|  |
| --- |
|  |
| МИНОБРНАУКИ РОССИИ |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА - Российский технологический университет»**  **РТУ МИРЭА** |

Институт Информационных Технологий

Кафедра Вычислительной Техники (ВТ)

**ОТЧЁТ ПО ПРАКТИЧЕСККОЙ РАБОТЕ №3**

«САПР ПЛИС. Язык описания аппаратуры»

по дисциплине

«Архитектура вычислительных машин и систем»

|  |  |
| --- | --- |
| Выполнил студент группы  ИВБО-11-23 | Туктаров Т.А. |
| Принял ассистент кафедры ВТ | Дуксина И.И. |
| Практическая работа выполнена | «2» октября 2024 г. |
| «Зачтено» | «2» октября 2024 г. |

Москва 2024

АННОТАЦИЯ

Данная работа включает в себя 1 рисунок, 1 таблицу, 1 листинг, 2 формулы. Количество страниц в работе — 10.

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 4](#_Toc180104997)

[Ход работы 5](#_Toc180104998)

[2.1 Постановка задачи 5](#_Toc180104999)

[2.2 Восстановление таблицы истинности 5](#_Toc180105000)

[2.3 Построение СДНФ и МДНФ 6](#_Toc180105001)

[2.4 Реализация при помощи Verilog 6](#_Toc180105002)

[2.5 Верификация 7](#_Toc180105003)

[ЗАКЛЮЧЕНИЕ 9](#_Toc180105004)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 10](#_Toc180105005)

ВВЕДЕНИЕ

Vivado - система автоматизированного проектирования (САПР). Разработана компанией Xilinx.

Стиль описания устройств в Verilog не зависит от конкретной технологии.

Преимущество : Высокая мобильность устройств (простой перенос устройств на другую элементарную базу)

Недостатки : Отсутствие полного учёта всех специфических особенностей конкретной элементной базы

Уровни описания : структурный, поведенческий (!) .

Модуль основной элемент описания Как правило, для каждого модуля создаётся отдельный файл с на языке Verilog расширением .v .

input - ключевое слово для определения входного порта.

output - ключевое слово для определения выходного порта.

inout - ключевое слово для определения двунаправленного порта.

Для описания событий используются два ключевых слова: posedge и negedge.

posedge используется для связи события с изменением значения сигнала с 0 на 1.

negedge используется для связи события с изменением значения сигнала с 1 на 0.[1][2]

Ход работы

## 2.1 Постановка задачи

Для функций заданной в векторном виде соответствующей индивидуальному варианту необходимо реализовать СДНФ и МДНФ (для функции, минимизированной при помощи карты Карно) при помощи языка описания аппаратуры Verilog средствами САПР Vivado. Произвести верификацию полученных схем. Заданная логическая функция: 478E9C16.

## 2.2 Восстановление таблицы истинности

Имея логическую функцию в векторном виде 478E9C16 воссоздадим таблицу истинности(Таблица 2.1).

Таблица 2.1 — Таблица истинности для логической функции

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| X1 | X2 | X3 | X4 | X5 | F |
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 1 |
| 0 | 0 | 1 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 | 1 |

Продолжение Таблицы 2.1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1 | 0 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 | 1 | 1 |
| 1 | 0 | 1 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 1 | 0 | 1 |

## 2.3 Построение СДНФ и МДНФ

Построим СДНФ исходной функции по таблице истинности(Формула 2.1)

Минимизируем полученную СДНФ и получим следующую МДНФ(Формула 2.2)

## 2.4 Реализация при помощи Verilog

При помощи языка описания аппаратуры Verilog средствами САПР Vivado реализуем СДНФ и МДНФ для исходной функции(Листинг 2.1) [3]

Листинг 2.1 — Модуль main.v

`timescale 1ns / 1ps

module main(

input [4:0] x,

output f\_SDNF, f\_MDNF

);

assign f\_SDNF =

(~x[4] && ~x[3] && ~x[2] && ~x[1] && x[0]) ||

(~x[4] && ~x[3] && x[2] && ~x[1] && x[0]) ||

(~x[4] && ~x[3] && x[2] && x[1] && ~x[0]) ||

(~x[4] && ~x[3] && x[2] && x[1] && x[0]) ||

(~x[4] && x[3] && ~x[2] && ~x[1] && ~x[0]) ||

(~x[4] && x[3] && x[2] && ~x[1] && ~x[0]) ||

(~x[4] && x[3] && x[2] && ~x[1] && x[0]) ||

(~x[4] && x[3] && x[2] && x[1] && ~x[0]) ||

(x[4] && ~x[3] && ~x[2] && ~x[1] && ~x[0]) ||

(x[4] && ~x[3] && ~x[2] && x[1] && x[0]) ||

(x[4] && ~x[3] && x[2] && ~x[1] && ~x[0]) ||

(x[4] && ~x[3] && x[2] && ~x[1] && x[0]) ||

(x[4] && x[3] && ~x[2] && x[1] && x[0]) ||

(x[4] && x[3] && x[2] && ~x[1] && x[0]) ||

(x[4] && x[3] && x[2] && x[1] && ~x[0]);

assign f\_MDNF =

(~x[4] && ~x[3] && ~x[1] && x[0]) ||

(~x[4] && x[3] && ~x[1] && ~x[0]) ||

(x[4] && ~x[3] && ~x[1] && ~x[0]) ||

(x[3] && x[2] && x[1] && ~x[0]) ||

(x[4] && ~x[2] && x[1] && x[0]) ||

(~x[4] && ~x[3] && x[2] && x[1]) ||

(x[2] && ~x[1] && x[0]);

endmodule

## 2.5 Верификация

Произведем верификацию модуля main.v, для это создадим модуль testbench.v(Листинг 2.2).

Листинг 2.2 — Модуль testbench.v

`timescale 1ns / 1ps

module testbench();

reg [4:0] args;

reg clk;

wire sdnf, mdnf;

reg [0:31] reference\_reg, error\_reg\_sdnf, error\_reg\_mdnf;

initial begin

reference\_reg = 32'h478E9C16;

args = 0;

clk = 0;

error\_reg\_sdnf = 0;

error\_reg\_mdnf = 0;

end

always #10 clk = ~clk;

Продолжение Листинга 2.2

always @(posedge clk) begin

error\_reg\_sdnf[args] <= (sdnf ~^ reference\_reg[args]);

error\_reg\_mdnf[args] <= (mdnf ~^ reference\_reg[args]);

args <= args + 1;

if (args == 32'h478E9C16)

$finish;

end

main mod\_f(

.x(args),

.f\_SDNF(sdnf),

.f\_MDNF(mdnf)

);

endmodule

Результат верификации представлен на Рисунке 2.1

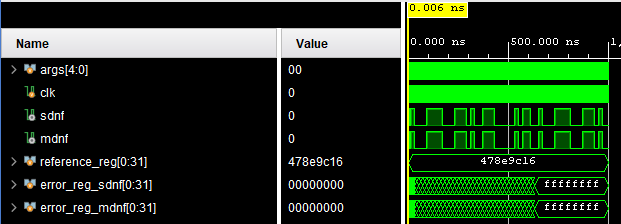


Рисунок 2.1 — Результат верификации.

ЗАКЛЮЧЕНИЕ

В данной работе были реализованы СДНФ и МДНФ при помощи языка описания аппаратуры Verilog средствами САПР Vivado для функции, заданной в векторном виде соответствующей индивидуальному варианту Произведена верификация полученных схем.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Методические указания по ПР № 3 — URL: <https://online-edu.mirea.ru/mod/resource/view.php?id=409135>.

2. Смирнов С.С. Информатика [Электронный ресурс]: Методические указания по выполнению практических и лабораторных работ / С.С. Смирнов — М., МИРЭА — Российский технологический университет, 2018. — 1 электрон. опт. диск (CD-ROM)

3. Тарасов И.Е. ПЛИС Xilinx. Языки описания аппаратуры VHDL и Verilog, САПР, приемы проектирования. — М.: Горячая линия — Телеком, 2021. — 538 с.: ил.

4. Жемчужникова Т.Н. Конспект лекций по дисциплине «Архитектура вычислительных машин и систем» — URL: [https://drive.google.com/file/d/12OAi2\_axJ6mRr4hCbXs-mYs8Kfp4YEfj/view?us p=sharing.](https://drive.google.com/file/d/12OAi2_axJ6mRr4hCbXs-mYs8Kfp4YEfj/view?us%20p=sharing.)