МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Ижевский государственный технический университет

имени М.Т. Калашникова»

Институт «Информатика и вычислительная техника»

Кафедра «Программное обеспечение»

Лабораторная работа №2

# подисциплине «Архитектура электронно-вычислительных машин»

Выполнил:

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

Принял: А. Х. Аль Аккад

Ижевск 2020

Цель работы

C использованием приложения QuartusPrimeLiteEdition, реализовать задание для практических занятий:

- Создать диаграмму.

- Получить VHDL.

- Сделать симуляцию дизайна.

Сумматор 1 бит

1. Построение диаграммы

Диаграмма представлена на рисунке 1:

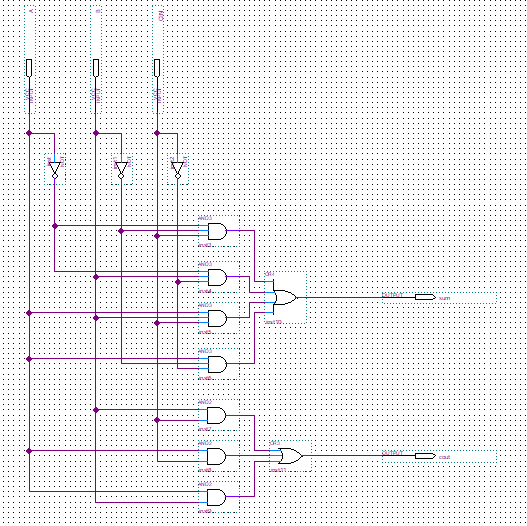


Рис. 1. Диаграмма сумматора 1 бит.

1. VHDL

-- Copyright (C) 2019 Intel Corporation. All rights reserved.

-- Your use of Intel Corporation's design tools, logic functions

-- and other software and tools, and any partner logic

-- functions, and any output files from any of the foregoing

-- (including device programming or simulation files), and any

-- associated documentation or information are expressly subject

-- to the terms and conditions of the Intel Program License

-- Subscription Agreement, the Intel Quartus Prime License Agreement,

-- the Intel FPGA IP License Agreement, or other applicable license

-- agreement, including, without limitation, that your use is for

-- the sole purpose of programming logic devices manufactured by

-- Intel and sold by Intel or its authorized distributors. Please

-- refer to the applicable agreement for further details, at

-- https://fpgasoftware.intel.com/eula.

-- PROGRAM "Quartus Prime"

-- VERSION "Version 19.1.0 Build 670 09/22/2019 SJ Lite Edition"

-- CREATED "Thu Apr 09 13:31:43 2020"

LIBRARY ieee;

USE ieee.std\_logic\_1164.all;

LIBRARY work;

ENTITY laba2 IS

PORT

(

A : IN STD\_LOGIC;

B : IN STD\_LOGIC;

CIN : IN STD\_LOGIC;

sum : OUT STD\_LOGIC;

cout : OUT STD\_LOGIC

);

END laba2;

ARCHITECTURE bdf\_type OF laba2 IS

SIGNAL SYNTHESIZED\_WIRE\_0 : STD\_LOGIC;

SIGNAL SYNTHESIZED\_WIRE\_1 : 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 SYNTHESIZED\_WIRE\_6 : STD\_LOGIC;

SIGNAL SYNTHESIZED\_WIRE\_13 : STD\_LOGIC;

SIGNAL SYNTHESIZED\_WIRE\_14 : STD\_LOGIC;

SIGNAL SYNTHESIZED\_WIRE\_15 : STD\_LOGIC;

BEGIN

SYNTHESIZED\_WIRE\_13 <= NOT(A);

SYNTHESIZED\_WIRE\_14 <= NOT(B);

sum <= SYNTHESIZED\_WIRE\_0 OR SYNTHESIZED\_WIRE\_1 OR SYNTHESIZED\_WIRE\_2 OR SYNTHESIZED\_WIRE\_3;

cout<= SYNTHESIZED\_WIRE\_4 OR SYNTHESIZED\_WIRE\_5 OR SYNTHESIZED\_WIRE\_6;

SYNTHESIZED\_WIRE\_15 <= NOT(CIN);

SYNTHESIZED\_WIRE\_0 <= SYNTHESIZED\_WIRE\_13 AND SYNTHESIZED\_WIRE\_14 AND CIN;

SYNTHESIZED\_WIRE\_3 <= SYNTHESIZED\_WIRE\_13 AND B AND SYNTHESIZED\_WIRE\_15;

SYNTHESIZED\_WIRE\_1 <= A AND B AND CIN;

SYNTHESIZED\_WIRE\_2 <= A AND SYNTHESIZED\_WIRE\_14 AND SYNTHESIZED\_WIRE\_15;

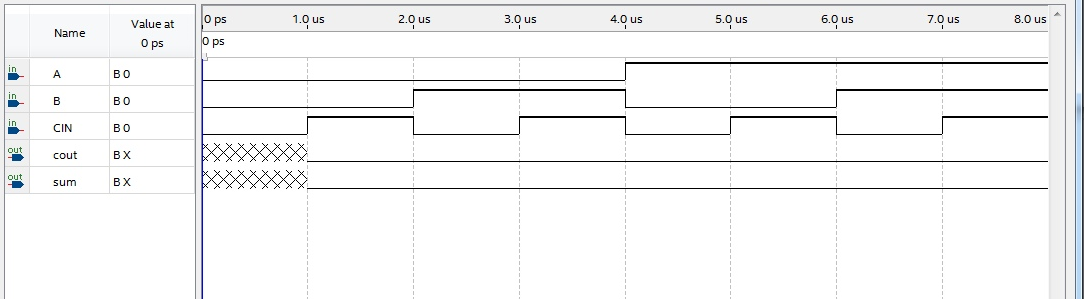
SYNTHESIZED\_WIRE\_6 <= B AND CIN;

SYNTHESIZED\_WIRE\_4 <= A AND CIN;

SYNTHESIZED\_WIRE\_5 <= A AND B;

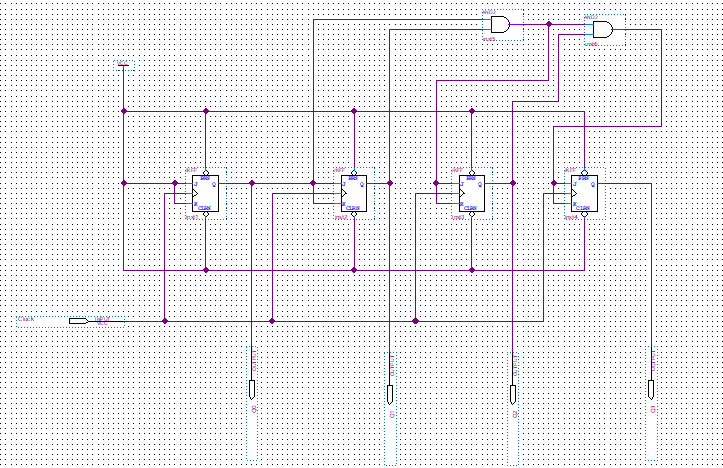
END bdf\_type;

1. Симуляция дизайна



Счетчик 4 бита

1. Построение диаграммы



1. VHDL

-- Copyright (C) 2019 Intel Corporation. All rights reserved.

-- Your use of Intel Corporation's design tools, logic functions

-- and other software and tools, and any partner logic

-- functions, and any output files from any of the foregoing

-- (including device programming or simulation files), and any

-- associated documentation or information are expressly subject

-- to the terms and conditions of the Intel Program License

-- Subscription Agreement, the Intel Quartus Prime License Agreement,

-- the Intel FPGA IP License Agreement, or other applicable license

-- agreement, including, without limitation, that your use is for

-- the sole purpose of programming logic devices manufactured by

-- Intel and sold by Intel or its authorized distributors. Please

-- refer to the applicable agreement for further details, at

-- https://fpgasoftware.intel.com/eula.

-- PROGRAM "Quartus Prime"

-- VERSION "Version 19.1.0 Build 670 09/22/2019 SJ Lite Edition"

-- CREATED "Thu Apr 09 14:16:40 2020"

LIBRARY ieee;

USE ieee.std\_logic\_1164.all;

LIBRARY work;

ENTITY project IS

PORT

(

Clock : IN STD\_LOGIC;

Q0 : OUT STD\_LOGIC;

Q1 : OUT STD\_LOGIC;

Q2 : OUT STD\_LOGIC;

Q3 : OUT STD\_LOGIC

);

END project;

ARCHITECTURE bdf\_type OF project IS

SIGNAL SYNTHESIZED\_WIRE\_15 : STD\_LOGIC;

SIGNAL SYNTHESIZED\_WIRE\_16 : STD\_LOGIC;

SIGNAL SYNTHESIZED\_WIRE\_17 : STD\_LOGIC;

SIGNAL SYNTHESIZED\_WIRE\_18 : STD\_LOGIC;

SIGNAL JKFF\_inst2 : STD\_LOGIC;

SIGNAL JKFF\_inst3 : STD\_LOGIC;

BEGIN

Q0 <= SYNTHESIZED\_WIRE\_16;

Q1 <= JKFF\_inst2;

Q2 <= JKFF\_inst3;

SYNTHESIZED\_WIRE\_15 <= '1';

PROCESS(Clock,SYNTHESIZED\_WIRE\_15,SYNTHESIZED\_WIRE\_15)

VARIABLE synthesized\_var\_for\_SYNTHESIZED\_WIRE\_16 : STD\_LOGIC;

BEGIN

IF (SYNTHESIZED\_WIRE\_15 = '0') THEN

synthesized\_var\_for\_SYNTHESIZED\_WIRE\_16 := '0';

ELSIF (SYNTHESIZED\_WIRE\_15 = '0') THEN

synthesized\_var\_for\_SYNTHESIZED\_WIRE\_16 := '1';

ELSIF (RISING\_EDGE(Clock)) THEN

synthesized\_var\_for\_SYNTHESIZED\_WIRE\_16 := (NOT(synthesized\_var\_for\_SYNTHESIZED\_WIRE\_16) AND SYNTHESIZED\_WIRE\_15) OR (synthesized\_var\_for\_SYNTHESIZED\_WIRE\_16 AND (NOT(SYNTHESIZED\_WIRE\_15)));

END IF;

SYNTHESIZED\_WIRE\_16 <= synthesized\_var\_for\_SYNTHESIZED\_WIRE\_16;

END PROCESS;

PROCESS(Clock,SYNTHESIZED\_WIRE\_15,SYNTHESIZED\_WIRE\_15)

VARIABLE synthesized\_var\_for\_JKFF\_inst2 : STD\_LOGIC;

BEGIN

IF (SYNTHESIZED\_WIRE\_15 = '0') THEN

synthesized\_var\_for\_JKFF\_inst2 := '0';

ELSIF (SYNTHESIZED\_WIRE\_15 = '0') THEN

synthesized\_var\_for\_JKFF\_inst2 := '1';

ELSIF (RISING\_EDGE(Clock)) THEN

synthesized\_var\_for\_JKFF\_inst2 := (NOT(synthesized\_var\_for\_JKFF\_inst2) AND SYNTHESIZED\_WIRE\_16) OR (synthesized\_var\_for\_JKFF\_inst2 AND (NOT(SYNTHESIZED\_WIRE\_16)));

END IF;

JKFF\_inst2 <= synthesized\_var\_for\_JKFF\_inst2;

END PROCESS;

PROCESS(Clock,SYNTHESIZED\_WIRE\_15,SYNTHESIZED\_WIRE\_15)

VARIABLE synthesized\_var\_for\_JKFF\_inst3 : STD\_LOGIC;

BEGIN

IF (SYNTHESIZED\_WIRE\_15 = '0') THEN

synthesized\_var\_for\_JKFF\_inst3 := '0';

ELSIF (SYNTHESIZED\_WIRE\_15 = '0') THEN

synthesized\_var\_for\_JKFF\_inst3 := '1';

ELSIF (RISING\_EDGE(Clock)) THEN

synthesized\_var\_for\_JKFF\_inst3 := (NOT(synthesized\_var\_for\_JKFF\_inst3) AND SYNTHESIZED\_WIRE\_17) OR (synthesized\_var\_for\_JKFF\_inst3 AND (NOT(SYNTHESIZED\_WIRE\_17)));

END IF;

JKFF\_inst3 <= synthesized\_var\_for\_JKFF\_inst3;

END PROCESS;

PROCESS(Clock,SYNTHESIZED\_WIRE\_15,SYNTHESIZED\_WIRE\_15)

VARIABLE synthesized\_var\_for\_Q3 : STD\_LOGIC;

BEGIN

IF (SYNTHESIZED\_WIRE\_15 = '0') THEN

synthesized\_var\_for\_Q3 := '0';

ELSIF (SYNTHESIZED\_WIRE\_15 = '0') THEN

synthesized\_var\_for\_Q3 := '1';

ELSIF (RISING\_EDGE(Clock)) THEN

synthesized\_var\_for\_Q3 := (NOT(synthesized\_var\_for\_Q3) AND SYNTHESIZED\_WIRE\_18) OR (synthesized\_var\_for\_Q3 AND (NOT(SYNTHESIZED\_WIRE\_18)));

END IF;

Q3 <= synthesized\_var\_for\_Q3;

END PROCESS;

SYNTHESIZED\_WIRE\_17 <= SYNTHESIZED\_WIRE\_16 AND JKFF\_inst2;

SYNTHESIZED\_WIRE\_18 <= SYNTHESIZED\_WIRE\_17 AND JKFF\_inst3;

END bdf\_type;

1. Симуляция дизайна

