



УНИВЕРСИТЕТ ИТМО

Факультет программной инженерии и компьютерной техники

Направление подготовки 09.03.04 Программная инженерия

Дисциплина «Функциональная схемотехника»

Лабораторная работа №1

Вариант 2

Выполнили:

Чмурова М.В.

Кузенина В.Н.

P3332

Преподаватель:

Кустарёв П.В.

Санкт-Петербург

2024 г.

Оглавление

Цель работы.....	3
Задание	3
1.1 Схема разработанного вентиля.....	4
1.2 Символ вентиля и схема тестирования.....	4
1.3 Временная диаграмма процесса тестирования вентиля.....	5
1.4 Результат измерения задержки распространения сигнала через вентиль	6
1.5 Максимальная частота работы вентиля.....	6
1.6 Схема разработанного БОЭ	7
1.7 Символ разработанного БОЭ и схема тестирования	8
1.7 Временные диаграммы разработанного БОЭ	9
1.8 Результат измерения задержки распространения сигнала через БОЭ.....	10
1.9 Максимальная частота.....	10
2.1. Код разработанного модуля БОЭ	11
2.2. Код разработанного тестового окружения БОЭ	12
2.3. Временная диаграмма процесса тестирования БОЭ	14
3. Выводы по работе	15

Цель работы

1. Получить базовые знания о принципах построения цифровых интегральных схем с использованием технологии КМОП.
2. Познакомиться с технологией SPICE-моделирования схем на транзисторах.
3. Получить навыки описания схем базовых операционных элементов (БОЭ) комбинационного типа на вентиляном уровне с использованием языка описания аппаратуры Verilog HDL.

Задание

Разработать БОЭ – полный четырехразрядный компаратор с использованием логического базиса NAND

1.1 Схема разработанного вентиля

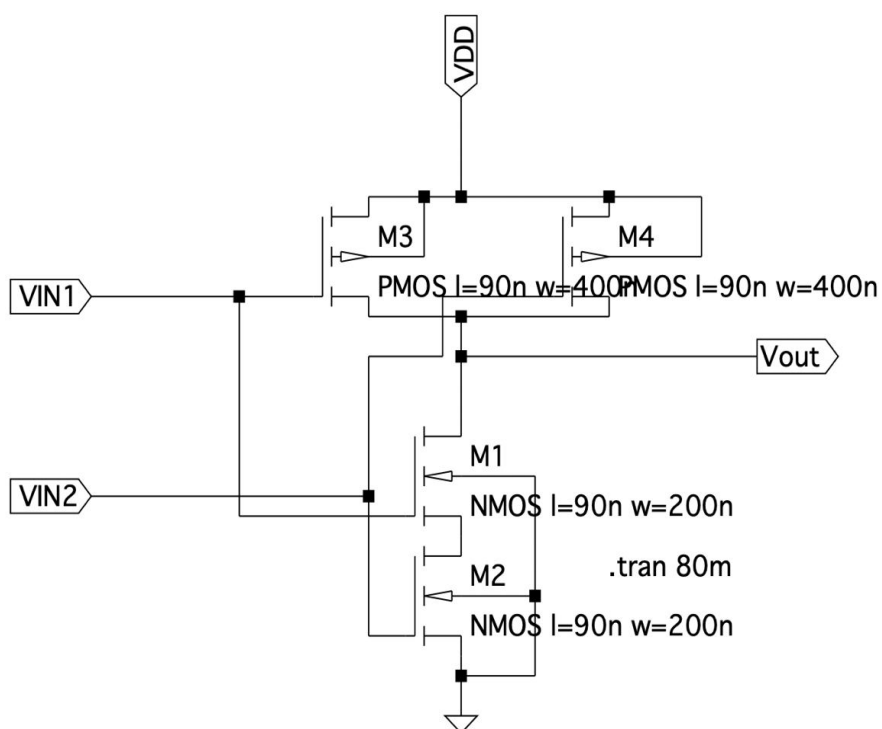


Рисунок 1. Схема вентиля

1.2 Символ вентиля и схема тестирования

Рисунок полученного символа вентиля:

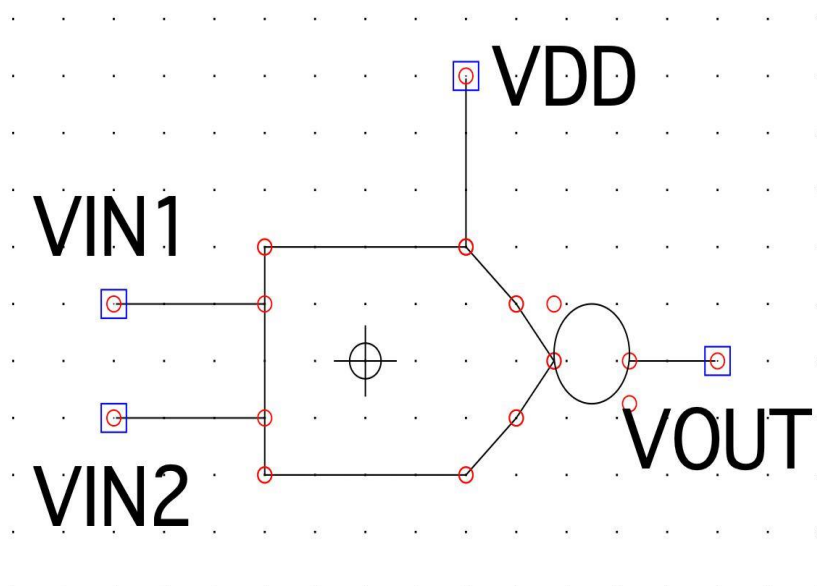


Рисунок 2. Символ вентиля

Схема, используемая для тестирования:

```
.include 90nm_bulk.txt
.tran 0 4n 0 1p
```

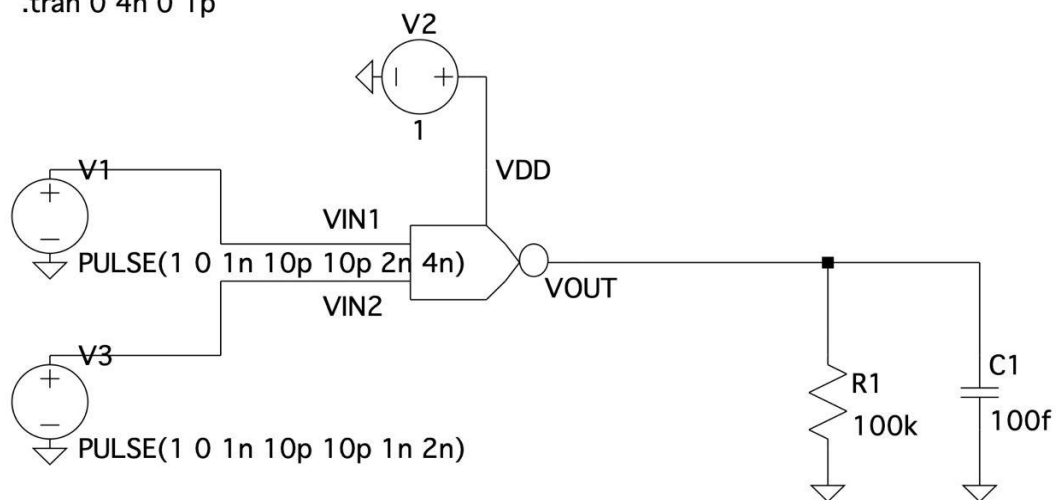


Рисунок 3. Схема тестирования

1.3 Временная диаграмма процесса тестирования вентиля

Временная диаграмма, полученная в процессе тестирования вентиля NAND:

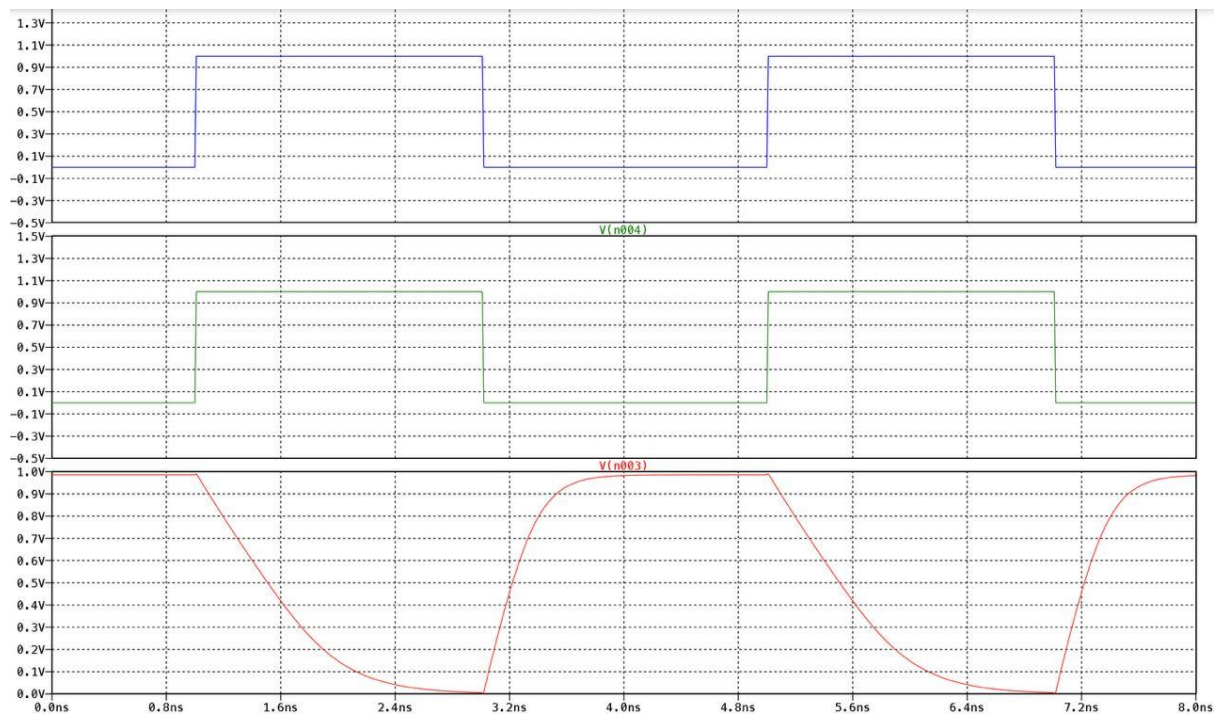


Рисунок 4. Временная диаграмма тестирования вентиля NAND

Согласно временной диаграмме заметно, что элемент корректно выполняет операцию NAND:

V(002) - in1	V(004) - in2	V(003) - out
1	1	0
0	0	1
0	1	1
1	0	1

1.4 Результат измерения задержки распространения сигнала через вентиль

Расчёт задержки по фронту и спаду:

$$d_o = 3.236 - 3.033 = 0.203 \text{ [нс]}$$

$$d_c = 5.530 - 5.018 = 0.512 \text{ [нс]}$$

Нахождение средней задержки:

$$d = \frac{0.203 + 0.512}{2} = 0.3575 \text{ [нс]}$$

1.5 Максимальная частота работы вентиля

Нахождение частоты вентиля:

$$f = \frac{1}{0.3575 \cdot 10^{-9}} = 2.797 \text{ [ГГц]}$$

1.6 Схема разработанного БОЭ

Схема разработанного БОЭ была написана с использованием вспомогательных логических элементов. Таких как:

AND:

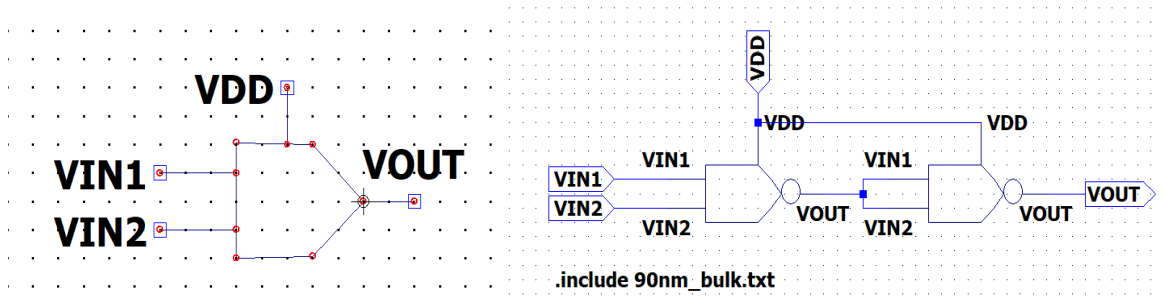


Рисунок 5. Схема AND

NOR:

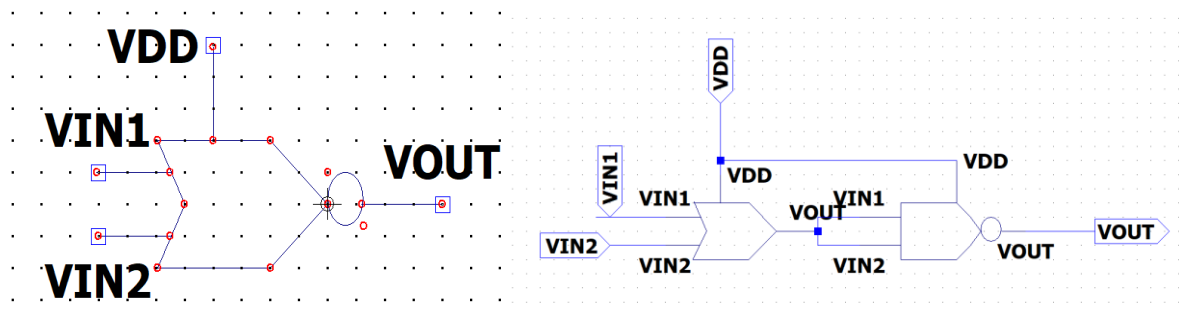


Рисунок 6. Схема NOR

XNOR:

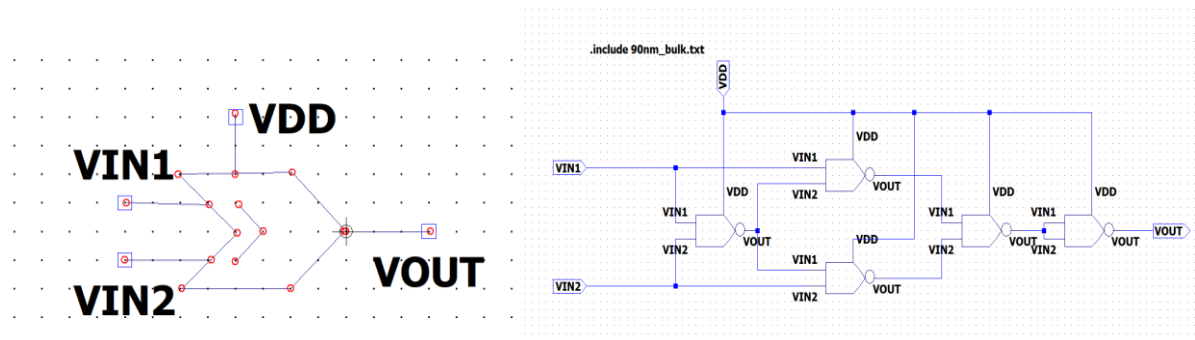


Рисунок 7. Схема XNOR

Полученная схема:

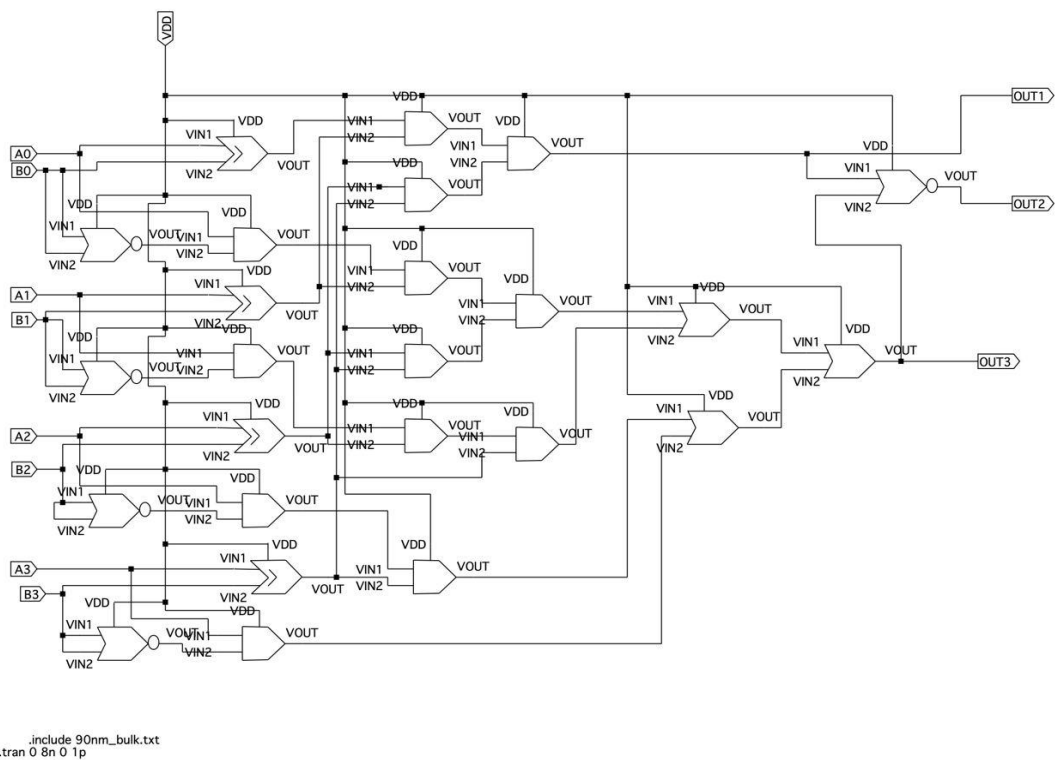


Рисунок 8. Схема компаратора

1.7 Символ разработанного БОЭ и схема тестирования

Полученный символ разработанного БОЭ:

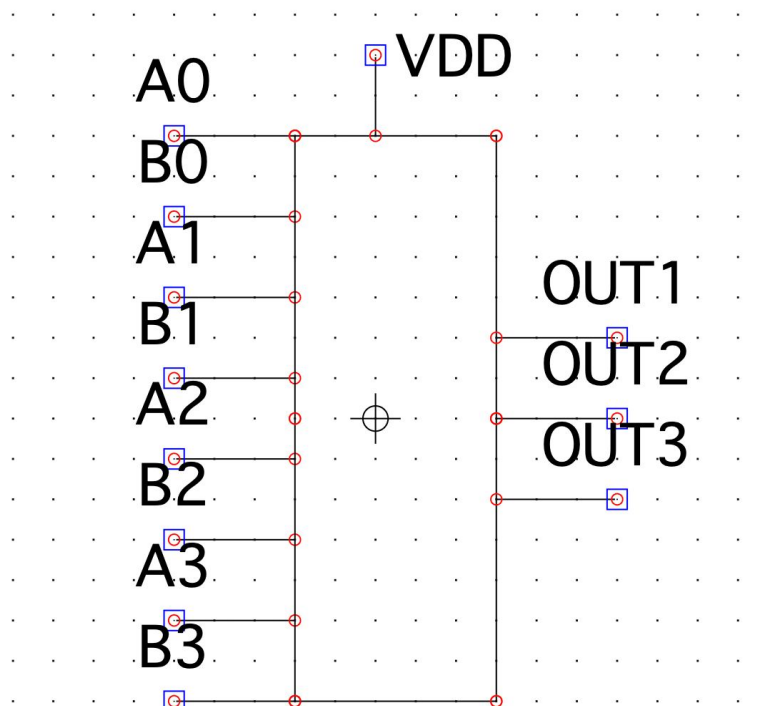


Рисунок 9. Символ БОЭ

Полученная схема тестирования разработанного БОЭ:

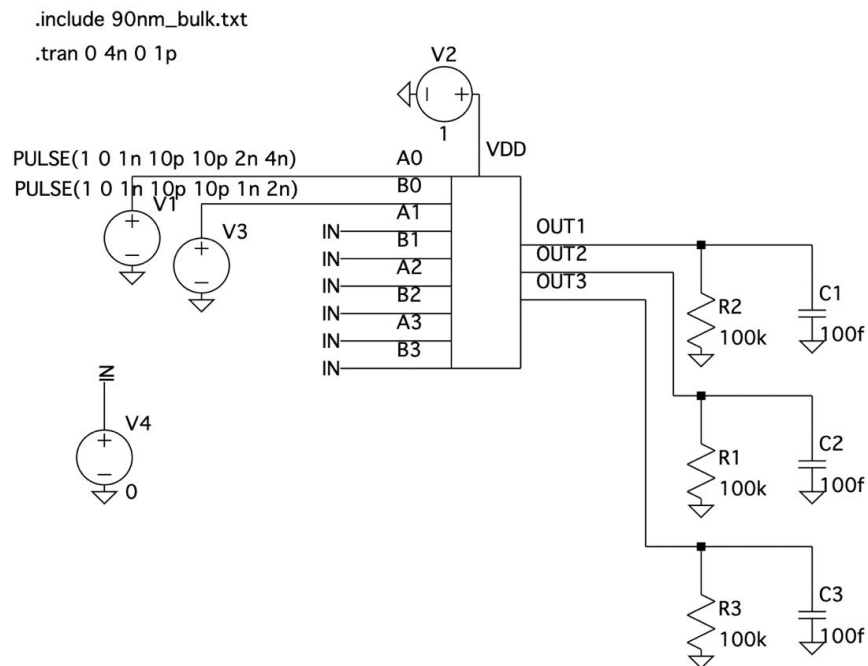


Рисунок 10. Схема тестирования разработанного БОЭ

1.7 Временные диаграммы разработанного БОЭ

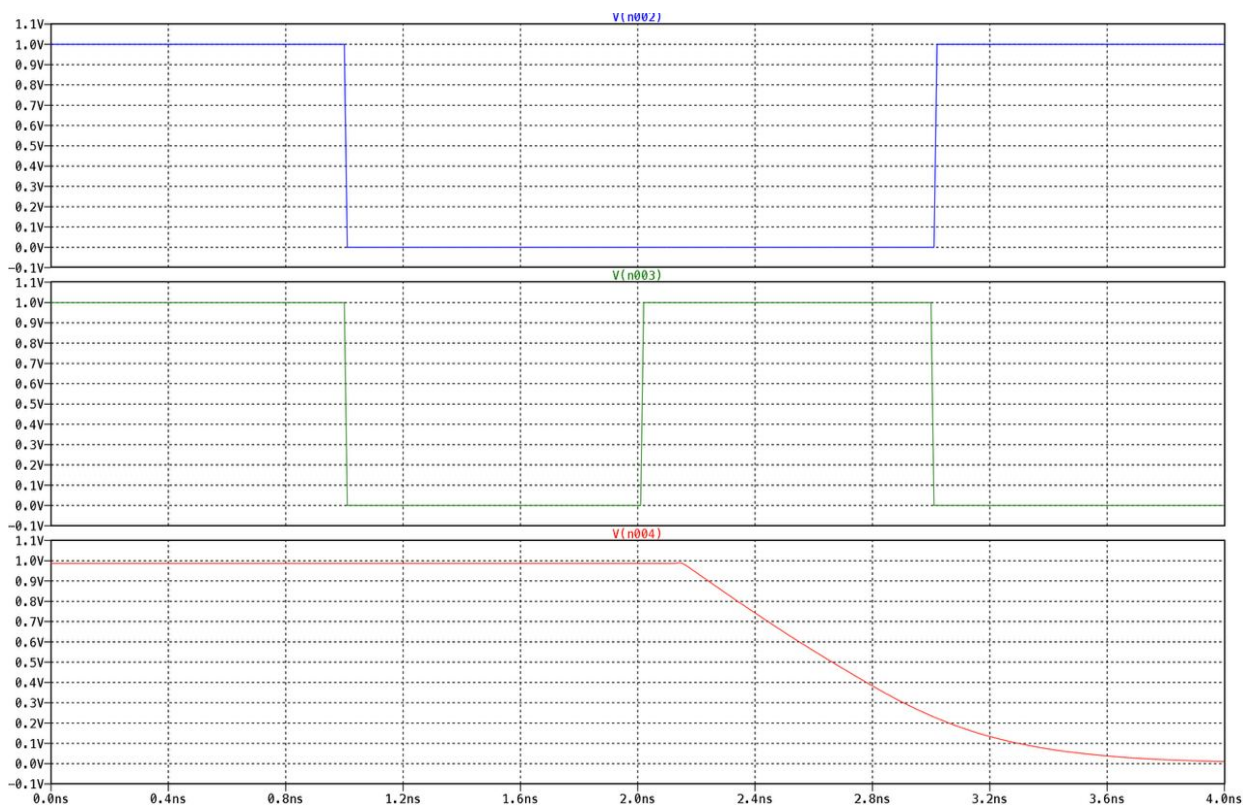


Рисунок 11. Временная диаграмма разработанного БОЭ

Согласно временной диаграмме видим, что схема корректно выполняется (V(004) - выход проверяющий на равенство):

V(002) - in1	V(003) - in2	V(004) - out
1	1	1
0	0	1
0	1	0
0	0	0

1.8 Результат измерения задержки распространения сигнала через БОЭ

Расчёт задержки по фронту и спаду:

$$d_c = 2.766 - 2.025 = 0.741 \text{ [нс]}$$

$$d_\phi = 4.383 - 4.043 = 0.343 \text{ [нс]}$$

Нахождение средней задержки:

$$d = \frac{0.741 + 0.343}{2} = 0.542 \text{ [нс]}$$

1.9 Максимальная частота

Нахождение частоты вентиля:

$$f = \frac{1}{0.542 \cdot 10^{-9}} = 1.845 \text{ [ГГц]}$$

2.1. Код разработанного модуля БОЭ

```
`timescale 1ns / 1ps

module comparator(
    input a0,
    input a1,
    input a2,
    input a3,

    input b0,
    input b1,
    input b2,
    input b3,

    output a_greater_b,
    output a_equals_b,
    output a_smaller_b
);

    wire not_b0, not_b1, not_b2, not_b3;
    wire xnor_0, xnor_1, xnor_2, xnor_3;
    wire an0, an1, an2, an3;
    wire and_greater_0, and_greater_1, and_greater_2;

    not(not_b0, b0);
    not(not_b1, b1);
    not(not_b2, b2);
    not(not_b3, b3);

    xnor(xnor_0, a0, b0);
    xnor(xnor_1, a1, b1);
    xnor(xnor_2, a2, b2);
    xnor(xnor_3, a3, b3);

    and(and_0, a0, not_b0);
    and(and_1, a1, not_b1);
    and(and_2, a2, not_b2);
    and(and_3, a3, not_b3);

    and(and_greater_0, and_0, xnor_1, xnor_2, xnor_3);
```

```

and(and_greater_1, and_1, xnor_2, xnor_3);
and(and_greater_2, and_2, xnor_3);

and(a_equals_b, xnor_0, xnor_1, xnor_2, xnor_3);
or(a_greater_b, and_greater_0, and_greater_1, and_greater_2, and_3);
nor(a_smaller_b, a_equals_b, a_greater_b);
endmodule

```

2.2. Код разработанного тестового окружения БОЭ

```

`timescale 1ns / 1ps

module comparator_tb;

    reg a0_in, a1_in, a2_in, a3_in, b0_in, b1_in, b2_in, b3_in;
    wire result_equals_out, result_smaller_out, result_greater_out;

    comparator comparator_1(
        .a0(a0_in),
        .a1(a1_in),
        .a2(a2_in),
        .a3(a3_in),

        .b0(b0_in),
        .b1(b1_in),
        .b2(b2_in),
        .b3(b3_in),

        .a_equals_b(result_equals_out),
        .a_greater_b(result_greater_out),
        .a_smaller_b(result_smaller_out)
    );

    integer i, j;
    reg [3:0] test_a, test_b;
    reg expected_eq, expected_gr, expected_low;

    initial begin
        $display("Starting comparator test...");
    end

```

```

for (i = 0; i < 16; i = i + 1) begin
    test_a = i;
    a0_in = test_a[0];
    a1_in = test_a[1];
    a2_in = test_a[2];
    a3_in = test_a[3];

    for (j = 0; j < 16; j = j + 1) begin
        test_b = j;
        b0_in = test_b[0];
        b1_in = test_b[1];
        b2_in = test_b[2];
        b3_in = test_b[3];

        #10;

        expected_eq = (test_a == test_b);
        expected_gr = (test_a > test_b);
        expected_low = (test_a < test_b);

        if (result_equals_out != expected_eq ||
            result_greater_out != expected_gr ||
            result_smaller_out != expected_low) begin
            $display("Test failed for a=%b, b=%b. Expected: g=%b,
e=%b, s=%b. Got: g=%b, e=%b, s=%b",
                    test_a, test_b,
                    expected_gr, expected_eq, expected_low,
                    result_greater_out, result_equals_out,
result_smaller_out);
        end else begin
            $display("Test passed for a=%b, b=%b. g=%b, e=%b, s=%b",
                    test_a, test_b,
                    result_greater_out, result_equals_out,
result_smaller_out);
        end
    end
end

$display("Comparator test completed.");
#10 $stop;
end

```

endmodule

2.3. Временная диаграмма процесса тестирования БОЭ

3. Выводы по работе

В ходе выполнения первой лабораторной работы были изучены базовые принципы проектирования цифровых схем с использованием КМОП-транзисторов и LTspice-моделирования. Были получены навыки построения иерархических схем в LTspice и моделирования их работы. Это позволило понять особенности задержки сигнала и максимальной частоты работы логических элементов.

В процессе выполнения работы была создана схема компаратора в среде LTspice, построена её тестовая модель и определены временные характеристики, такие как задержка распространения сигнала и максимальная рабочая частота. В среде Vivado Design Suite на языке Verilog HDL была разработана и протестирована функциональная модель компаратора. Полученные результаты подтвердили работоспособность разработанной схемы для выполнения функции сравнения четырехразрядных чисел.