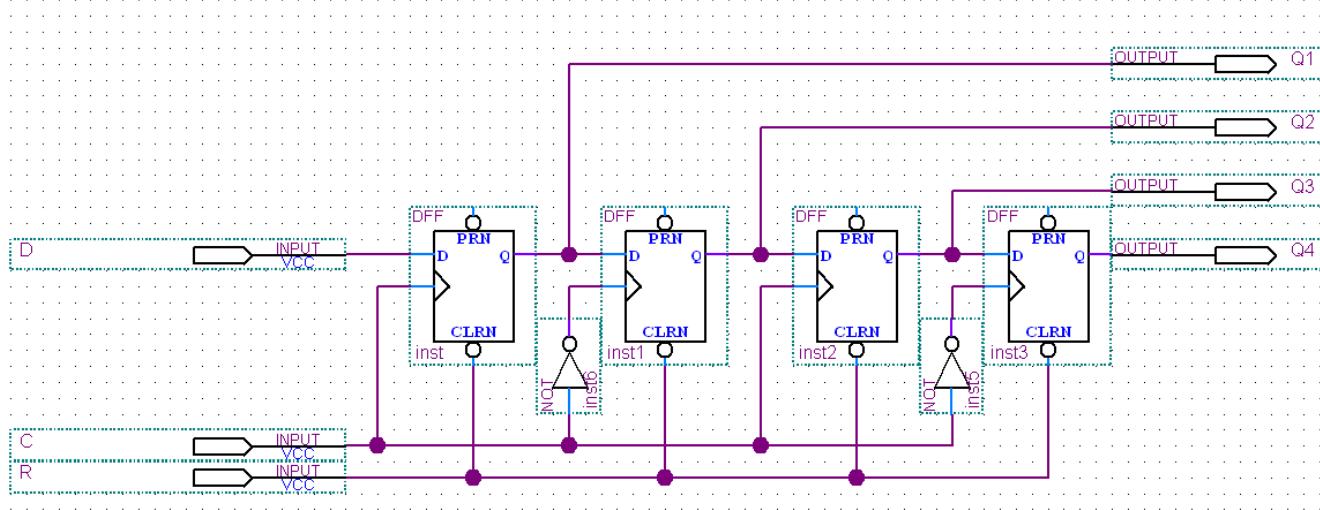


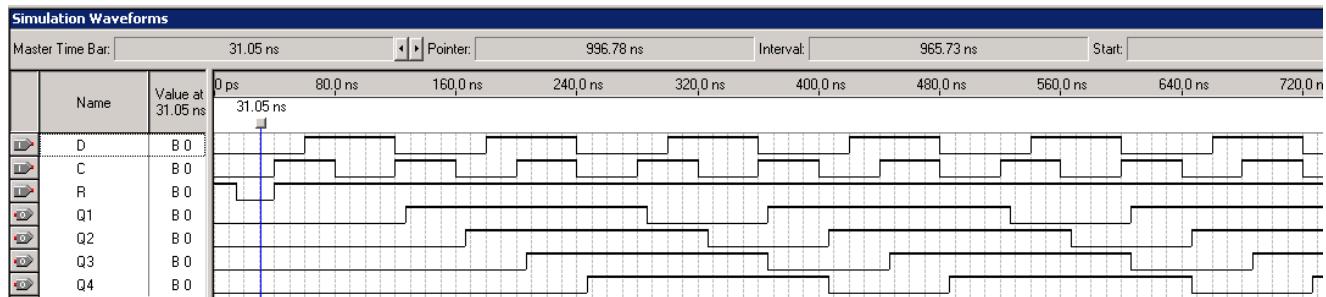
## 2.1

Создать в графическом редакторе проект схемы двухразрядного сдвигающего регистра на D-триггерах (элемент DFF) с одним тактовым входом, откомпилировать и промоделировать его работу:



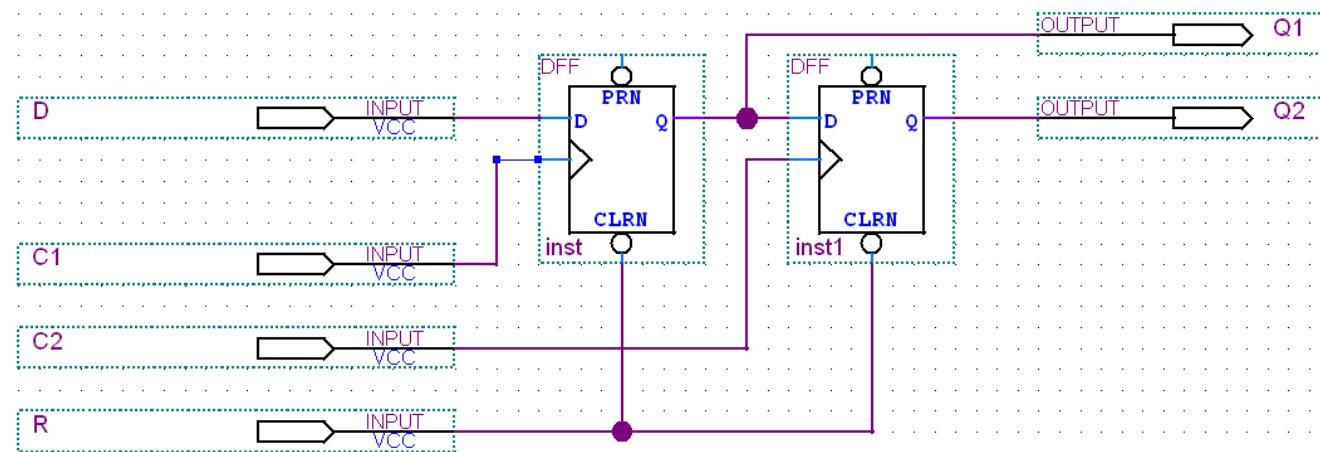
## 2.2

Зарисовать временные диаграммы



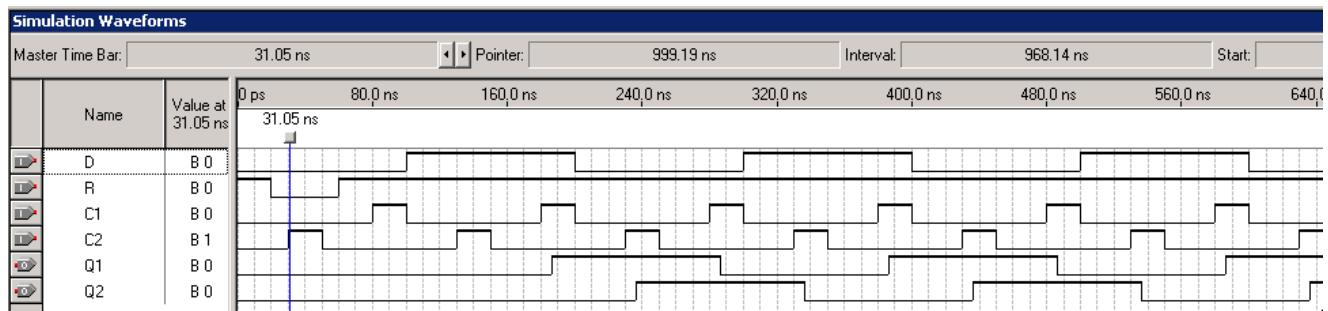
## 3.1

Создать в графическом редакторе проект схемы D-триггера с двумя тактовыми входами на элементах DFF из библиотеки примитивов, откомпилировать и промоделировать его работу.



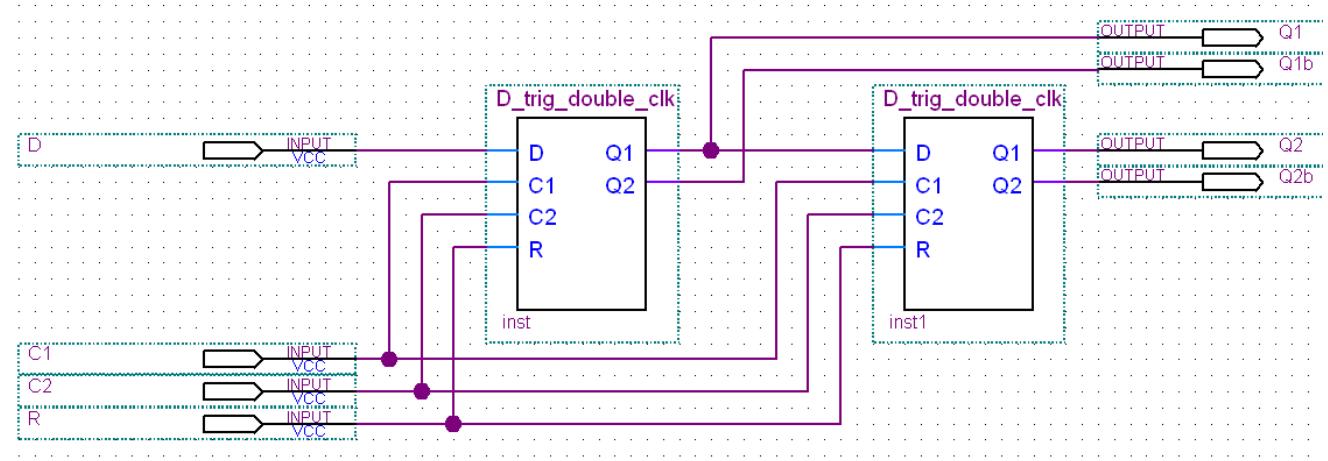
## 3.2

Зарисовать временные диаграммы:



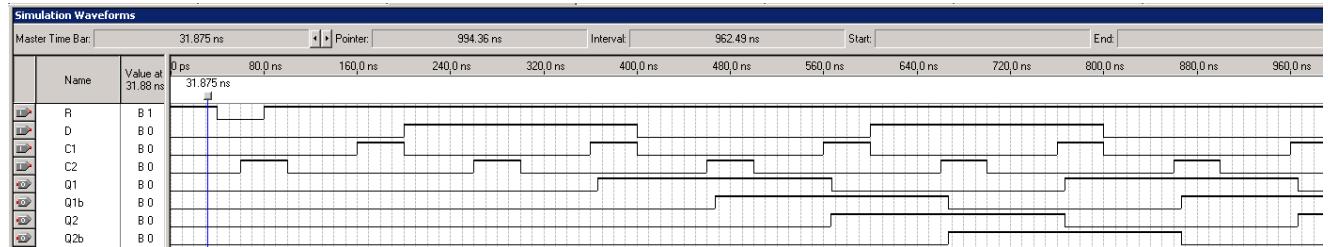
## 4.1

Создать в графическом редакторе проекты схем двухразрядного сдвигающего регистра на D-триггерах с двумя тактовыми входами, откомпилировать и промоделировать работу этих схем.



## 4.2

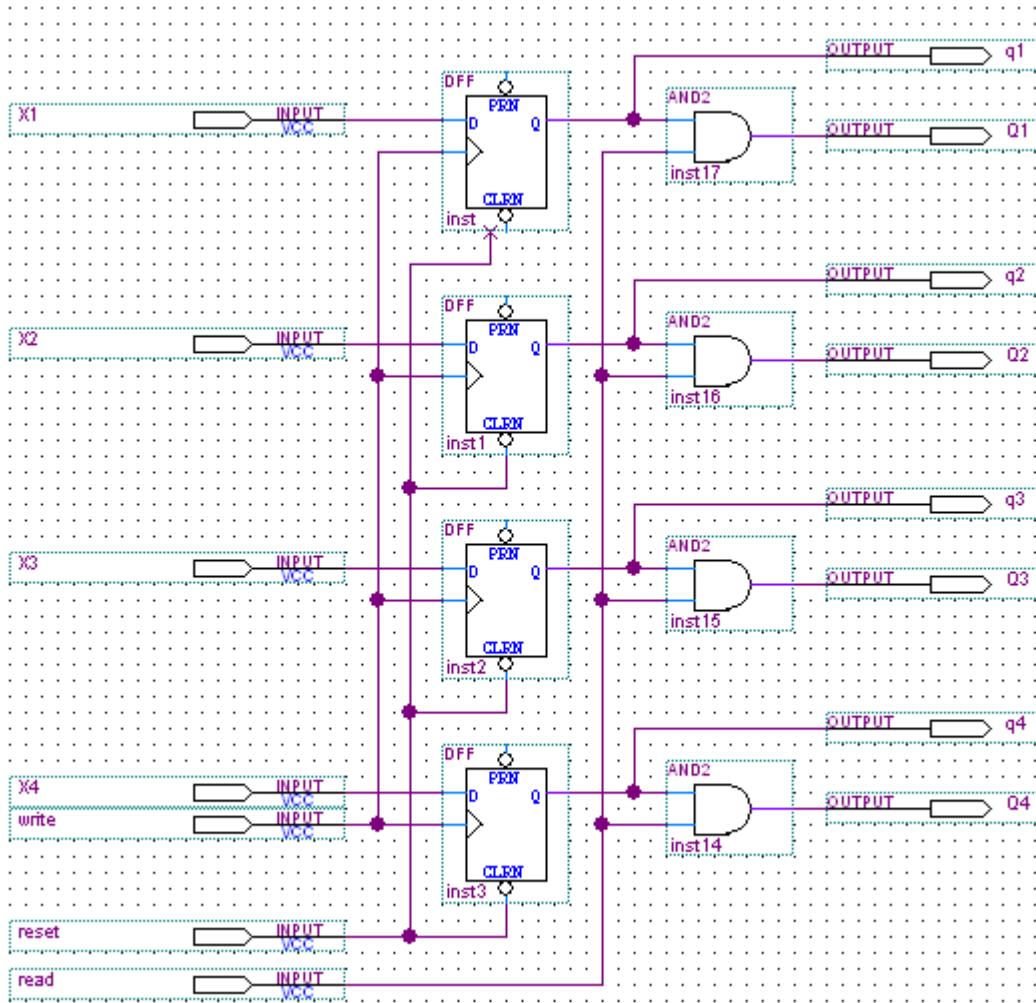
Зарисовать временные диаграммы:



## 5.1

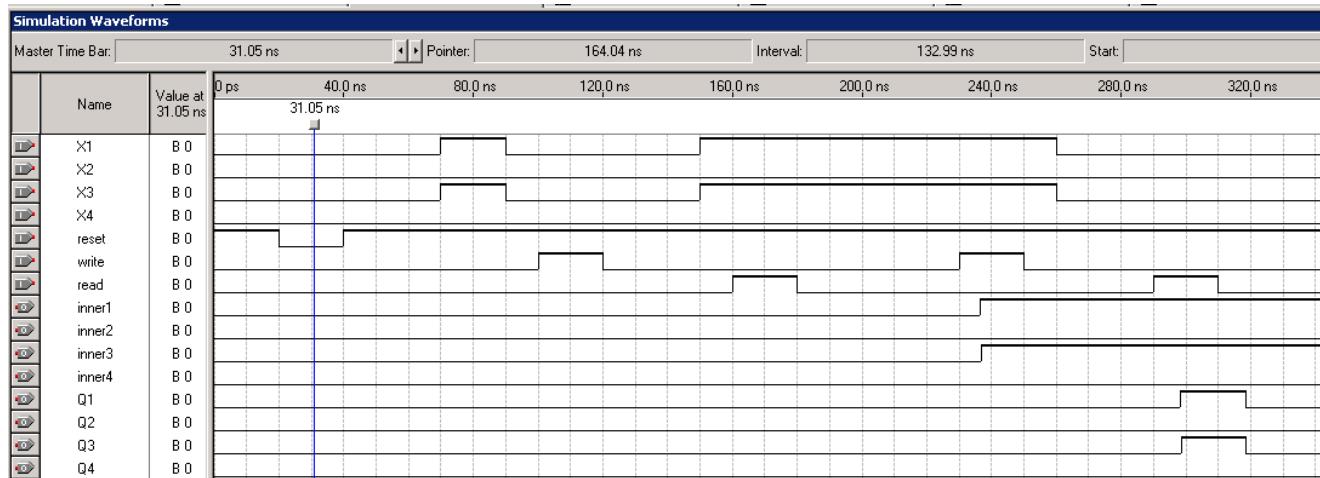
Создать в графическом редакторе проект схемы четырехразрядного регистра памяти на D-триггерах (элемент DFF), откомпилировать и

промоделировать его работу



## 5.2

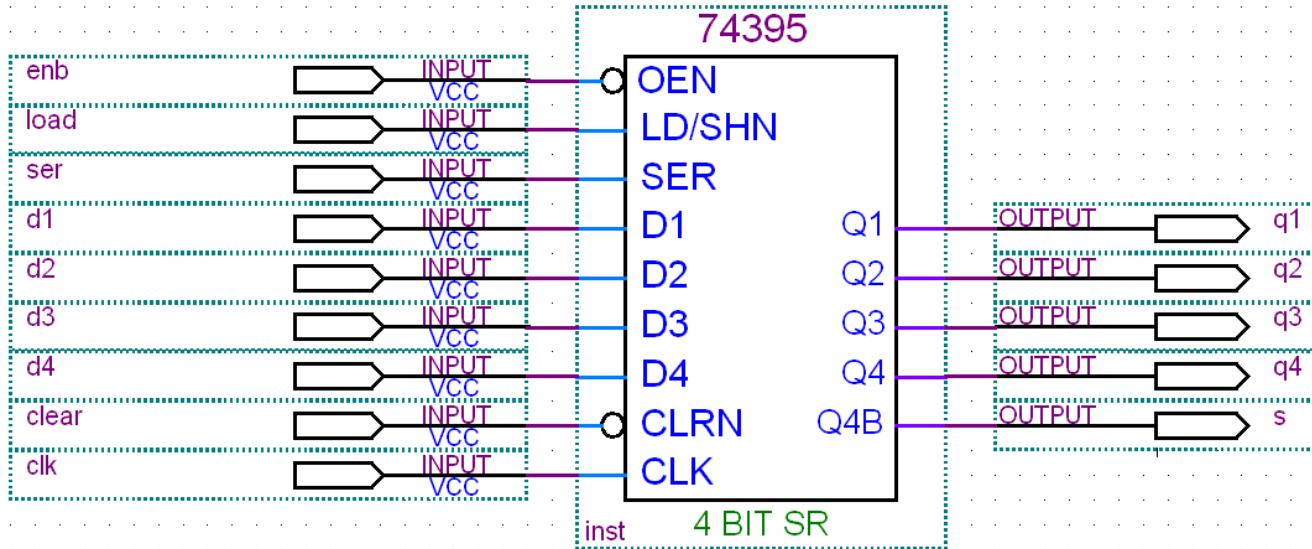
Зарисовать временные диаграммы:



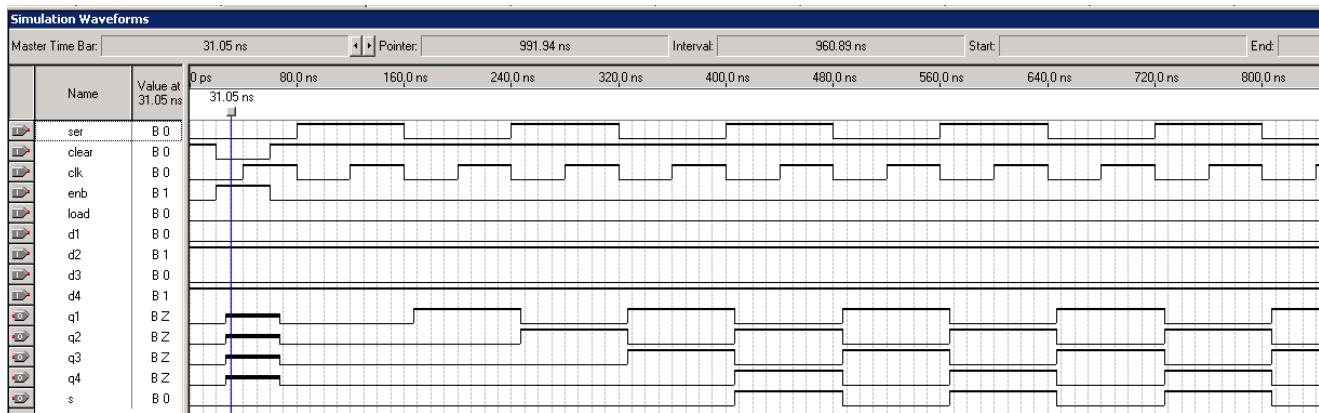
## 6

Исследовать работу элемента 74395 библиотеки макрофункций mf.

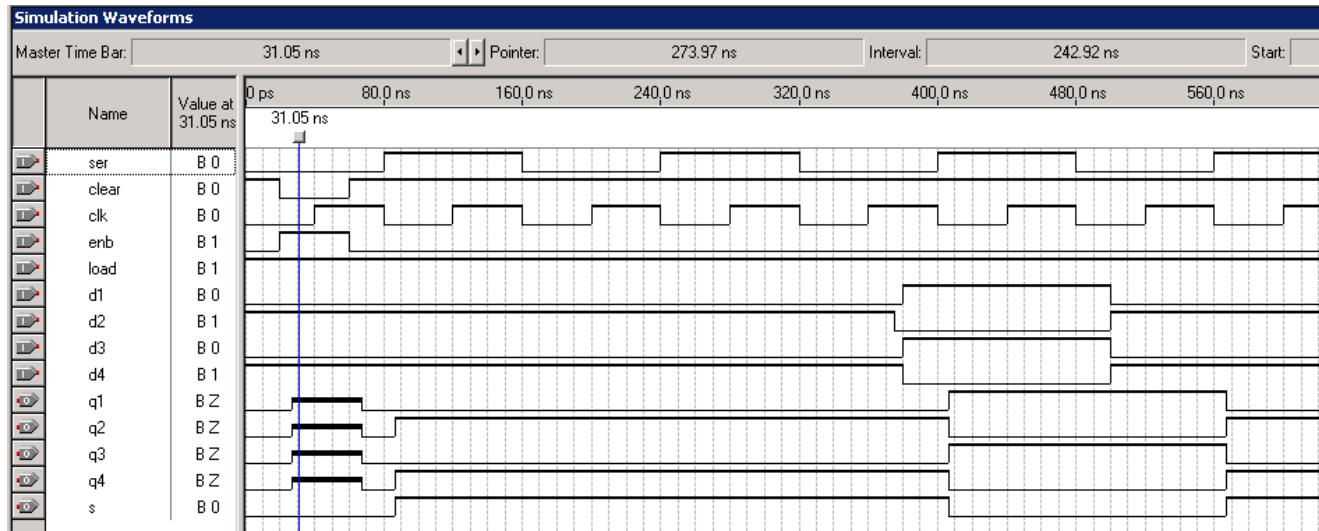
Откомпилировать проект и получить временные диаграммы работы регистра в разных режимах



В последовательном режиме:



В параллельном режиме:



## 7.1

Реализовать проект двухразрядного сдвигающего регистра с одним тактовым входом на языке программирования VHDL, используя поведенческую модель функционирования устройства.

```

LIBRARY ieee;
USE ieee.std_logic_1164.ALL;

ENTITY SH_reg_2bit IS

```

```

PORT( D : IN std_logic;
      C : IN std_logic;
      Q0 : OUT std_logic;
      Q1 : OUT std_logic);
END SH_reg_2bit;

ARCHITECTURE behav OF SH_reg_2bit IS
  SIGNAL reg : std_logic_vector(1 downto 0);
BEGIN PROCESS (C)
  BEGIN
    IF rising_edge(C) THEN
      reg(1) <= reg(0);
      reg(0) <= D;
    END IF;
  END PROCESS;

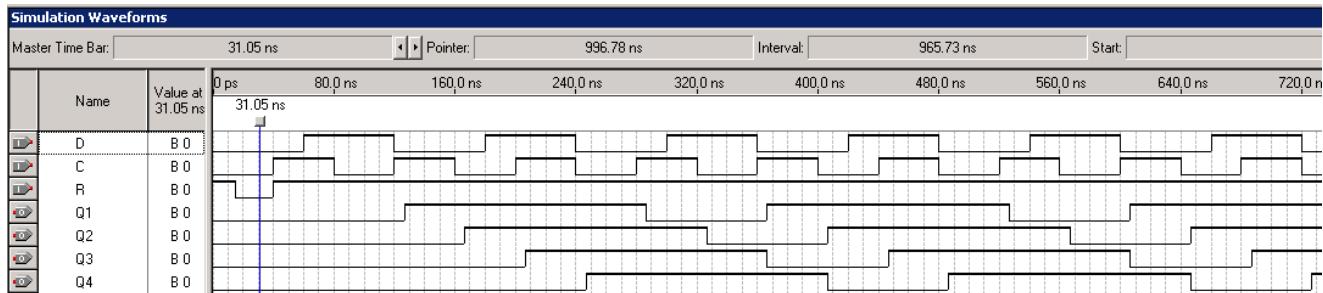
  Q0 <= reg(0);
  Q1 <= reg(1);
END behav;

```

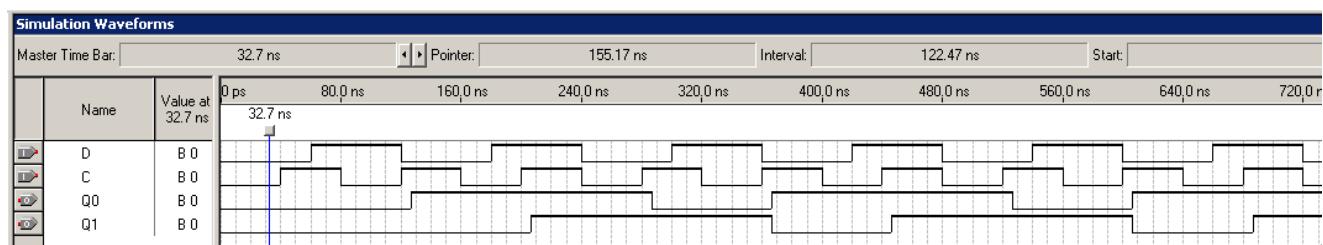
## 7.2

Откомпилировать проект и получить временные диаграммы. Сравнить их с диаграммами, полученными в п.2:

Сравним диаграмму созданного элемента в граф. ред.



и элемента написанного на vhdl



Диаграммы совпали.

## 8.1

Реализовать проект двухразрядного сдвигающего регистра с двумя тактовыми входами, основанного на двухтактных D-триггерах на языке программирования VHDL, используя поведенческую модель функционирования устройства

```

LIBRARY ieee;
USE ieee.std_logic_1164.ALL;

ENTITY SH_2b_reg IS
    PORT(    D : IN  std_logic;
              C1 : IN  std_logic;
              C2 : IN  std_logic;
              R : IN  std_logic;
              Q : OUT std_logic_vector (1 downto 0) );
END SH_2b_reg;

ARCHITECTURE beh OF SH_2b_reg IS

    SIGNAL stg1 : std_logic_vector(1 downto 0);
    SIGNAL stg2 : std_logic_vector(1 downto 0);

    BEGIN PROCESS (C1, R)
        BEGIN
            IF (R = '1') THEN
                stg1 <= "00";
            ELSIF rising_edge (C1) THEN
                stg1(1) <= D;
                stg1(0) <= stg1(1);
            END IF;
        END PROCESS;

        PROCESS (C2, R)
        BEGIN
            IF (R = '1') THEN
                stg2 <= "00";
            ELSIF rising_edge (C2) THEN
                stg2(1) <= stg1(0);
                stg2(0) <= stg2(1);
            END IF;
        END PROCESS;

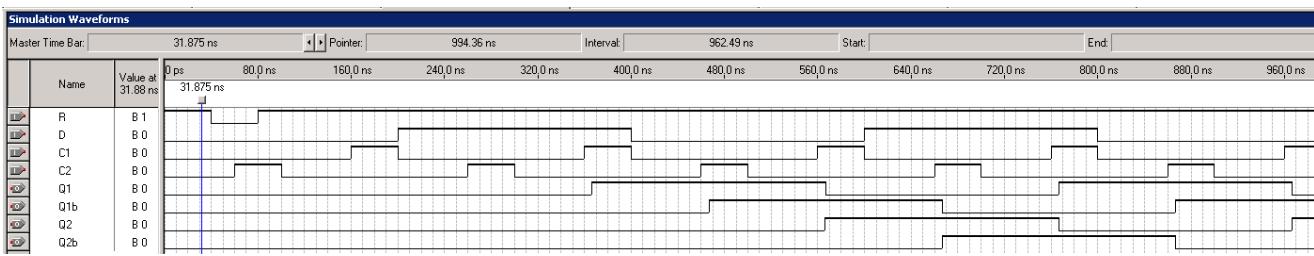
        Q <= stg2;
    END beh;

```

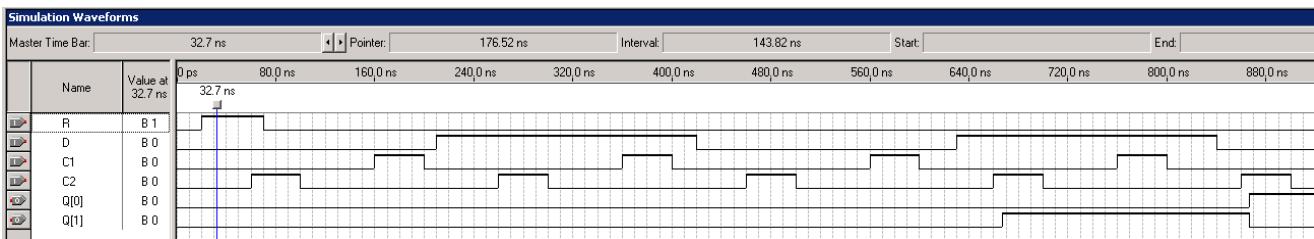
## 8.2

Откомпилировать проект и получить временные диаграммы. Сравнить их с диаграммами, полученными в п.4:

Сравним диаграмму созданного элемента в граф. ред.



и элемента написанного на vhdl



Диаграммы совпали.

## 9.1

Реализовать проект регистра памяти на языке программирования VHDL, используя поведенческую модель функционирования устройства.

```

LIBRARY ieee;
USE ieee.std_logic_1164.ALL;

ENTITY mem_reg IS
  PORT(
    X1, X2, X3, X4 : IN std_logic;
    write           : IN std_logic;
    reset           : IN std_logic;
    read            : IN std_logic;
    Q1, Q2, Q3, Q4 : OUT std_logic );
END mem_reg;

ARCHITECTURE behav OF mem_reg IS
  SIGNAL reg : std_logic_vector (3 DOWNTO 0);
BEGIN
  PROCESS(write, reset)
  BEGIN
    IF (reset = '0') THEN
      reg <= "0000";
    ELSIF (write = '1') THEN
      reg(3) <= X1;
      reg(2) <= X2;
      reg(1) <= X3;
      reg(0) <= X4;
    END IF;
  END PROCESS;
  Q1 <= reg(3) AND read;

```

```

Q2 <= reg(2) AND read;
Q3 <= reg(1) AND read;
Q4 <= reg(0) AND read;

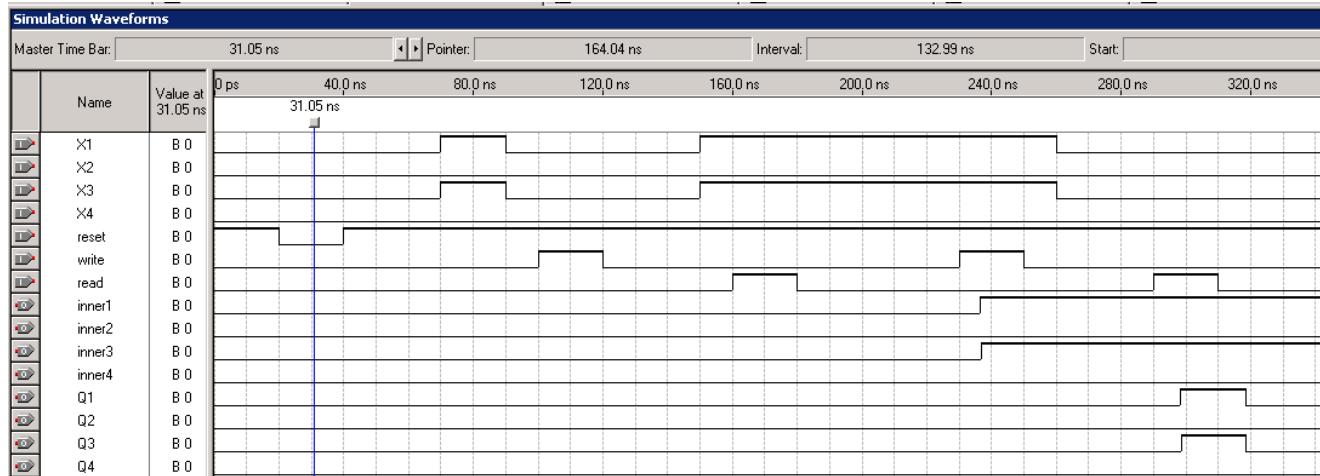
```

```
END behav;
```

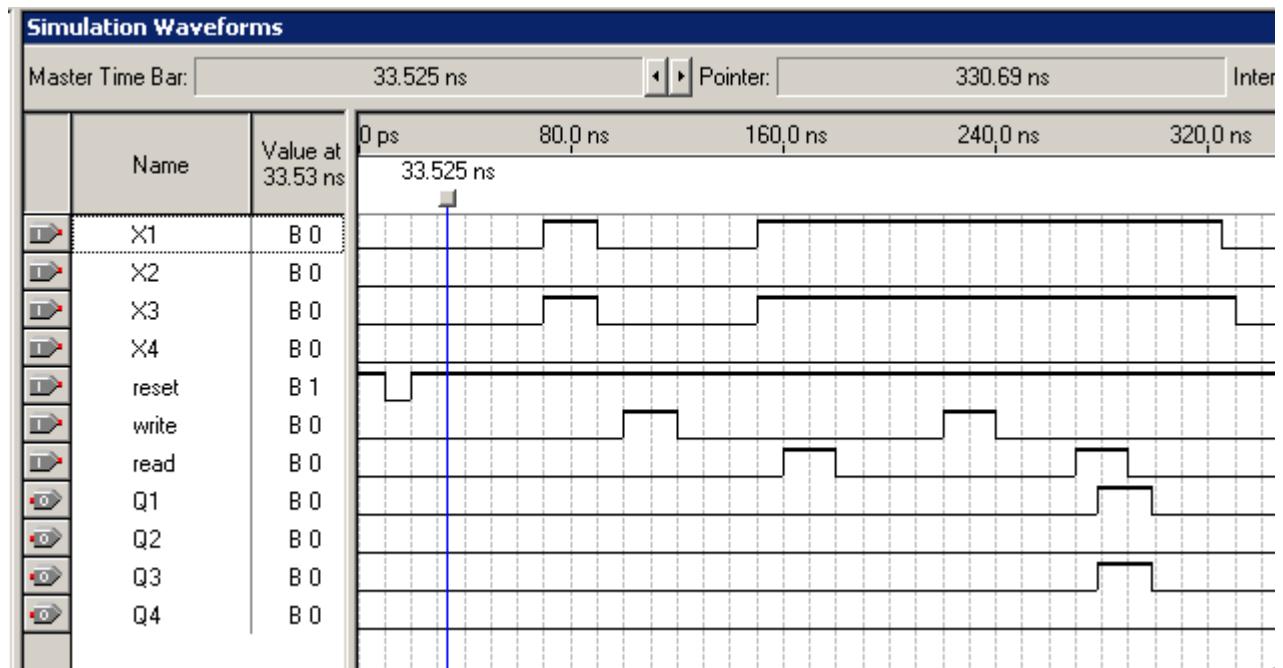
## 9.2

Откомпилировать проект и получить временные диаграммы. Сравнить их с диаграммами, полученными в п.5:

Сравним диаграмму созданного элемента в граф. ред.



и элемента написанного на vhdl

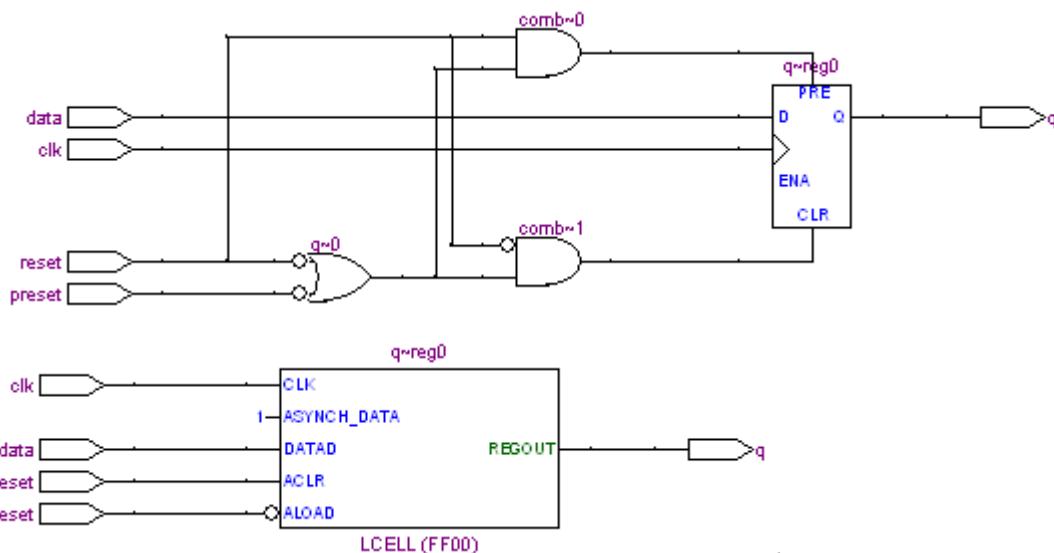


Диаграммы совпали.

## 10.1

## D-триггер с асинхронным сбросом

Flow Status	Successful - Fri Mar 11 19:15:47 2022
Quartus II Version	5.0 Build 148 04/26/2005 SJ Full Version
Revision Name	Lab04
Top-level Entity Name	dff_async
Family	Stratix
Met timing requirements	Yes
Total logic elements	1 / 10,570 (< 1 %)
Total pins	5 / 336 (1 %)
Total virtual pins	0
Total memory bits	0 / 920,448 (0 %)
DSP block 9-bit elements	0 / 48 (0 %)
Total PLLs	0 / 6 (0 %)
Total DLLs	0 / 2 (0 %)
Device	EP1S10F484C5
Timing Models	Final



## 10.2

### четырехразрядный регистр сдвига с асинхронным сбросом

Flow Status	Successful - Fri Mar 11 19:19:40 2022
Quartus II Version	5.0 Build 148 04/26/2005 SJ Full Version
Revision Name	Lab04
Top-level Entity Name	shift_reg
Family	Stratix
Met timing requirements	Yes
Total logic elements	4 / 10,570 (< 1 %)
Total pins	4 / 336 (1 %)
Total virtual pins	0
Total memory bits	0 / 920,448 (0 %)
DSP block 9-bit elements	0 / 48 (0 %)
Total PLLs	0 / 6 (0 %)
Total DLLs	0 / 2 (0 %)
Device	EP1S10F484C5
Timing Models	Final

