

«Национальный исследовательский университет ИТМО»

# Лабораторная работа №1 по дисциплину "Функциональная схемотехника"

Введение в проектирование цифровых интегральных схем

Выполнил: Чан Куанг Лок  
Группа: Р33011

Санкт-Петербург  
2021 г

# 1 Цель работы

1. Получить базовые знания о принципах построения цифровых интегральных схем с использованием технологии КМОП.
2. Получить навыки описания схем базовых операционных элементов (БОЭ) комбинационного типа на вентиляльном уровне с использованием языка описания аппаратуры Verilog HDL.

## 2 Описание задания

Но варианта: 3

Логический базис	БОЭ
NOR	Схема мажоритарного контроля с 5-ю входами

## 3 Выполнение

### 3.1 Ссылка на github

<https://github.com/quangloc99/electric-engineering-lab1>

### 3.2 Код разработанного модуля БОЭ

```
1  `timescale 1ns / 1ps
2  ///////////////////////////////////////////////////////////////////
3  // Company:
4  // Engineer:
5  //
6  // Create Date: 03/19/2021 05:20:52 PM
7  // Design Name:
8  // Module Name: majority_5
9  // Project Name:
10 // Target Devices:
11 // Tool Versions:
12 // Description:
13 //
14 // Dependencies:
15 //
16 // Revision:
17 // Revision 0.01 - File Created
18 // Additional Comments:
19 //
20 ///////////////////////////////////////////////////////////////////
21
22
23 module majority_5(
24     output y,
25     input x0, input x1, input x2, input x3, input x4
26 );
27
28     wire not_x0, not_x1, not_x2, not_x3, not_x4;
29     nor(not_x0, x0, x0);
30     nor(not_x1, x1, x1);
31     nor(not_x2, x2, x2);
32     nor(not_x3, x3, x3);
33     nor(not_x4, x4, x4);
34     wire term012, term013, term014, term023, term024, term034, term123, term124, term134, term234;
35     nor(term012, not_x0, not_x1, not_x2);
36     nor(term013, not_x0, not_x1, not_x3);
37     nor(term014, not_x0, not_x1, not_x4);
38     nor(term023, not_x0, not_x2, not_x3);
39     nor(term024, not_x0, not_x2, not_x4);
40     nor(term034, not_x0, not_x3, not_x4);
41     nor(term123, not_x1, not_x2, not_x3);
42     nor(term124, not_x1, not_x2, not_x4);
43     nor(term134, not_x1, not_x3, not_x4);
44     nor(term234, not_x2, not_x3, not_x4);
45     wire not_y;
46     nor(not_y, term012, term013, term014, term023, term024, term034, term123, term124, term134, term234);
```

```

47     nor(y, not_y, not_y);
48 endmodule

```

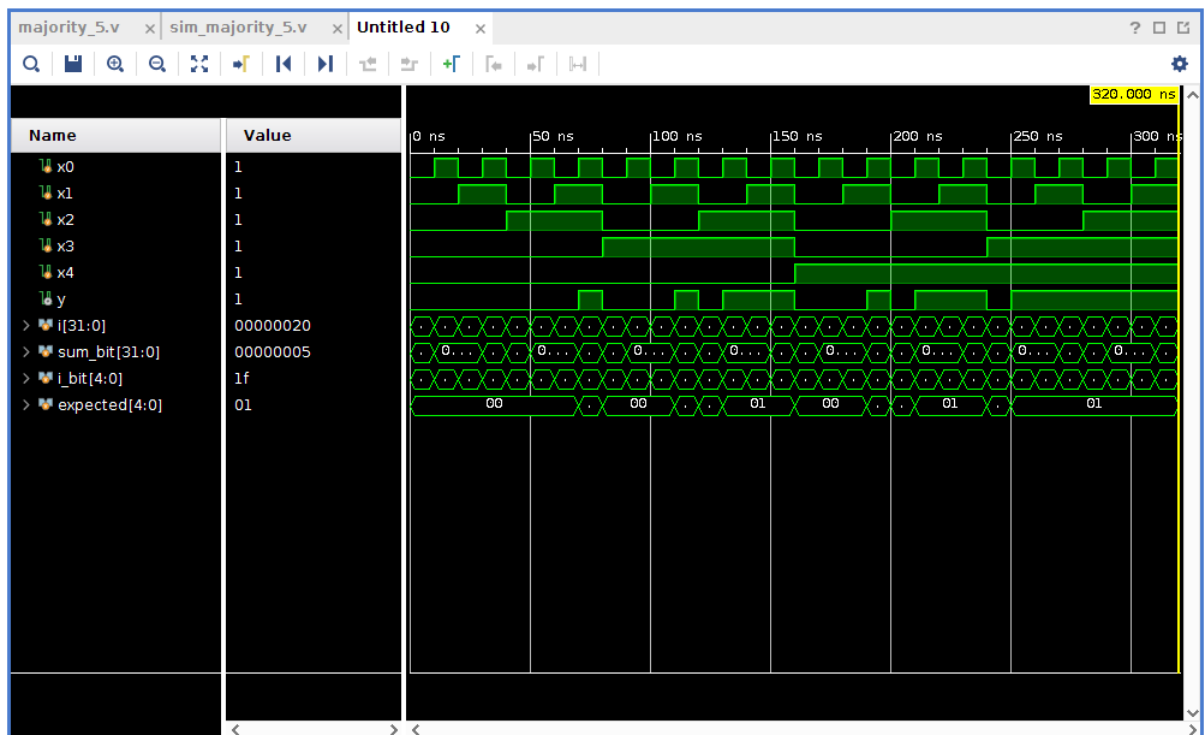
### 3.3 Код разработанного тестового окружения БОЭ

```

1  `timescale 1ns / 1ps
2  ///////////////////////////////////////////////////////////////////
3  // Company:
4  // Engineer:
5  //
6  // Create Date: 03/19/2021 05:59:36 PM
7  // Design Name:
8  // Module Name: sim_majority_5
9  // Project Name:
10 // Target Devices:
11 // Tool Versions:
12 // Description:
13 //
14 // Dependencies:
15 //
16 // Revision:
17 // Revision 0.01 - File Created
18 // Additional Comments:
19 //
20 ///////////////////////////////////////////////////////////////////
21
22
23 module sim_majority_5();
24     reg x0, x1, x2, x3, x4;
25     wire y;
26     majority_5 majority(.y(y), .x0(x0), .x1(x1), .x2(x2), .x3(x3), .x4(x4));
27
28
29     integer i, sum_bit;
30     reg[4:0] i_bit, expected;
31
32     initial begin
33         for (i = 0; i < 32; i = i + 1) begin
34             i_bit = i;
35             x0 = i_bit[0];
36             x1 = i_bit[1];
37             x2 = i_bit[2];
38             x3 = i_bit[3];
39             x4 = i_bit[4];
40             sum_bit = i_bit[0] + i_bit[1] + i_bit[2] + i_bit[3] + i_bit[4];
41             if (sum_bit >= 3) begin
42                 expected = 1;
43             end else begin
44                 expected = 0;
45             end
46             #10;
47             if (expected == y) begin
48                 $display("Test %d correct. Output = %b", i, y);
49             end else begin
50                 $display("Test %d incorrect. Expected = %b. Actual = %b", i, expected, y);
51             end
52         end
53
54         $stop;
55     end
56 endmodule

```

### 3.4 Временный диаграмм



## 4 Вывод

При выполнении лабораторной работы, я познакомился с языком Verilog: его основным синтаксисом, типами, основными операциями и некоторыми базовыми операторами управления. Код, который мне нужно написать, прост и не слишком много. Большую часть времени в лаборатории я тратил на настройку окружения и схемотехнику. Схема стала сложнее, так как мне разрешено использовать только 1 универсальный логический базис. На мой взгляд, схема получается немного сложной. После лабораторной работы я понял основы языка и причину, по которой мы используем только логический базис вместо других.