МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное

учреждение высшего образования

«Ижевский государственный технический университет имени М.Т. Калашникова»

(ФГБОУ ВО «ИжГТУ имени М.Т. Калашникова»)

Кафедра «Программное обеспечение»

Отчет

по лабораторной работе №3

по дисциплине «Архитектура электронно-вычислительных машин»

Выполнил:

студент группы Б04-191- 3 Р. А. Гумметов

Принял: А. Х. Аль Аккад

Ижевск 2020

**ЗАДАНИЕ 1.**

**Реализация SISO регистра в Logisim.**

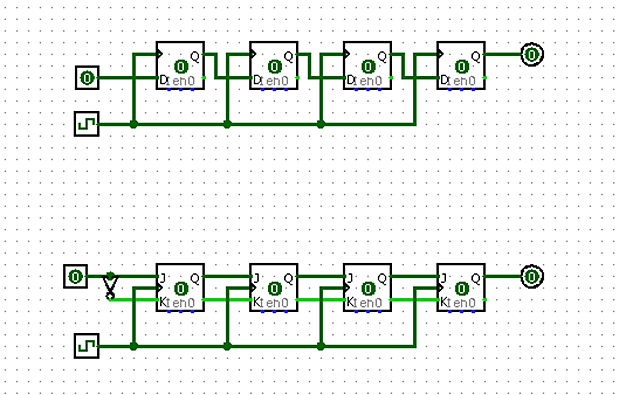
****

Рис. 1 – реализация регистра SISOв Logisim

**Реализация PISO регистра в Logisim.**

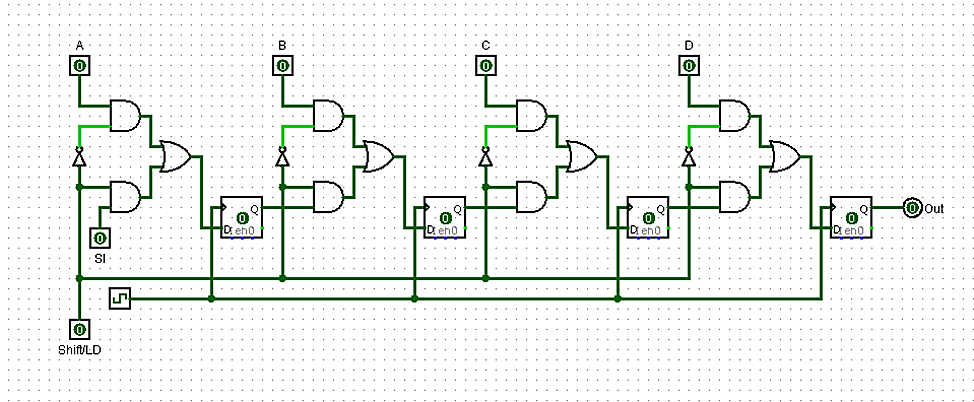


Рис. 2 – реализация регистра PISO в Logisim

**ЗАДАНИЕ 2.**

**Реализация SISO регистра в Quartus.**

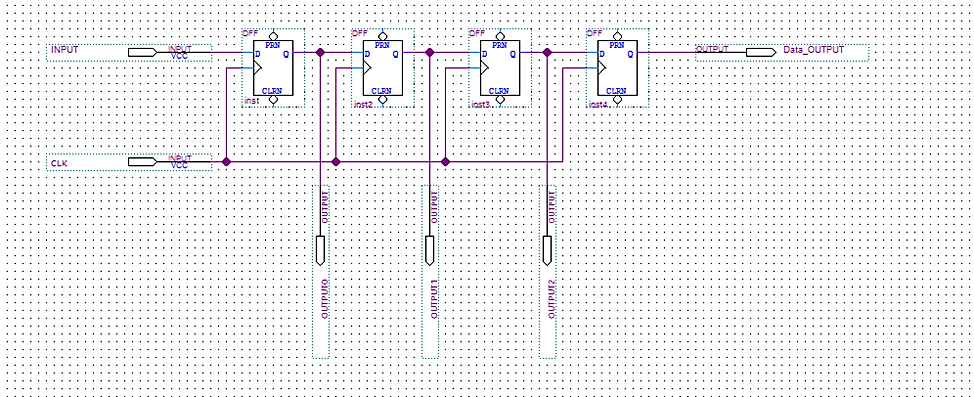
****

Рис. 3 – реализация регистра SISO в Quartus

**Полученный VHDL:**

LIBRARY ieee;

USE ieee.std\_logic\_1164.all;

LIBRARY work;

ENTITY Block1 IS

PORT

(

CLK : IN STD\_LOGIC;

INPUT : IN STD\_LOGIC;

Data\_OUTPUT : OUT STD\_LOGIC;

OUTPUT2 : OUT STD\_LOGIC;

OUTPUT1 : OUT STD\_LOGIC;

OUTPUT0 : OUT STD\_LOGIC

);

END Block1;

ARCHITECTURE bdf\_type OF Block1 IS

SIGNAL DFF\_inst : STD\_LOGIC;

SIGNAL DFF\_inst2 : STD\_LOGIC;

SIGNAL DFF\_inst3 : STD\_LOGIC;

BEGIN

OUTPUT2 <= DFF\_inst3;

OUTPUT1 <= DFF\_inst2;

OUTPUT0 <= DFF\_inst;

PROCESS(CLK)

BEGIN

IF (RISING\_EDGE(CLK)) THEN

DFF\_inst<= INPUT;

END IF;

END PROCESS;

PROCESS(CLK)

BEGIN

IF (RISING\_EDGE(CLK)) THEN

DFF\_inst2 <= DFF\_inst;

END IF;

END PROCESS;

PROCESS(CLK)

BEGIN

IF (RISING\_EDGE(CLK)) THEN

DFF\_inst3 <= DFF\_inst2;

END IF;

END PROCESS;

PROCESS(CLK)

BEGIN

IF (RISING\_EDGE(CLK)) THEN

Data\_OUTPUT<= DFF\_inst3;

END IF;

END PROCESS;

END bdf\_type;

**Симуляция дизайна:**

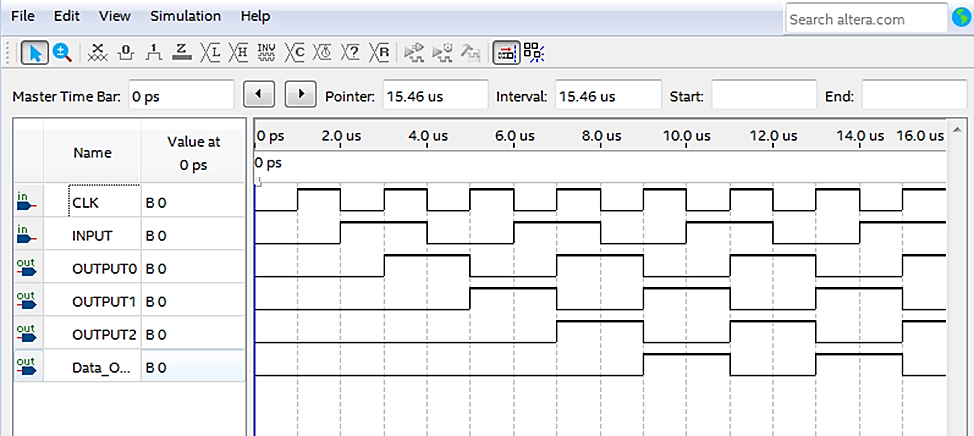


Рис. 4

**Реализация PISO регистра в Quartus.**

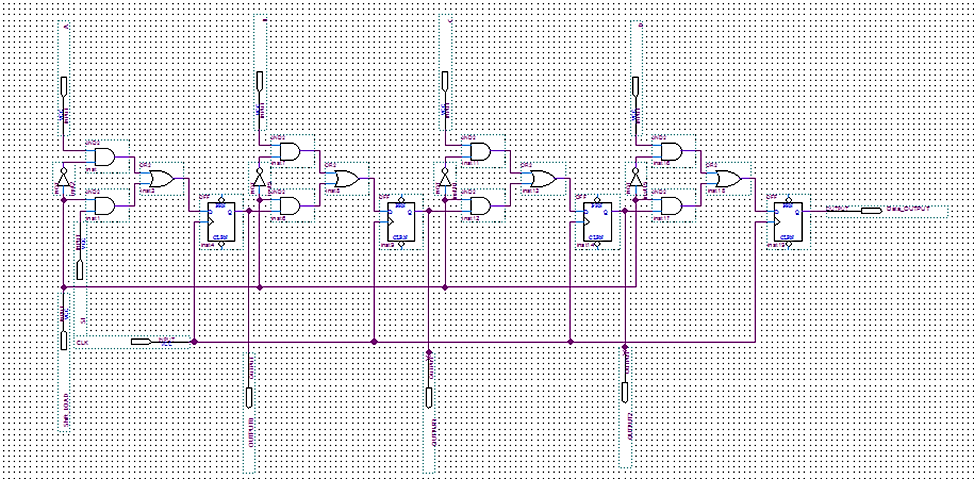
****

Рис. 5 – реализация регистра PISO в Quartus

**Полученный VHDL:**

LIBRARY ieee;

USE ieee.std\_logic\_1164.all;

LIBRARY work;

ENTITY Block1 IS

PORT

(

A : IN STD\_LOGIC;

SI : IN STD\_LOGIC;

B : IN STD\_LOGIC;

C : IN STD\_LOGIC;

D : IN STD\_LOGIC;

Shift\_LOAD : IN STD\_LOGIC;

CLK : IN STD\_LOGIC;

Data\_OUTPUT : OUT STD\_LOGIC;

OUTPUT0 : OUT STD\_LOGIC;

OUTPUT1 : OUT STD\_LOGIC;

OUTPUT2 : OUT STD\_LOGIC

);

END Block1;

ARCHITECTURE bdf\_type OF Block1 IS

SIGNAL SYNTHESIZED\_WIRE\_0 : STD\_LOGIC;

SIGNAL SYNTHESIZED\_WIRE\_1 : STD\_LOGIC;

SIGNAL DFF\_inst9 : STD\_LOGIC;

SIGNAL SYNTHESIZED\_WIRE\_2 : STD\_LOGIC;

SIGNAL SYNTHESIZED\_WIRE\_3 : STD\_LOGIC;

SIGNAL SYNTHESIZED\_WIRE\_4 : STD\_LOGIC;

SIGNAL SYNTHESIZED\_WIRE\_5 : STD\_LOGIC;

SIGNAL DFF\_inst14 : STD\_LOGIC;

SIGNAL SYNTHESIZED\_WIRE\_6 : STD\_LOGIC;

SIGNAL SYNTHESIZED\_WIRE\_7 : STD\_LOGIC;

SIGNAL SYNTHESIZED\_WIRE\_8 : STD\_LOGIC;

SIGNAL SYNTHESIZED\_WIRE\_9 : STD\_LOGIC;

SIGNAL SYNTHESIZED\_WIRE\_10 : STD\_LOGIC;

SIGNAL SYNTHESIZED\_WIRE\_11 : STD\_LOGIC;

SIGNAL DFF\_inst4 : STD\_LOGIC;

SIGNAL SYNTHESIZED\_WIRE\_12 : STD\_LOGIC;

SIGNAL SYNTHESIZED\_WIRE\_13 : STD\_LOGIC;

SIGNAL SYNTHESIZED\_WIRE\_14 : STD\_LOGIC;

SIGNAL SYNTHESIZED\_WIRE\_15 : STD\_LOGIC;

BEGIN

OUTPUT0 <= DFF\_inst4;

OUTPUT1 <= DFF\_inst9;

OUTPUT2 <= DFF\_inst14;

SYNTHESIZED\_WIRE\_10 <= A AND SYNTHESIZED\_WIRE\_0;

SYNTHESIZED\_WIRE\_9 <= Shift\_LOAD AND SI;

SYNTHESIZED\_WIRE\_1 <= NOT(Shift\_LOAD);

SYNTHESIZED\_WIRE\_3 <= C AND SYNTHESIZED\_WIRE\_1;

SYNTHESIZED\_WIRE\_2 <= Shift\_LOAD AND DFF\_inst9;

SYNTHESIZED\_WIRE\_4 <= SYNTHESIZED\_WIRE\_2 OR SYNTHESIZED\_WIRE\_3;

PROCESS(CLK)

BEGIN

IF (RISING\_EDGE(CLK)) THEN

DFF\_inst14 <= SYNTHESIZED\_WIRE\_4;

END IF;

END PROCESS;

SYNTHESIZED\_WIRE\_5 <= NOT(Shift\_LOAD);

SYNTHESIZED\_WIRE\_7 <= D AND SYNTHESIZED\_WIRE\_5;

SYNTHESIZED\_WIRE\_6 <= Shift\_LOAD AND DFF\_inst14;

SYNTHESIZED\_WIRE\_8 <= SYNTHESIZED\_WIRE\_6 OR SYNTHESIZED\_WIRE\_7;

PROCESS(CLK)

BEGIN

IF (RISING\_EDGE(CLK)) THEN

Data\_OUTPUT<= SYNTHESIZED\_WIRE\_8;

END IF;

END PROCESS;

SYNTHESIZED\_WIRE\_0 <= NOT(Shift\_LOAD);

SYNTHESIZED\_WIRE\_11 <= SYNTHESIZED\_WIRE\_9 OR SYNTHESIZED\_WIRE\_10;

PROCESS(CLK)

BEGIN

IF (RISING\_EDGE(CLK)) THEN

DFF\_inst4 <= SYNTHESIZED\_WIRE\_11;

END IF;

END PROCESS;

SYNTHESIZED\_WIRE\_12 <= NOT(Shift\_LOAD);

SYNTHESIZED\_WIRE\_13 <= Shift\_LOAD AND DFF\_inst4;

SYNTHESIZED\_WIRE\_14 <= B AND SYNTHESIZED\_WIRE\_12;

SYNTHESIZED\_WIRE\_15 <= SYNTHESIZED\_WIRE\_13 OR SYNTHESIZED\_WIRE\_14;

PROCESS(CLK)

BEGIN

IF (RISING\_EDGE(CLK)) THEN

DFF\_inst9 <= SYNTHESIZED\_WIRE\_15;

END IF;

END PROCESS;

END bdf\_type;

**Симуляция дизайна:**

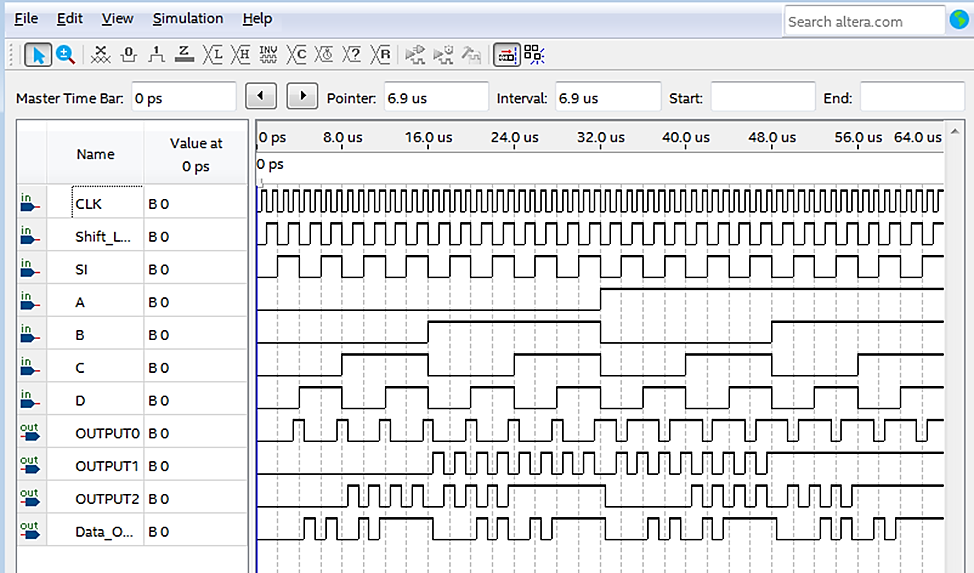


Рис. 6