



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«МИРЭА - Российский технологический университет»

РТУ МИРЭА

Институт Информационных Технологий
Кафедра Вычислительной Техники (ВТ)

ОТЧЁТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ № 3

«САПР ПЛИС. Язык описание аппаратуры»

по дисциплине

«Архитектура вычислительных машин и систем»

Выполнил студент группы
ИВБО-11-23

Туктаров Т.А

Принял ассистент кафедры ВТ

Дуксина И.И.

Лабораторная работа выполнена

« __ » _____ 2024 г.

«Зачтено»

« __ » _____ 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
Ход работы.....	5
2.1 Постановка задачи.....	5
2.2 Восстановление таблицы истинности.....	5
2.3 Построение СДНФ и МДНФ.....	6
2.4 Реализация при помощи Verilog.....	6
2.5 Верификация.....	7
ЗАКЛЮЧЕНИЕ	9
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	10

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 ХОД РАБОТЫ

1.1 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.

1.2 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	0
1	1	1	1	0	1
1	1	1	1	0	1

1.3 Построение СДНФ и МДНФ

Построим СДНФ исходной функции по таблице истинности(Формула 1.1)

$$\begin{aligned}
 F_{\text{сднф}} = & \overline{x_0}x_1x_2x_3x_4 + \overline{x_0}x_1x_2x_3\overline{x_4} + \overline{x_0}x_1x_2x_3x_4 + \overline{x_0}x_1x_2\overline{x_3}x_4 + \overline{x_0}x_1\overline{x_2}x_3x_4 \\
 & + \overline{x_0}x_1\overline{x_2}x_3\overline{x_4} + \overline{x_0}x_1\overline{x_2}x_3x_4 + \overline{x_0}x_1x_2x_3x_4 + x_0\overline{x_1}x_2x_3x_4 \\
 & + x_0\overline{x_1}x_2x_3\overline{x_4} + x_0\overline{x_1}x_2x_3\overline{x_4} + x_0\overline{x_1}x_2x_3x_4 + x_0x_1\overline{x_2}x_3x_4 \\
 & + x_0x_1\overline{x_2}x_3x_4 + x_0x_1\overline{x_2}x_3\overline{x_4} + x_0x_1x_2\overline{x_3}x_4. \quad (1.1)
 \end{aligned}$$

Минимизируем полученную СДНФ и получим следующую МДНФ (Формула 1.2)

$$\begin{aligned}
 F_{\text{мднф}} = & \overline{x_1}x_2x_4x_5 + \overline{x_1}x_2\overline{x_4}x_5 + x_1\overline{x_2}x_4x_5 + x_2x_3x_4\overline{x_5} + x_1\overline{x_3}x_4x_5 + \\
 & \overline{x_1}x_2x_3x_4 + x_3\overline{x_4}x_5 \quad (1.2)
 \end{aligned}$$

1.4 Реализация при помощи Verilog

При помощи языка описания аппаратуры Verilog средствами САПР Vivado реализуем СДНФ и МДНФ для исходной функции(Листинг 1.1)

Листинг 1.1 — Модуль реализации СДНФ и МДНФ

```
`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
```

1.5 Верификация

Произведем верификацию модуля `main.v`, для это создадим модуль `testbench.v`(Листинг 1.2).

Листинг 1.2 — Модуль верификации

```
`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 1.2

```
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 1.1

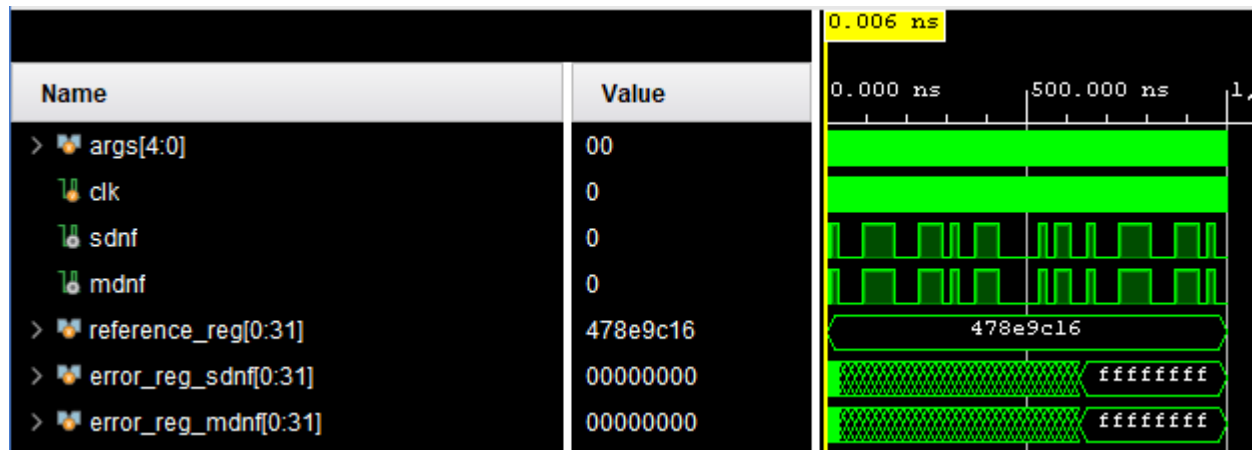


Рисунок 1.1 — Результат верификации.

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 № 1 — URL: <https://online-edu.mirea.ru/mod/resource/view.php?id=405132> (Дата обращения: 23.09.2024).
2. Методические указания по ПР № 2 — URL: <https://online-edu.mirea.ru/mod/resource/view.php?id=409130> (Дата обращения: 23.09.2024).
3. 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).
4. Тарасов И.Е. ПЛИС Xilinx. Языки описания аппаратуры VHDL и Verilog, САПР, приемы проектирования. — М.: Горячая линия — Телеком, 2021. — 538 с.: ил.
5. Антик М.И. Дискретная математика [Электронный ресурс]: Учебное пособие / Антик М.И., Казанцева Л.В. — М.: МИРЭА — Российский технологический университет, 2018 — 1 электрон. опт. диск (CD-ROM).
6. Антик М.И. Математическая логика и программирование в логике [Электронный ресурс]: Учебное пособие / Антик М.И., Бражникова Е.В.— М.: МИРЭА – Российский технологический университет, 2018. — 1 электрон. опт. диск (CD-ROM).
7. Жемчужникова Т.Н. Конспект лекций по дисциплине «Архитектура вычислительных машин и систем» — URL: https://drive.google.com/file/d/12OAi2_axJ6mRr4hCbXs-mYs8Kfp4YEfj/view?usp=sharing (Дата обращения: 23.09.2024).
8. Антик М.И. Теория автоматов в проектировании цифровых схем [Электронный ресурс]: Учебное пособие / Антик М.И., Казанцева Л.В. — М.: МИРЭА – Российский технологический университет, 2020. — 1 электрон. опт. диск (CD-ROM).
9. Орлов С.А. Организация ЭВМ и систем: Учебник для вузов. 4-е изд. — СПб.: Питер, 2018. — 688 с.: ил.