Университет ИТМО

Факультет Программной инженерии и компьютерной техники

Лабораторная работа №1

Дисциплина «Функциональная схемотехника»

Вариант 6

NAND Позиционный дешифратор «3 в 8»

Выполнили:

Лушникова А.С.

Чернова Е.А.

Группа P33302

Преподаватель:

Кустарев Павел Валерьевич

Табунщик Сергей Михайлович

Санкт-Петербург

2024

Текст задания:

Часть 1:

1. Постройте в LTspice на транзисторах схему вентиля, составляющего основу логического базиса согласно варианту задания.
2. Создайте символ для разработанного вентиля как иерархического элемента.
3. С использованием созданного иерархического элемента постройте схему тестирования вентиля.
4. Проведите моделирование работы схемы и определите задержку распространения сигнала через тестируемый вентиль.
5. Определите максимальную частоту изменения входных сигналов, при которой построенная схема сохраняет работоспособность.
6. Постройте БОЭ на базе созданного вентиля согласно варианту задания.
7. Создайте символ для построенного БОЭ.
8. Проведите моделирование работы схемы и определите задержку распространения сигнала через БОЭ.
9. Определите максимальную частоту изменения входных сигналов, при которой построенная схема сохраняет работоспособность.
10. Составьте отчет по результатам выполнения заданий первой части лабораторной работы.

8

Часть 2:

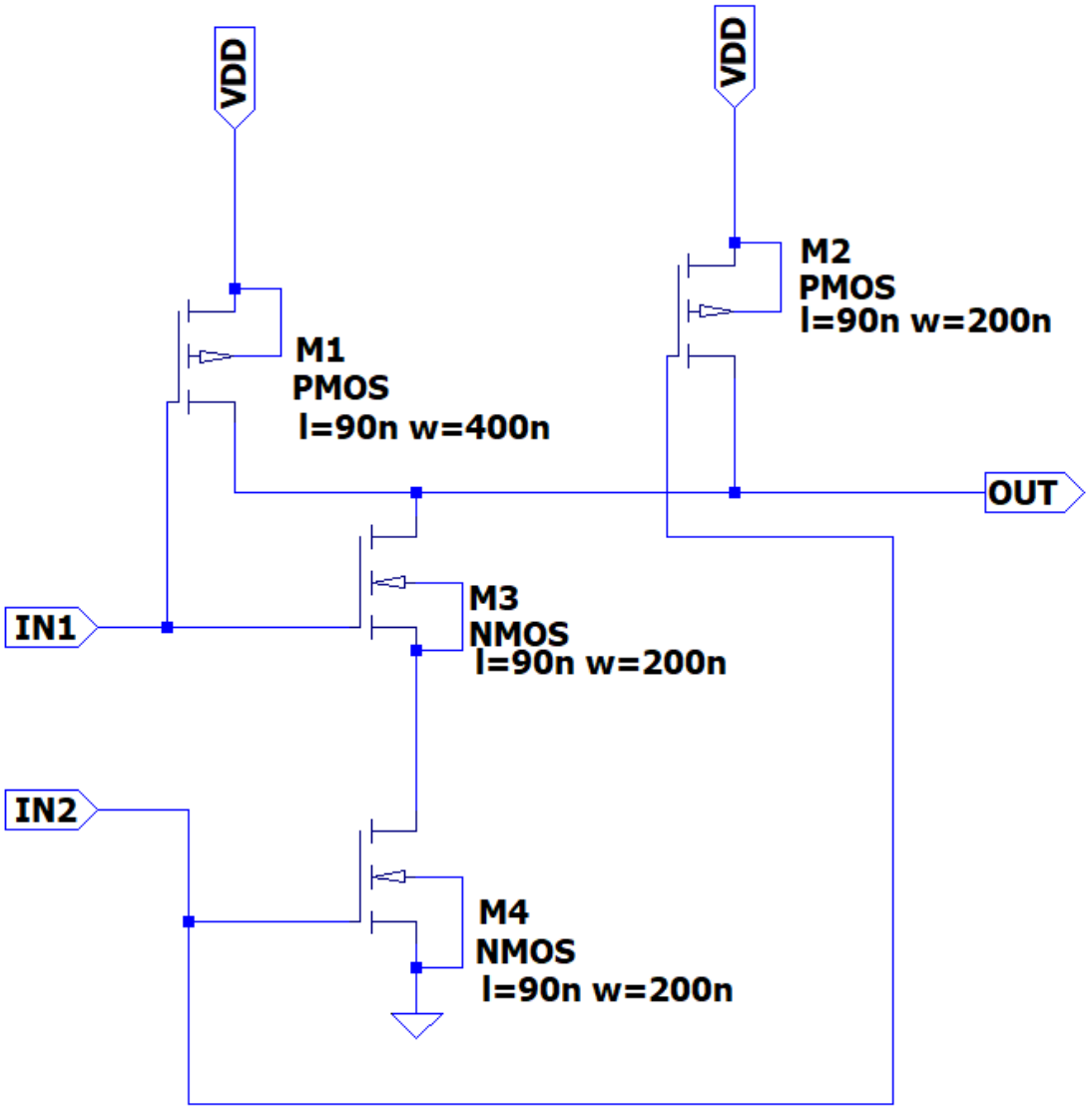
1. Опишите на Verilog HDL на вентильном уровне модуль, реализующий функцию БОЭ в указанном логическом базисе согласно варианту задания.
2. Разработайте тестовое окружение для созданного модуля.
3. Проведите моделирование работы схемы.
4. Составьте отчет по результатам выполнения заданий второй части лабораторной работы.

Часть 1

Логический базис состоит из элемента NAND, значение которого является логическим нулем только если оба входных аргумента равны 1, иначе значение - логическая единица.

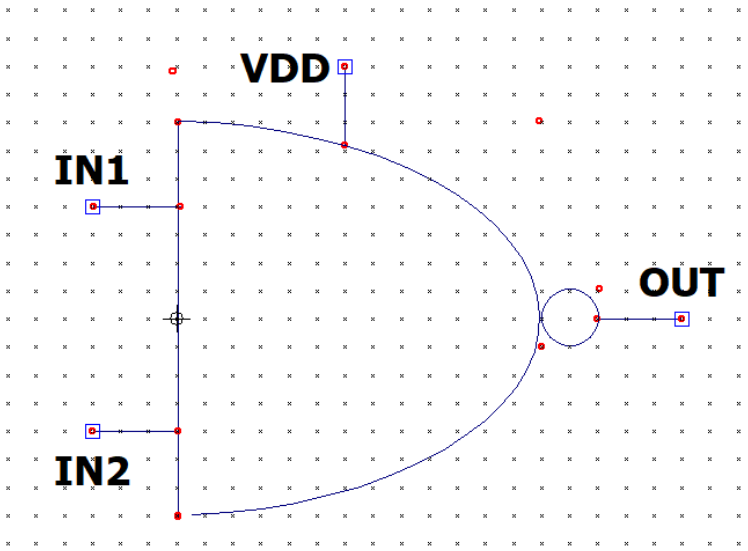
Truth Table		
A	B	Q
0	0	1
0	1	1
1	0	1
1	1	0

В приложении LTspice была разработана схема вентиля на транзисторах:



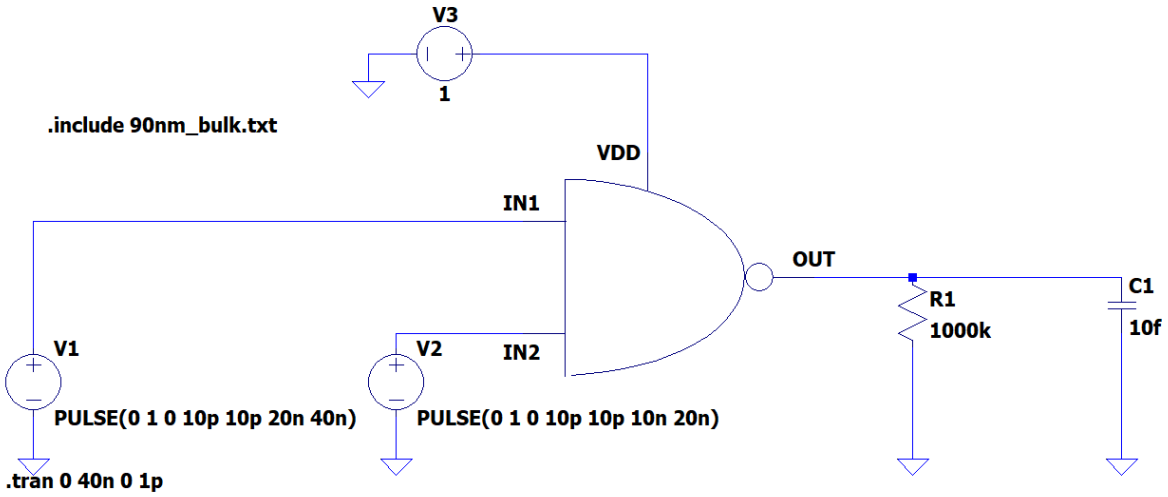
Создание символа вентиля

Была создан символ для разработанного вентиля.



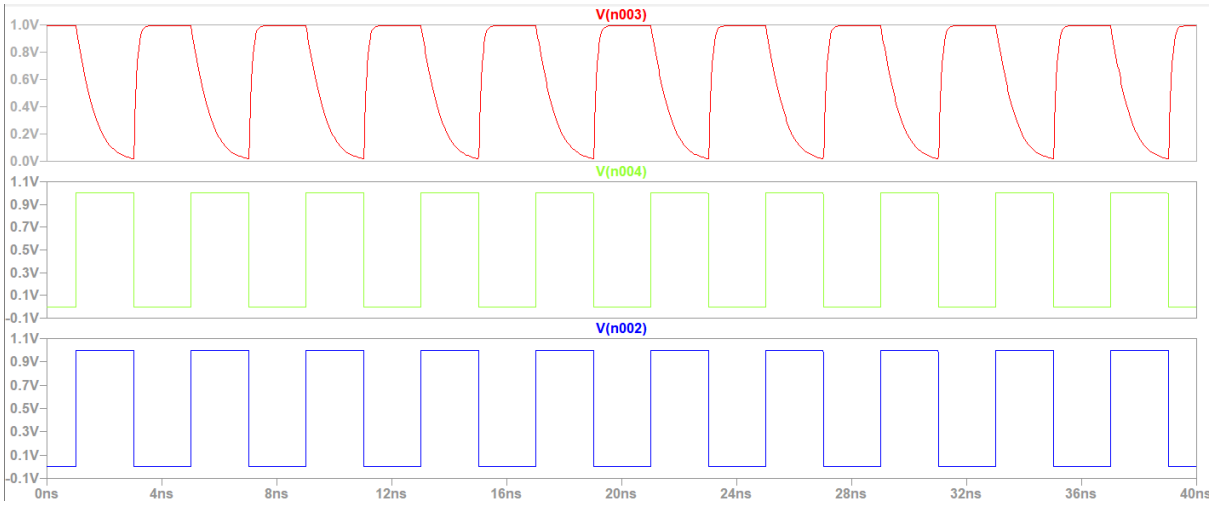
Построение схемы тестирования

Была построена схема тестирования разработанного вентиля с использованием созданного элемента.



Моделирование работы схемы и определение задержки вентиля

Моделирование при одинаковой задержке генераторов импульса. В данном случае генерируются пары аргументов (1, 1) и (0, 0), с ожидаемым значением вентиля 0 и 1 соответственно. (n003 - выходной сигнал).



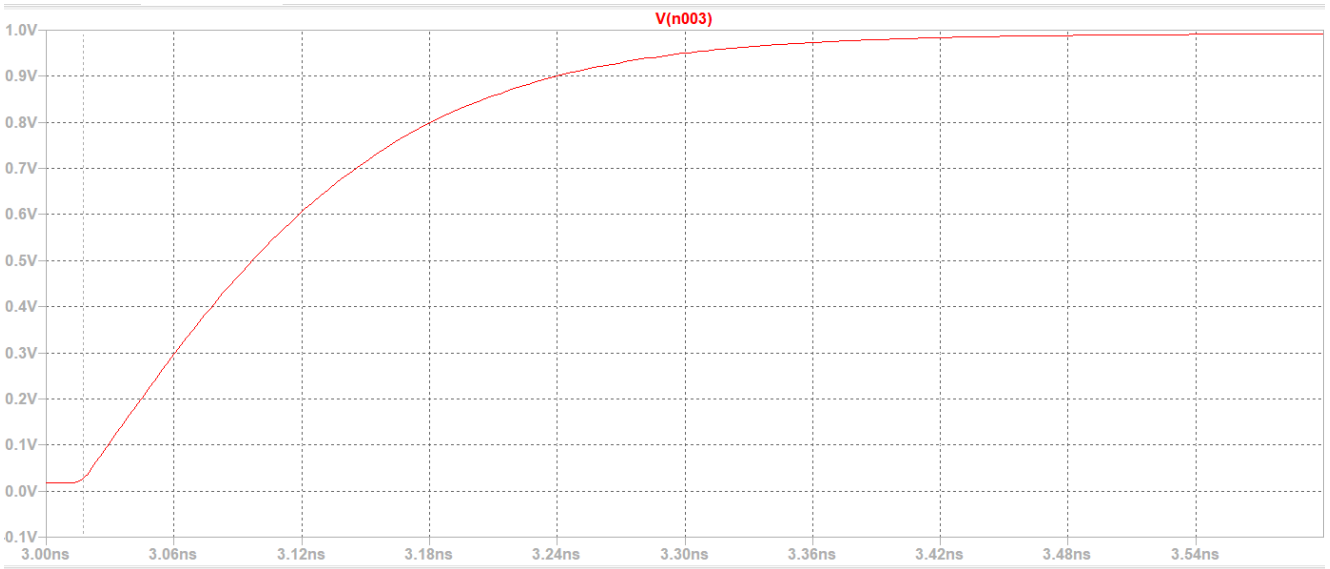
Также необходимо произвести измерения задержки распространения сигнала через вентиль. Для этого необходимо приблизить график и опустить перпендикуляры от крайних точек перехода на ось абсцисс, после чего найти разницу ординат проекций точек. Поскольку, данный график построен не при помощи приборов измерения напряжения, а были взяты в качестве результатов компьютерной симуляции, погрешность измерения величины по данному графику многократно меньше, погрешности, вызванной сопоставления проекций точек значениям на оси. Следовательно, графическим способом мы можем нанести дополнительные деления на ось, находящиеся по середине делений оси (отмечены красным на рисунке) и принять погрешность данных измерений четверти цены деления.

Время задержки сигнала τ:

$t1 = 3.01$ нс

$t2 = 3.48$ нс

$\tau = t2 - t1 = 0.47$ нс



Максимальная частота изменения сигнала вентиля

Аналитический подсчет максимальной частоты работа вентиля зависит от времени задержки сигнала: ***vMax* = 1/τ ≈ 2.127 ГГц.**
Построение БОЭ на базе созданного вентиля
Дешифратор - комбинационное устройство, преобразующее n-разрядный двоичный код в логический сигнал, появляющийся на том выходе, десятичный номер которого соответствует двоичному коду. Число входов и выходов в так называемом полном дешифраторе связано соотношением m= 2n, где n- число входов, а m— число выходов. Дешифратор 3 в 8 имеет 3 входа (IN1, IN2, IN3) и 8 выходов.

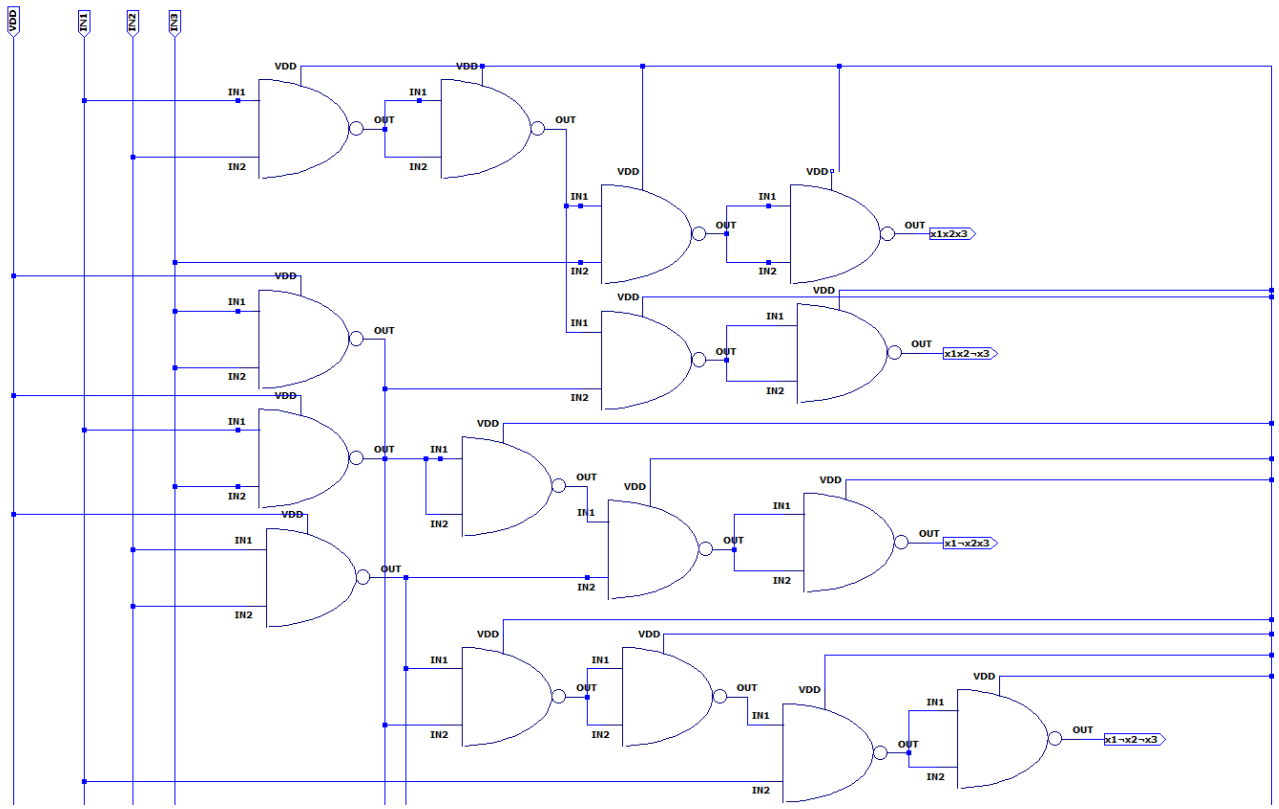
По данному описанию БОЭ построим ожидаемую таблицу истинности.

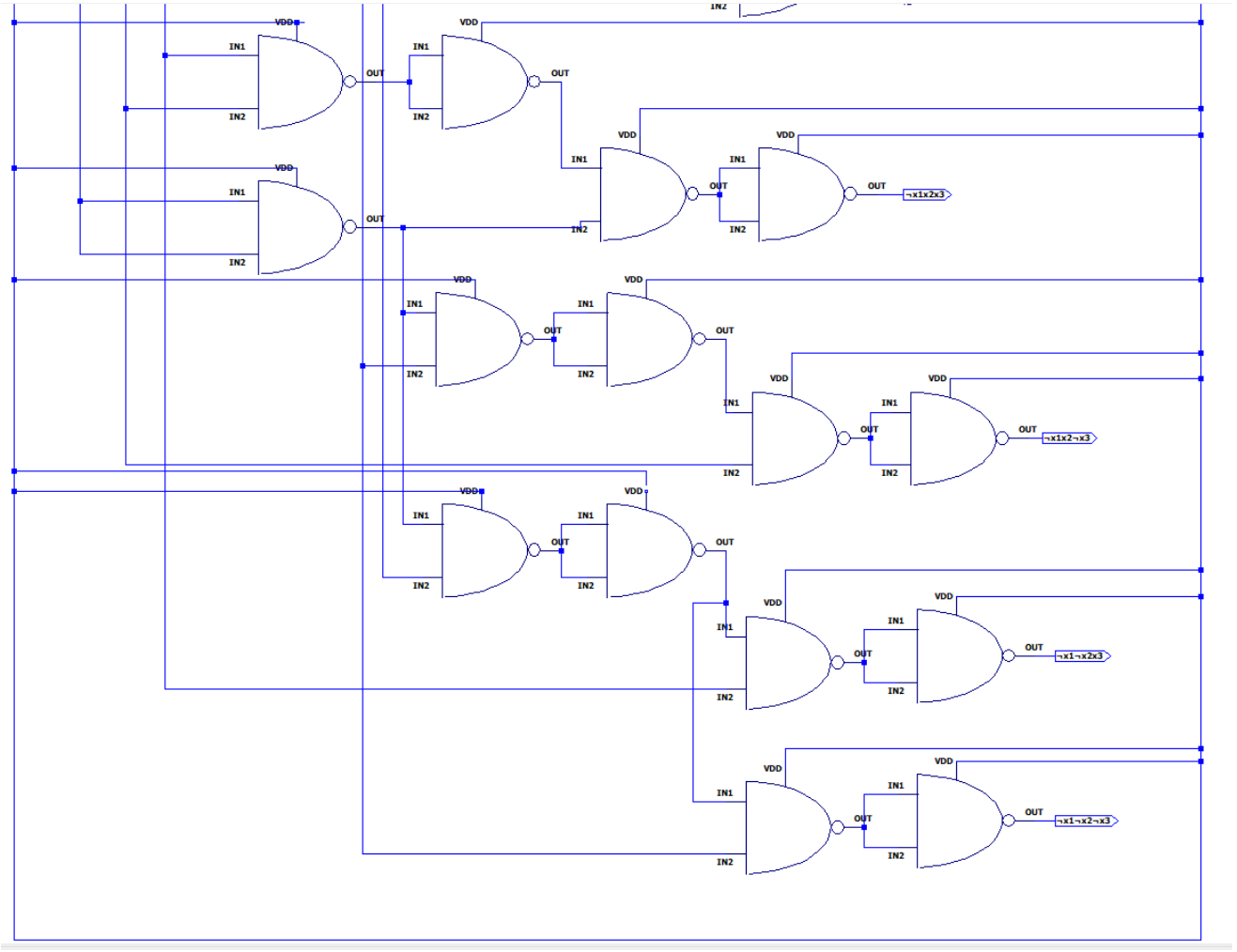
A2	A1	A0	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

Построим для каждого выхода элемента логическую функцию:
Y0 = ¬A2 * ¬A1 * ¬A0
Y1 = ¬A2 * ¬A1 * A0
Y2 = ¬A2 * A1 * ¬A0
Y3 = ¬A2 * A1 * A0
Y4 = A2 * ¬A1 * ¬A0
Y5 = A2 * ¬A1 * A0
Y6 = A2 * A1 * ¬A0
Y7 = A2 * A1 * A0

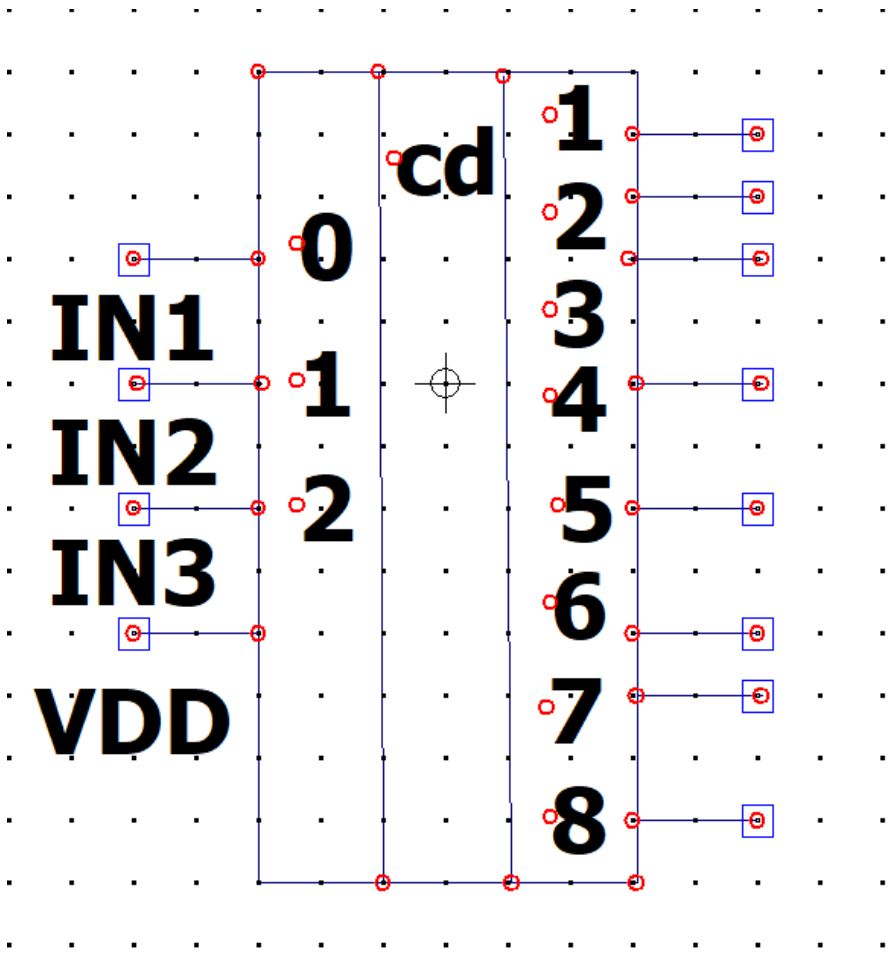
¬ обозначают инвертированные входные сигналы.
Выразим логические функции для каждого выхода относительно базиса NAND.
Y0 = ((A2 NAND A2) NAND (A1 NAND A1)) NAND (A0 NAND A0)
Y1 = ((A2 NAND A2) NAND (A1 NAND A1)) NAND (A0 NAND A0 NAND A0)
Y2 = ((A2 NAND A2) NAND (A1 NAND A1 NAND A1)) NAND (A0 NAND A0)
Y3 = ((A2 NAND A2) NAND (A1 NAND A1 NAND A1)) NAND (A0 NAND A0 NAND A0)
Y4 = ((A2 NAND A2 NAND A2) NAND (A1 NAND A1)) NAND (A0 NAND A0)
Y5 = ((A2 NAND A2 NAND A2) NAND (A1 NAND A1)) NAND (A0 NAND A0 NAND A0)
Y6 = ((A2 NAND A2 NAND A2) NAND (A1 NAND A1 NAND A1)) NAND (A0 NAND A0)
Y7 = ((A2 NAND A2 NAND A2) NAND (A1 NAND A1 NAND A1)) NAND (A0 NAND A0 NAND A0)

На основе полученных функций составим схему дешифратора 3 в 8:



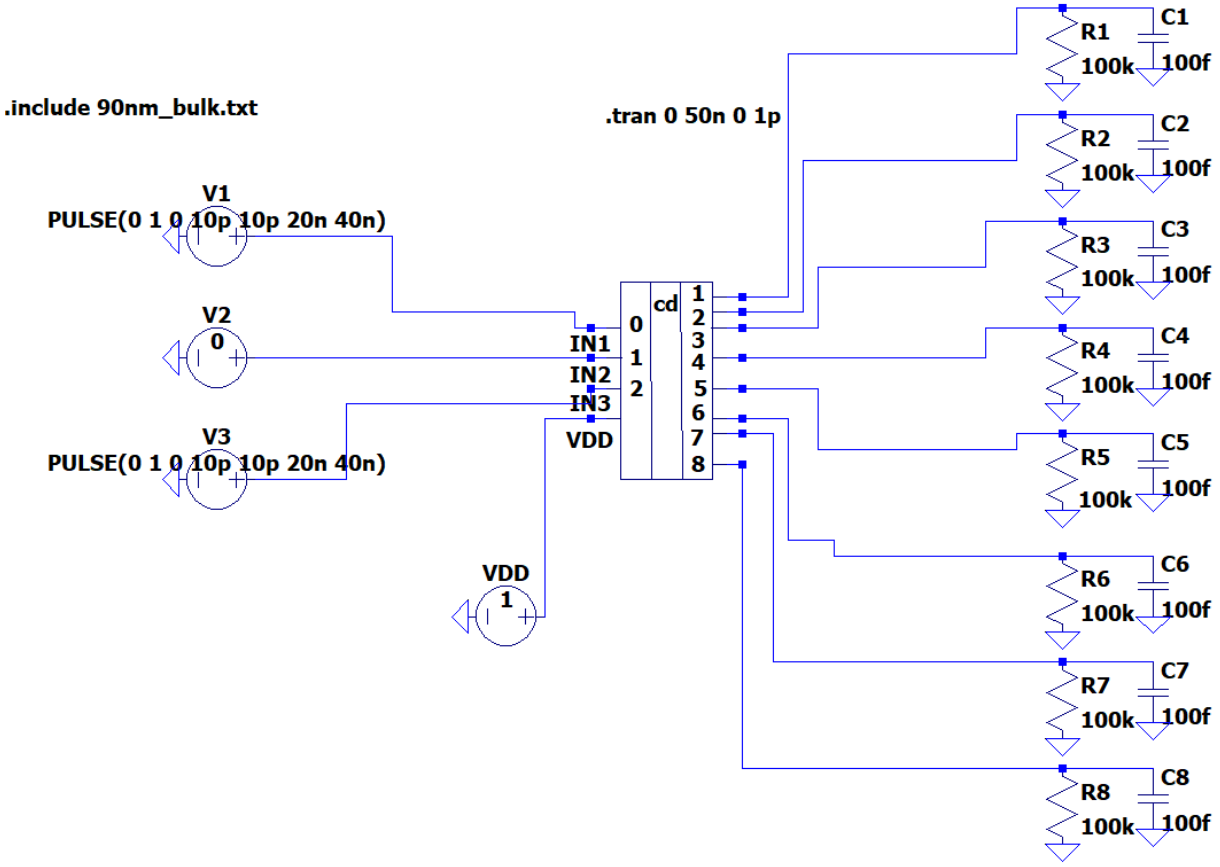

VDD
IN1
IN2
OUT
¬x1x2x3
¬x1x2¬x3
¬x1¬x2x3
¬x1¬x2¬x3

Создание символа для построенного БОЭ

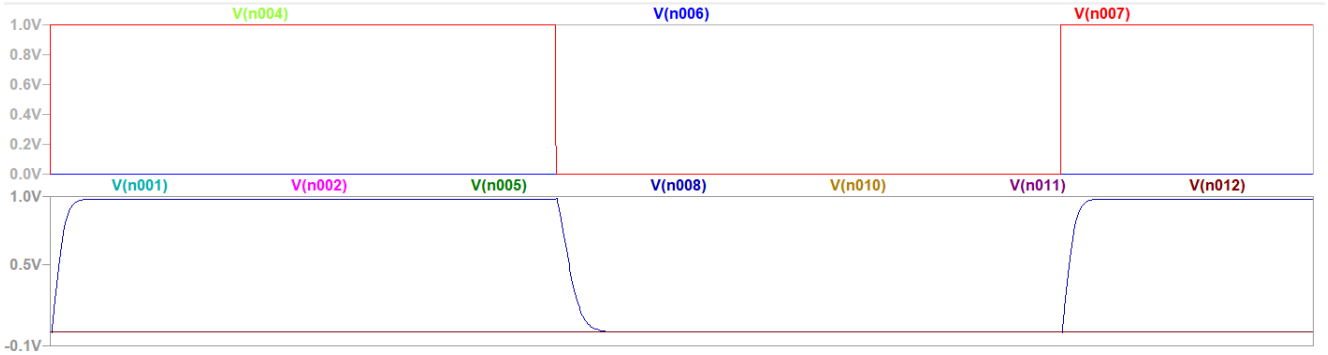


Моделирование работы схемы и определение задержки БОЭ

Для проведения моделирования работы схемы необходимо составить схему, состоящую из генераторов импульса и разработанного БОЭ.



После построения схемы необходимо запустить симуляцию и проверить, что значения на выходе схемы, при разных значения на входе схемы соответствуют таблице истинности.

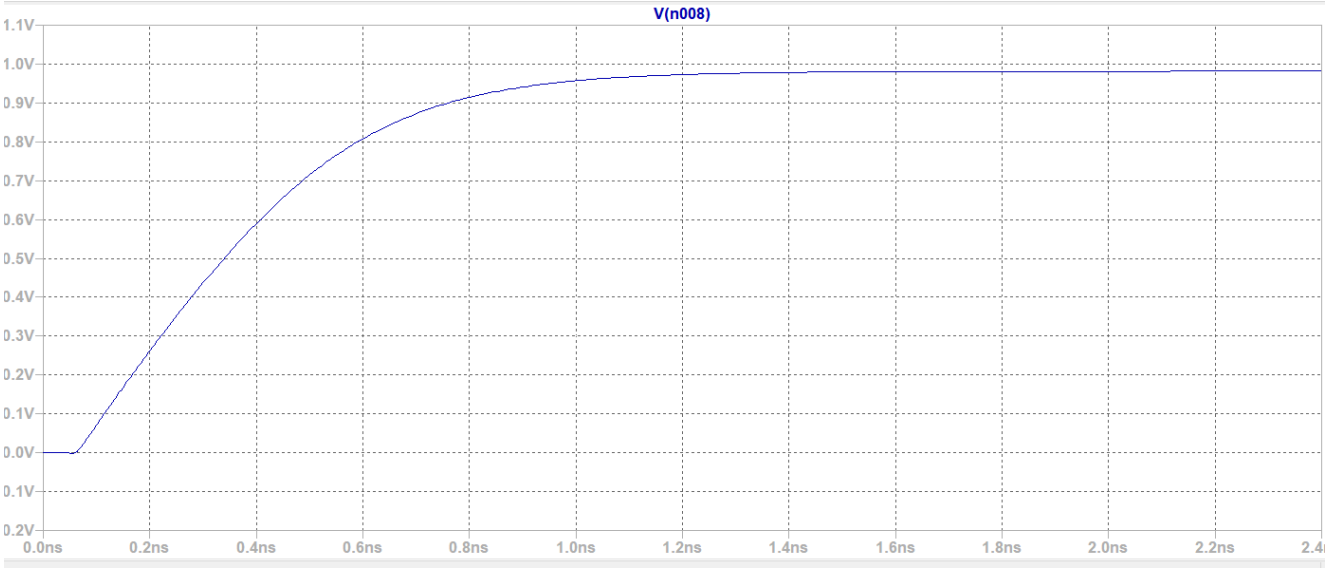


При заданных значениях на входе (0, 1, 0) только на выходе логической функции Y5 = A2 * ¬A1 * A0 должна быть логическая единица, что мы и видим на графике. Vn008 как раз соответствует этому выходу (номер 5 на схеме).

Также необходимо произвести измерения задержки распространения сигнала через БОЭ. Для этого необходимо приблизить график и опустить перпендикуляры от крайних точек перехода на ось абсцисс, после чего найти разницу ординат проекций точек.

Поскольку, данный график построен не при помощи приборов измерения напряжения, а были взяты в качестве результатов компьютерной симуляции, погрешность измерения величины по данному графику многократно меньше, погрешности, вызванной сопоставления проекций точек значениям на оси. Следовательно, графическим способом мы можем нанести дополнительные деления на ось, находящиеся посередине делений оси

(отмечены красным на рисунке) и принять погрешность данных измерений четверти цены деления.



Время задержки сигнала τ: $t1 = 0.08$ нс $t2 = 1.5$ нс $\tau = t2 - t1 = 1.4$ нс

Максимальная частота изменения сигнала БОЭ

Аналитический подсчет максимальной частоты работа вентиля зависит от времени задержки сигнала: $\boldsymbol{vMax} = \mathbf{1}/\boldsymbol{\tau} \approx 414$ МГц

Часть 2

В рамках данной лабораторной работы было разработано два модуля, элемент and и сама программа реализующая БОЭ.

And

Реализация **and** через **nand**

```
`timescale 1ns / 1ps

module and_impl(
    input x0,
    input x1,
    input x2,
    output res);

    wire not_x0, not_x1,not_x2, not_and_x0_x1, and_x0_x1, not_x0_x1_x2;
    nand(not_x0,x0,x0);
    nand(not_x1,x1,x1);
    nand(not_x2,x2,x2);

    nand(not_and_x0_x1,x0,x1);
    nand(and_x0_x1,not_and_x0_x1, not_and_x0_x1);
    nand(not_x0_x1_x2, and_x0_x1,x2);
    nand(res, not_x0_x1_x2, not_x0_x1_x2);

endmodule
```

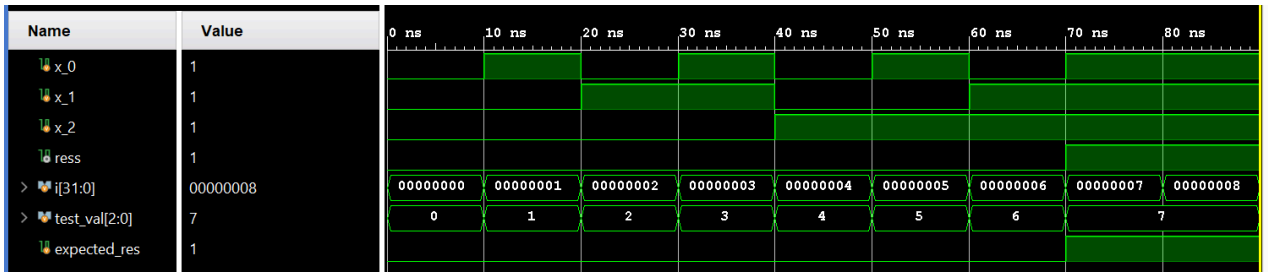
Тестирование

```
`timescale 1ns / 1ps
module and_tb;
reg x_0,x_1,x_2;
wire ress;
and_impl andd(
    .x0(x_0),
    .x1(x_1),
    .x2(x_2),
    .res(ress));

integer i;
reg [2:0] test_val;
reg expected_res;
initial begin
    for(i=0;i<8;i=i+1) begin
        test_val=i;
        x_0=test_val[0];
        x_1=test_val[1];
        x_2=test_val[2];
        expected_res=&test_val;

        #10
          if (ress==expected_res) begin
           $display ("and:Correct!!! x_0=%b x_1=%b x_2=%b res=%b", x_0,x_1,x_2,ress);
          end else begin
           $display ("and:Wrong!!! x_0=%b x_1=%b x_2=%b res=%b, expected=&b", x_0,x_1,x_2,ress, expected_res);
          end
    end
        #10 $stop;
end
```

Результаты работы:

```
# run 1000ns
Correct!!! x_0=0 x_1=0 x_2=0 res=0
Correct!!! x_0=1 x_1=0 x_2=0 res=0
Correct!!! x_0=0 x_1=1 x_2=0 res=0
Correct!!! x_0=1 x_1=1 x_2=0 res=0
Correct!!! x_0=0 x_1=0 x_2=1 res=0
Correct!!! x_0=1 x_1=0 x_2=1 res=0
Correct!!! x_0=0 x_1=1 x_2=1 res=0
Correct!!! x_0=1 x_1=1 x_2=1 res=1
$stop called at time : 90 ns : File "C:/Users/yfcni/lab1/lab1.srcs/sim_1/new/and_impl_test.v" Line 30
```

График:



Основной модуль

Реализация БОЭ

```
`timescale 1ns / 1ps

module prog(
    input x_0,
    input x_1,
    input x_2,
    output[7:0] res
    );
    wire not_x0, not_x1, not_x2;
    nand(not_x0, x_0,x_0);
    nand(not_x1, x_1,x_1);
    nand(not_x2, x_2,x_2);

    and_impl q0(.x0(x_0), .x1(x_1), .x2(x_2), .res(res[7]));
    and_impl q1(.x0(x_0), .x1(x_1), .x2(not_x2), .res(res[6]));
    and_impl q2(.x0(x_0), .x1(not_x1), .x2(x_2), .res(res[5]));
    and_impl q3(.x0(x_0), .x1(not_x1), .x2(not_x2), .res(res[4]));
    and_impl q4(.x0(not_x0), .x1(x_1), .x2(x_2), .res(res[3]));
    and_impl q5(.x0(not_x0), .x1(x_1), .x2(not_x2), .res(res[2]));
    and_impl q6(.x0(not_x0), .x1(not_x1), .x2(x_2), .res(res[1]));
    and_impl q7(.x0(not_x0), .x1(not_x1), .x2(not_x2), .res(res[0]));
endmodule
```

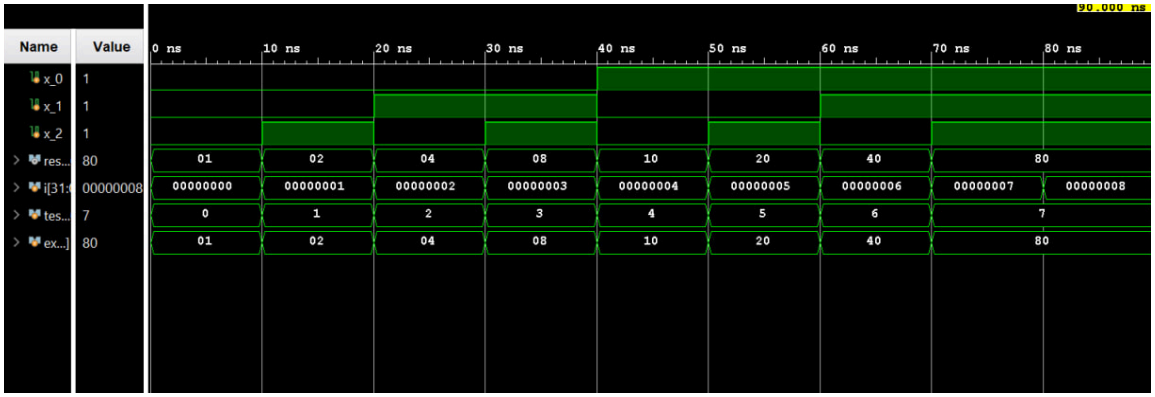
Тестирование:

```
`timescale 1ns / 1ps

module prog_tb;
  reg x_0, x_1, x_2;
  wire [7:0] ress;
  prog prog_1 (
    .x_0(x_0),
    .x_1(x_1),
    .x_2(x_2),
    .res(ress)
  );
  integer i;
  reg [2:0] test_val;
  reg [7:0] expected_res;

  initial begin
    for (i = 0; i < 8; i = i + 1) begin
      test_val = i;
      {x_0, x_1, x_2} = test_val;
      expected_res = $pow(2,i);
      #10

      if (expected_res==ress) begin
        $display("Correct!!! x_0=%b x_1=%b x_2=%b res=%b", x_0, x_1, x_2, $unsigned(ress));
      end else begin
        $display("Wrong!!! x_0=%b x_1=%b x_2=%b res=%b, expected=%b", x_0, x_1, x_2, ress, expected_res);
      end

    end

    #10 $stop;

  end
```

Результаты работы:

```
Correct!!! x_0=0 x_1=0 x_2=0 res=00000001
Correct!!! x_0=0 x_1=0 x_2=1 res=00000010
Correct!!! x_0=0 x_1=1 x_2=0 res=00000100
Correct!!! x_0=0 x_1=1 x_2=1 res=00001000
Correct!!! x_0=1 x_1=0 x_2=0 res=00010000
Correct!!! x_0=1 x_1=0 x_2=1 res=00100000
Correct!!! x_0=1 x_1=1 x_2=0 res=01000000
Correct!!! x_0=1 x_1=1 x_2=1 res=10000000
```

График:



Вывод

В процессе выполнения данной лабораторной работы я познакомилась со средой LTspice и с языком описания аппаратуры Verilog HDL.