Университет ИТМО

Факультет Программной инженерии и компьютерной техники

Лабораторная работа №1

по дисциплине “Функциональная схемотехника”

Введение в проектирование цифровых

интегральных схем

Выполнил: Иван Чепрасов

Группа: P33022

г. Санкт-Петербург

2021 г

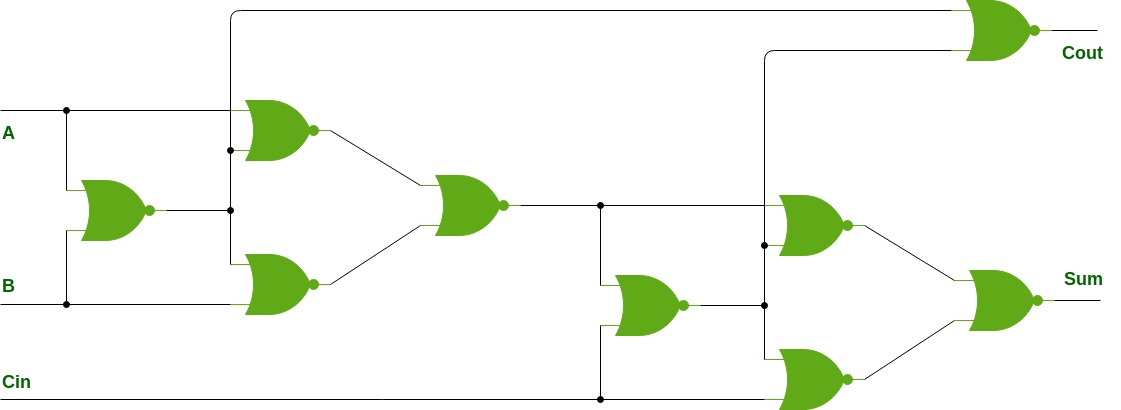
|  |  |  |
| --- | --- | --- |
| № варианта | Логический базис | БОЭ |
| 5 | NOR | Преобразователь BCD-кода в двоичный код (числа от 0 до 99) |

**Принцип работы разработанной программы**

Программа разделена на два модуля: основной, содержащий порты ввода-вывода и вспомогательные регистры для хранения промежуточного результата, и служебный, содержащий полный поразрядный двоичный сумматор, выполненный в нужном по варианту базисе. Программа реализует следующий алгоритм

* Выделение старшего числа (старшие 4 разряда) и младшего числа (младшие 4 разряда) из значений входных портов
* Сложение старшего числа с собой 10 раз (умножение на 10)
* Запись в массив результата промежуточных данных
* Сложение промежуточного результата с младшим числом
* Вывод результата на соответствующие порты вывода

**Схема сумматора**



**Листинг разработанных модулей**

module binary\_adder(

input a,

input b,

input c,

output s,

output p

);

wire d, e, f, g, h, i, j;

nor(d, a, b);

nor(e, d, a);

nor(f, d, b);

nor(g, e, f);

nor(h, g, c);

nor(i, g, h);

nor(j, c, h);

nor(s, i, j);

nor(p, d, h);

endmodule

module bcd\_to\_binary(

input a,

input b,

input c,

input d,

input e,

input f,

input g,

input h,

output reg i,

output reg j,

output reg k,

output reg l,

output reg m,

output reg n,

output reg o,

output reg p

);

reg input\_array [0:7];

reg result [0: 7];

integer outer\_loop, inner\_loop;

reg adder0, adder1, carry0;

wire sum, carry1;

binary\_adder ba0 (adder0, adder1, carry0, sum, carry1);

initial begin

adder0 = 0; adder1 = 0; carry0 = 0;

#1

input\_array[0] = d; input\_array[1] = c; input\_array[2] = b; input\_array[3] = a; input\_array[4] = 0;

input\_array[5] = 0; input\_array[6] = 0; input\_array[7] = 0;

result[0] = 0; result[1] = 0; result[2] = 0; result[3] = 0; result[4] = 0; result[5] = 0; result[6] = 0;

result[7] = 0;

for (outer\_loop = 0; outer\_loop < 10; outer\_loop = outer\_loop + 1) begin

for (inner\_loop = 0; inner\_loop < 8; inner\_loop = inner\_loop + 1) begin

adder0 = input\_array[inner\_loop];

adder1 = result[inner\_loop];

carry0 = carry1;

#1

result[inner\_loop] = sum;

end

end

input\_array[0] = h; input\_array[1] = g; input\_array[2] = f; input\_array[3] = e;

adder0 = 0; adder1 = 0; carry0 = 0;

#1

for (inner\_loop = 0; inner\_loop < 8; inner\_loop = inner\_loop + 1) begin

adder0 = input\_array[inner\_loop];

adder1 = result[inner\_loop];

carry0 = carry1;

#1

result[inner\_loop] = sum;

end

i = result[7]; j = result[6]; k = result[5]; l = result[4]; m = result[3]; n = result[2]; o = result[1];

p = result[0];

$display(i,j,k,l,m,n,o,p);

end

endmodule

**Листинг разработанных тестов**

module binary\_adder\_tb();

reg a, b, c;

wire s, p;

binary\_adder ba0 (a, b, c, s, p);

initial begin

a = 0; b = 0; c=0;

#1 a=0; b=0; c=1;

#1 a=0; b=1; c=0;

#1 a=0; b=1; c=1;

#1 a=1; b=0; c=0;

#1 a=1; b=0; c=1;

#1 a=1; b=1; c=0;

#1 a=1; b=1; c=1;

end

endmodule

module bcd\_to\_binary\_tb();

reg a, b, c, d, e, f, g, h;

wire i, j, k, l, m, n, o, p;

bcd\_to\_binary btb0 (a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p);

initial begin

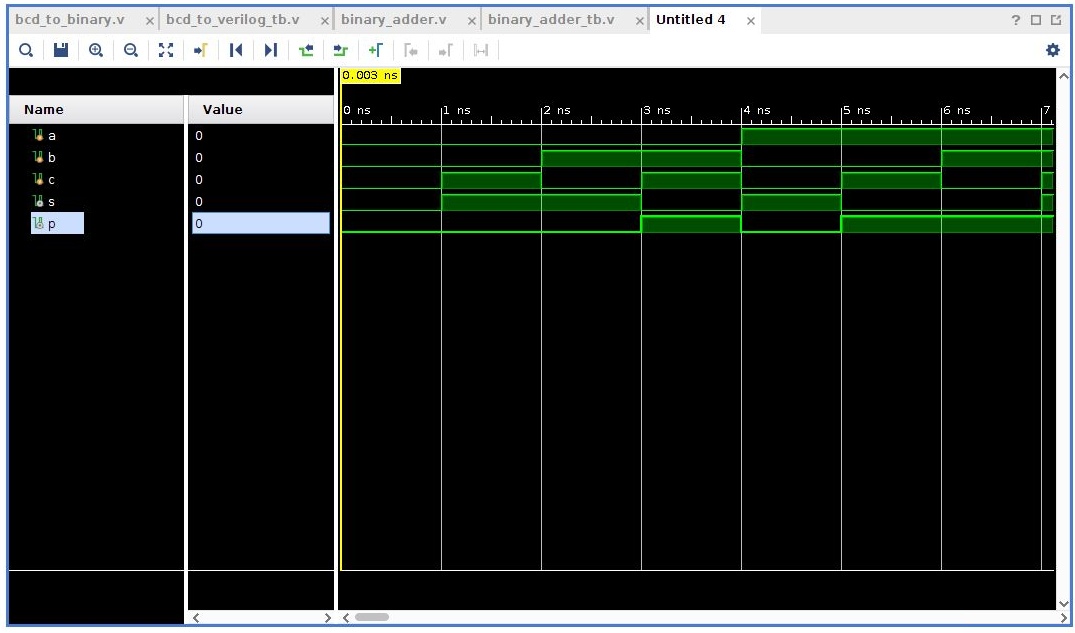
a=0; b=1; c=1; d=0; e=1; f=0; g=0; h=1;

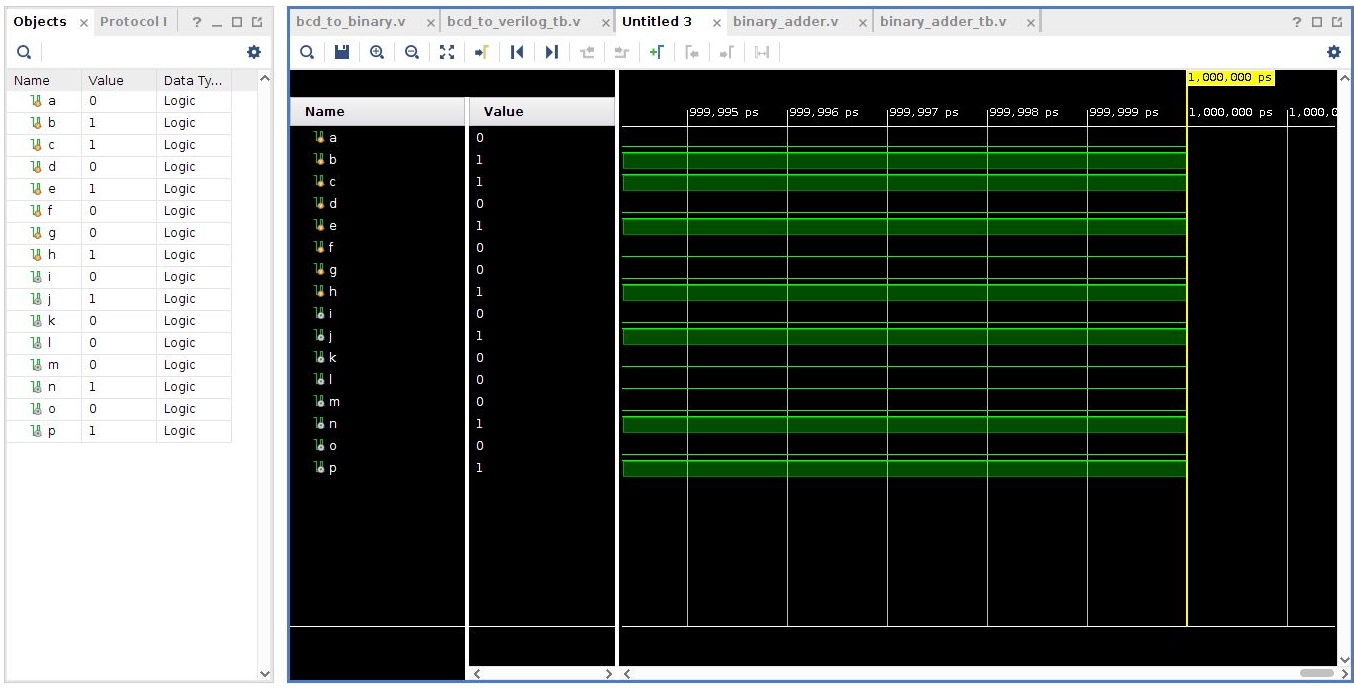
end

endmodule

**Временные диаграммы запущенных тестов**

Сумматор



Конвертер

**Вывод результата в консоль (дополнительно):**

https://sun9-65.userapi.com/impf/TClxY4obGWxGrCELwdUH_hOrMoAWUFBGo4LBhQ/qA1M7SydtFg.jpg?size=590x33&quality=96&sign=39a34bc561ab5d587e52a7b460853a4b&type=album

**Вывод:**

Я познакомился с языком Verilog. Использовал массивы регистров, вложенные циклы, логические элементы для формирования схемы. Столкнулся со следующими трудностями: 1) Определение оптимального способа взаимодействия с ide (установка на устройстве, подключение по ssh, сессии x2go). Остановился на последнем варианте из-за требовательности данной среды, медленной работе по ssh.

2) Получение промежуточного результата. Решение – внедрение задержек по времени после ввода значений во входные порты сумматора при помощи #.