УНИВЕРСИТЕТ ИТМО

Факультет программной инженерии и компьютерной техники

Направление подготовки 09.03.04 Программная инженерия

Дисциплина «Функциональная схемотехника»

Лабораторная работа №1

Вариант 8

Выполнил:

Голиков Д.И.

P33102

Преподаватель:

Васильев С.Е.

Санкт-Петербург

2024 г.

Цели 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 вентильном уровне с использованием языка описания аппаратуры Verilog HDL.

Выполнение:

1.1. Схема вентиля

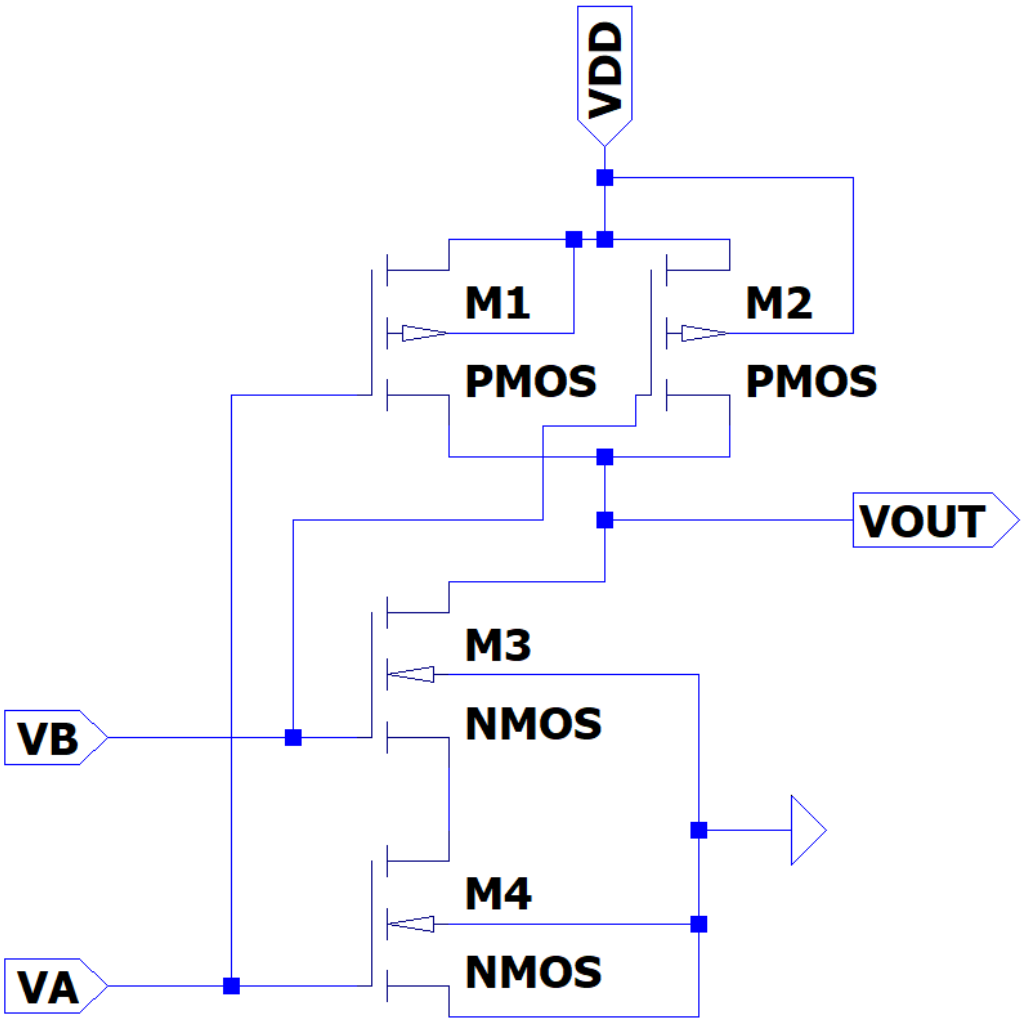


Рисунок 1. Схема вентиля.

1.2. Символ вентиля и схема тестирования

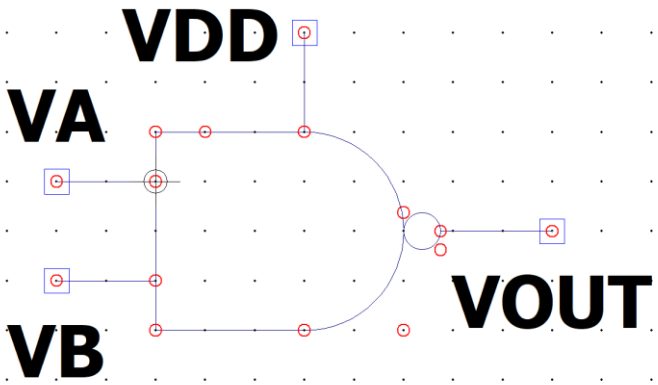


Рисунок 2. Символ вентиля NAND.

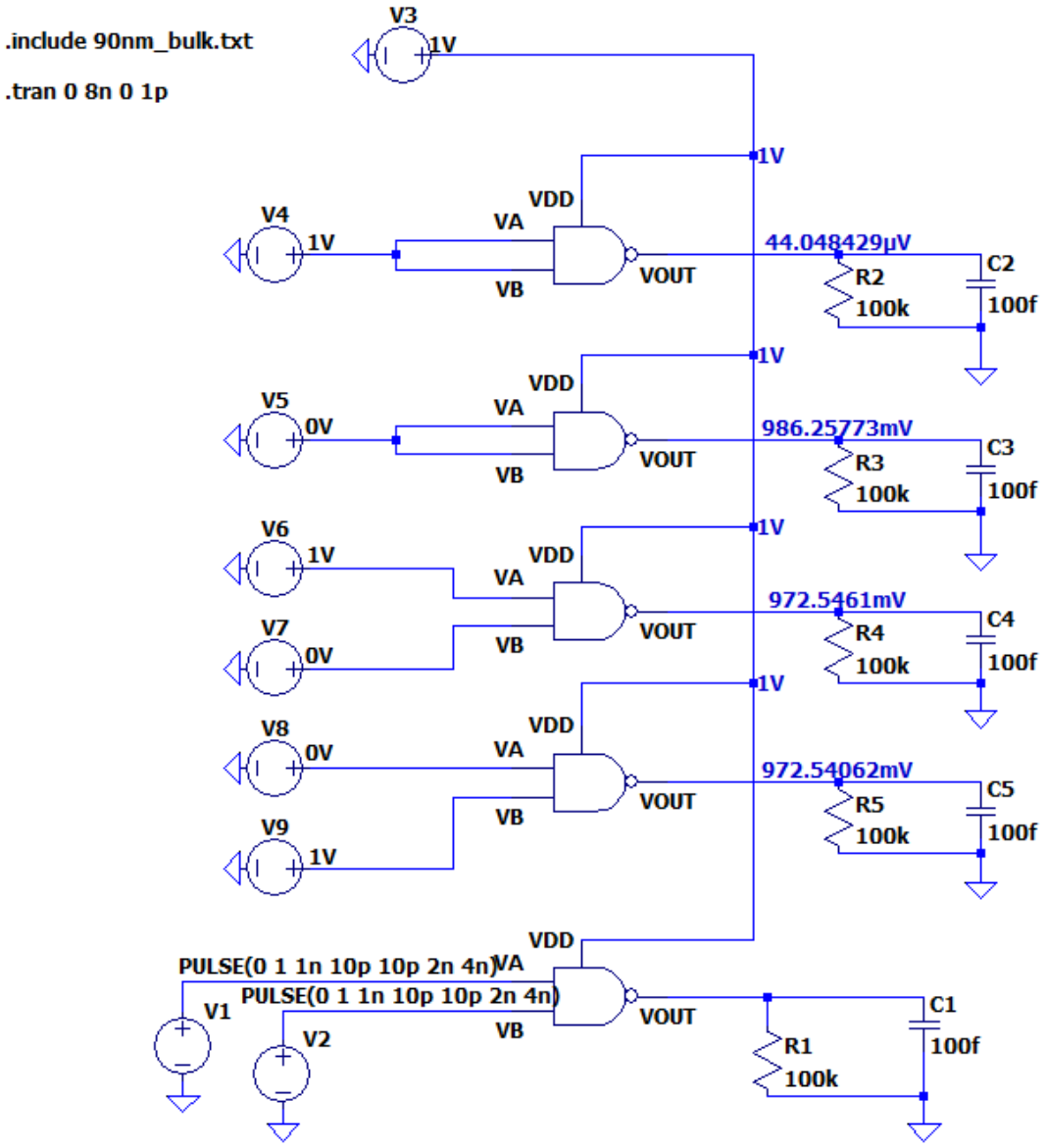


Рисунок 3. Схема тестирования.

В данной схеме первые четыре теста проверяют корректность таблица истинности NAND, последний тест необходим для расчёта задержек и максимальной частоты.

1.3. Временная диаграмма процесса тестирования вентиля

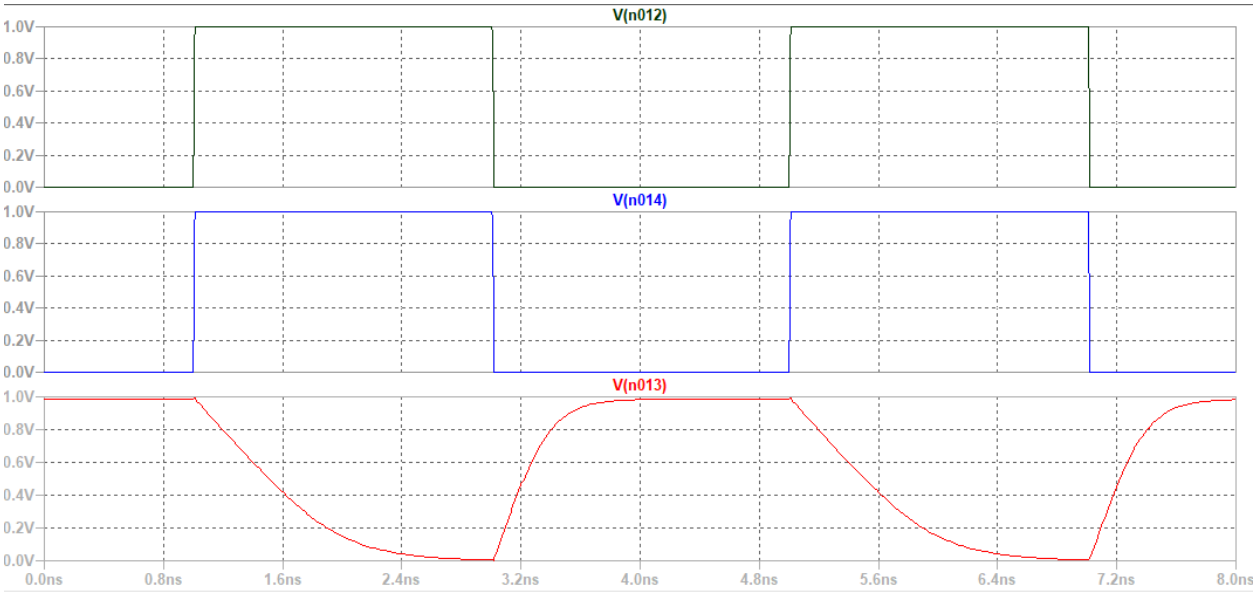


Рисунок 4. Временная диаграмма тестирования вентиля NAND.

1.4. Результат измерения задержки распространения сигнала через вентиль

Для расчета задержки по фронту и спаду сигнала из графика необходимо измерить временные интервалы между моментом, когда сигнал достигает 10% и 90% своего максимального значения.

Рассчитаем задержку по фронту и спаду:

$$d = t_{90\%} - t_{10\%}$$

$$d_r = 3.5276 - 3.011 = 0.5166\text{нс}$$

$$d_f = 5.0968 - 5.0089 = 0.0879\text{нс}$$

Найдём среднюю задержку:

$$d = \frac{0.5166 + 0.0879}{2} = 0.30225\text{нс}$$

1.5. Максимальная частота работы вентиля

Из найденной в предыдущем шаге задержки найдем частоту вентиля:

$$f = \frac{1}{0.30225 * 10^{-9}} = 3.3085\ \text{ГГц}$$

1.6. Схема разработанного БОЭ

Разрабатываемый БОЭ – Шифратор кода Грея для трехразрядного двоичного числа.

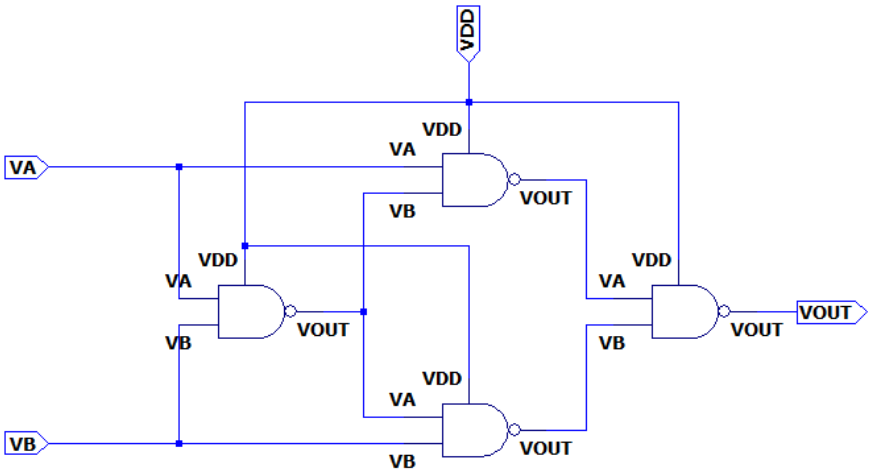


Рисунок 6. Схема XOR используя NAND.

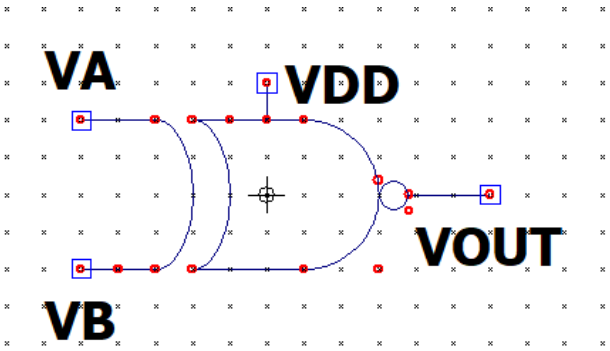


Рисунок 7. Символ вентиля XOR.

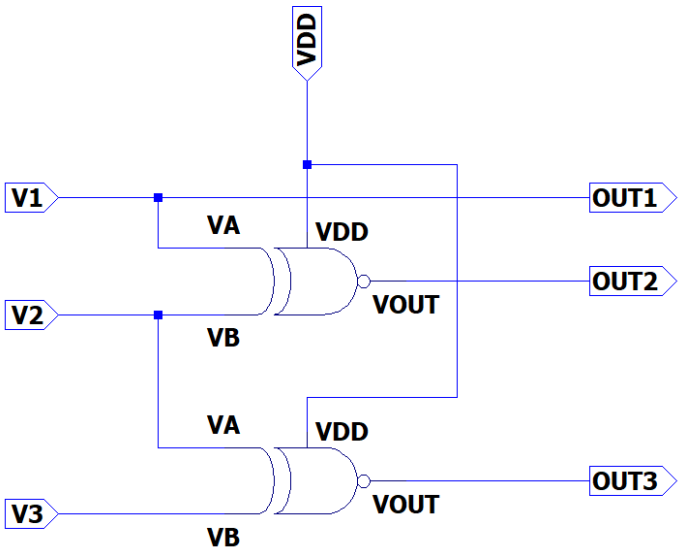


Рисунок 8. Схема БОЭ.

1.7. Символ разработанного БОЭ и схема тестирования

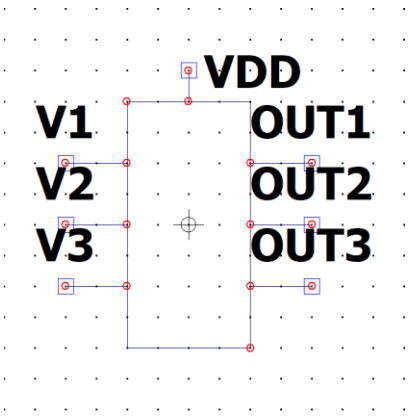
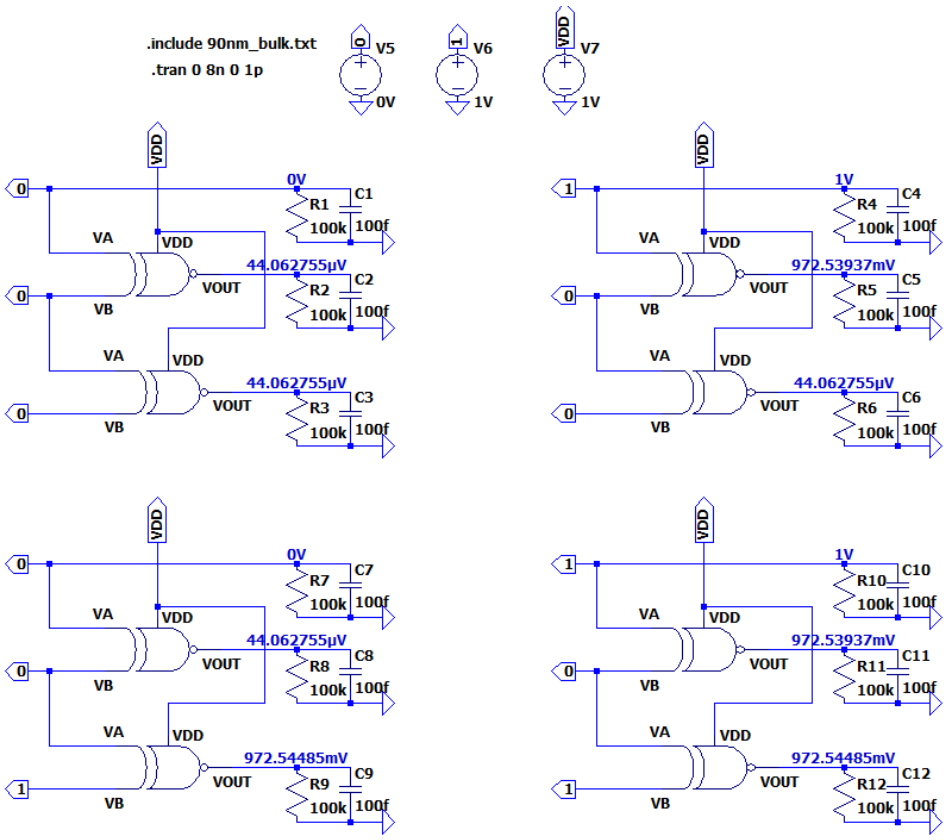


Рисунок 9. Символ БОЭ.



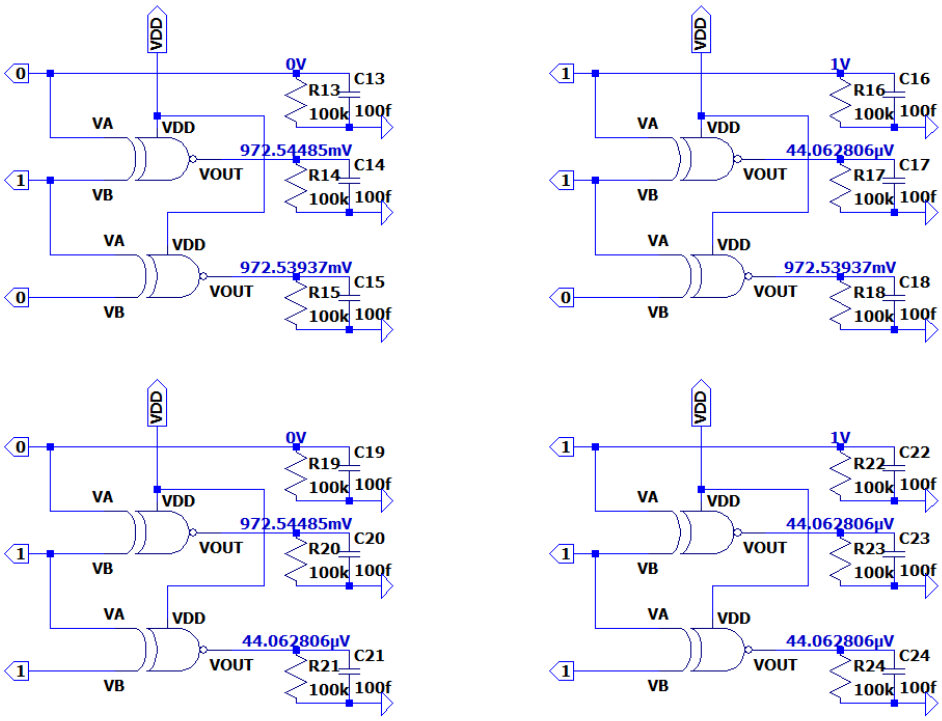


Рисунок 8. Схема тестирования БОЭ

В данной схеме восемь тестов проверяют корректность работы БОЭ согласно таблице истинности шифратора.

1.8. Временная диаграмма процесса тестирования БОЭ

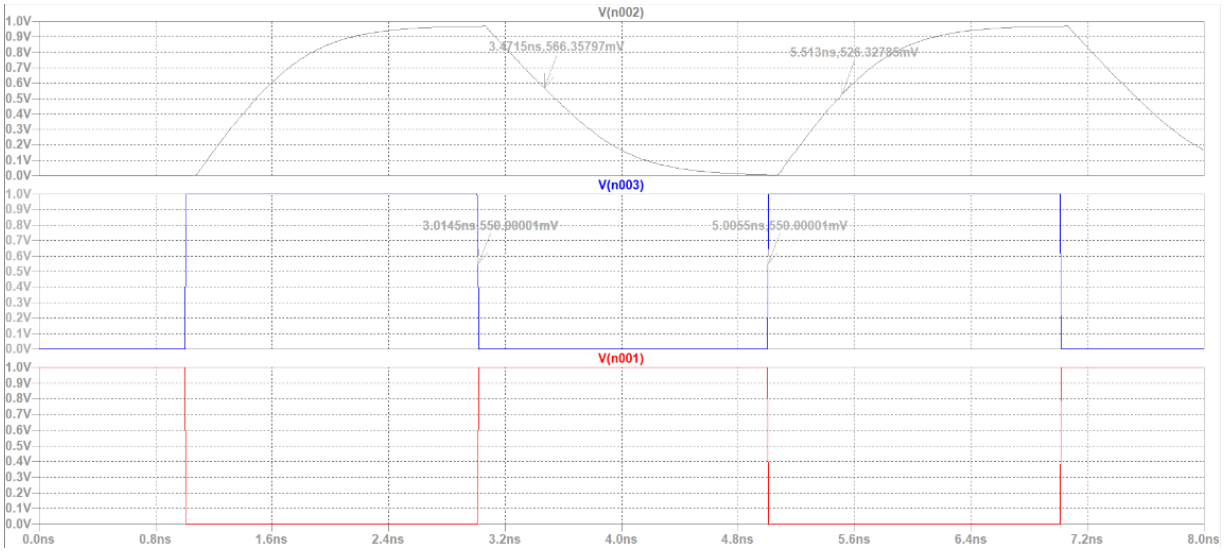


Рисунок 9. Временная диаграмма тестирования БОЭ.

1.9. Результат измерения задержки распространения сигнала через БОЭ

Рассчитаем задержки по фронту и спаду:

dr = 2.70 - 2.01 = 0.69

df = 3.74 - 3.01 = 0.73

d = 0.71

$$d_r = 3.0728 - 3.019 = 0.0538\text{нс}$$

$$d_f = 4.76732 - 4.012 = 0.7553\text{нс}$$

Найдём среднюю задержку:

$$d = \frac{0.0538 + 0.7553}{2} = 0.40456\text{нс}$$

1.10. Максимальная частота работы БОЭ

Из найденной на предыдущем шаге задержки найдем частоту вентиля:

$$f = \frac{1}{0.40456 * 10^{-9}} = 2.4718\text{ГГц}$$

2.1. Код разработанного модуля БОЭ

```
1   `timescale 1ns / 1ps
2
3   module nandxor(
4
5           input v1,
6           input v2,
7           output out
8           );
9
10          wire xor_v1_v2, xor_v1_out1, xor_v2_out1;
11
12          nand(xor_v1_v2, v1,v2);
13          nand(xor_v1_out1, v1, xor_v1_v2);
14          nand(xor_v2_out1, v2, xor_v1_v2);
15          nand(out, xor_v1_out1, xor_v2_out1);
16
17  endmodule
18
```

Рисунок 10. Реализация модуля XOR через NAND.

```
1   `timescale 1ns / 1ps
2
3   module my_gray(
4       input v1_in,
5       input v2_in,
6       input v3_in,
7       output v1_out,
8       output v2_out,
9       output v3_out
10       );
11      assign v1_out = v1_in;
12      nandxor nandxor_1(
13          .v1(v1_in),
14          .v2(v2_in),
15          .out(v2_out)
16      );
17
18      nandxor nandxor_2(
19          .v1(v2_in),
20          .v2(v3_in),
21          .out(v3_out)
22      );
23  endmodule
24
```

Рисунок 11. Реализация модуля БОЭ.

```
1   `timescale 1ns / 1ps
2
3   module master_gray(
4       input v1_in,
5       input v2_in,
6       input v3_in,
7       output v1_out,
8       output v2_out,
9       output v3_out
10       );
11      assign v1_out = v1_in;
12      xor(v2_out, v1_in, v2_in);
13      xor(v3_out, v2_in, v3_in);
14  endmodule
15
```

Рисунок 12. Реализация модуля БОЭ для тестирования кода грея.

2.2. Код разработанного тестового окружения БОЭ

```
module shifrator;

    reg v1_in, v2_in, v3_in;
    wire v1_out, v2_out, v3_out, v1_out_master, v2_out_master, v3_out_master;
    my_gray my(
        .v1_in(v1_in),
        .v2_in(v2_in),
        .v3_in(v3_in),
        .v1_out(v1_out),
        .v2_out(v2_out),
        .v3_out(v3_out)
    );

    master_gray master(
        .v1_in(v1_in),
        .v2_in(v2_in),
        .v3_in(v3_in),
        .v1_out(v1_out_master),
        .v2_out(v2_out_master),
        .v3_out(v3_out_master)
    );

    integer i,j,k;

    initial begin

        for(i=0; i <= 1; i = i + 1) begin
            v1_in = i;
            for(j=0; j <= 1; j = j + 1) begin
                v2_in = j;
                for(k=0; k <= 1; k = k+1) begin
                    v3_in = k;
                    #2
                    if((v1_out == v1_out_master) | (v2_out == v2_out_master) | (v3_out == v3_out_master)) begin
                     $display("Just take the nubmers v1_in=%b, v2_in=%b, v3_in=%b v1_out=%b, v2_out=%b, v3_out=%b,  v1_out_master=%b, v2_out_master=%b, v3_out_master=%b",
                      v1_in, v2_in, v3_in, v1_out, v2_out, v3_out, v1_out_master, v2_out_master, v3_out_master);
                    end
                    else begin
                      $display("error");
                    end
                end
            end

```

Рисунок 13. Программа тестирования БОЭ.

2.3. Временная диаграмма процесса тестирования БОЭ

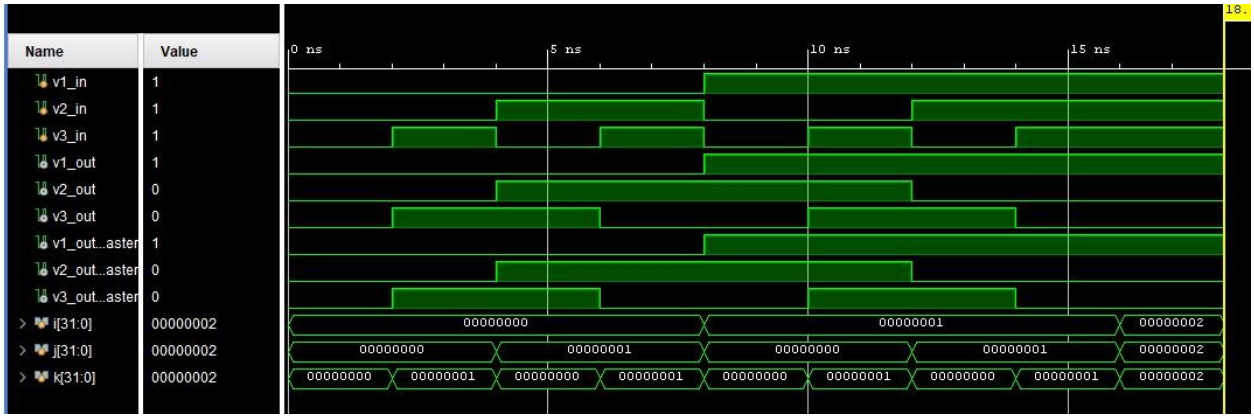


Рисунок 14. Временная диаграмма тестирования БОЭ.

```
Just take the nubmers v1_in=0, v2_in=0, v3_in=0 v1_out=0, v2_out=0, v3_out=0,  v1_out_master=0, v2_out_master=0, v3_out_master=0
Just take the nubmers v1_in=0, v2_in=0, v3_in=1 v1_out=0, v2_out=0, v3_out=1,  v1_out_master=0, v2_out_master=0, v3_out_master=1
Just take the nubmers v1_in=0, v2_in=1, v3_in=0 v1_out=0, v2_out=1, v3_out=1,  v1_out_master=0, v2_out_master=1, v3_out_master=1
Just take the nubmers v1_in=0, v2_in=1, v3_in=1 v1_out=0, v2_out=1, v3_out=0,  v1_out_master=0, v2_out_master=1, v3_out_master=0
Just take the nubmers v1_in=1, v2_in=0, v3_in=0 v1_out=1, v2_out=1, v3_out=0,  v1_out_master=1, v2_out_master=1, v3_out_master=0
Just take the nubmers v1_in=1, v2_in=0, v3_in=1 v1_out=1, v2_out=1, v3_out=1,  v1_out_master=1, v2_out_master=1, v3_out_master=1
Just take the nubmers v1_in=1, v2_in=1, v3_in=0 v1_out=1, v2_out=0, v3_out=1,  v1_out_master=1, v2_out_master=0, v3_out_master=1
Just take the nubmers v1_in=1, v2_in=1, v3_in=1 v1_out=1, v2_out=0, v3_out=0,  v1_out_master=1, v2_out_master=0, v3_out_master=0
$stop called at time : 18 ns : File "C:/Users/user/lab_1_v3/lab_1_v3.srcs/sim_1/new/shifrator.v" Line 47
```

Рисунок 15. Вывод консоли в результате работы программы.

Выводы по работе:

В ходе выполнения лабораторной работы были изучены базовые принципы создания цифровых интегральных схем с использованием технологии КМОП, созданы вентили NAND и XOR в среде моделирования LTspice, схема шифратора кода Грея для трехразрядного двоичного числа, а также схемы для их тестирования. Был изучен язык описания аппаратуры Verilog HDL, с помощью которого была схема БОЭ и протестирована её функциональность.

Почему средняя задержка средняя арифметическая?

Для расчёта средней задержки мы используем время для rise и fall, когда мощность достигает 50%. То есть мы будем считать, что после преодоления 50%-ой планки, мы принимаем переходное значение компонента. Следовательно, для того, чтобы рассчитать среднюю задержку, мы складываем rise и fall и делим на пополам.

Например:

Имея задержку фронта 0.5нс и задержку спада 0.3нс, мы должны получить среднее переходное значение немного больше, чем большее из задержек спадов. Если посчитать, используя большую, то это 0.25 по фронту.

Проверим среднюю задержку

$$d = \frac{0.5 + 0.3}{2} = 0.4$$

Полученное значение получилось больше чем половина от большего из двух задержек, следовательно, она подходит для дальнейших расчетов и не возникает противоречий.

Почему задержка на БОЭ с использованием множества компонентов не в разы больше, чем задержка на одном компоненте?

Полезным напряжением в данной лабораторной работе считается напряжение на конденсатор и резистор. Следовательно, у нас задержка становится прямо пропорциональной ёмкости конденсатора. Если принять значение конденсатора равным нулю, то задержка на БОЭ , содержащее 4 компонента будет в 4 раза больше чем на одном компоненте.

Также сами нмос и пмос резисторы частично являются конденсаторами, из-за этого тоже появляются задержки на всех схемах.