**Министерство образования и науки Российской Федерации**

**Национальный исследовательский университет**

**“Московский институт электронной техники”**

**Институт микроприборов и систем управления им. Л. Н. Преснухина**

**Лабораторная работа №1**

**По дисциплине “Интерфейсы вычислительных систем”**

**Тема: “Изучение и реализация процедур арифметического кодирования. Функциональное тестирование”**

**Выполнил:**

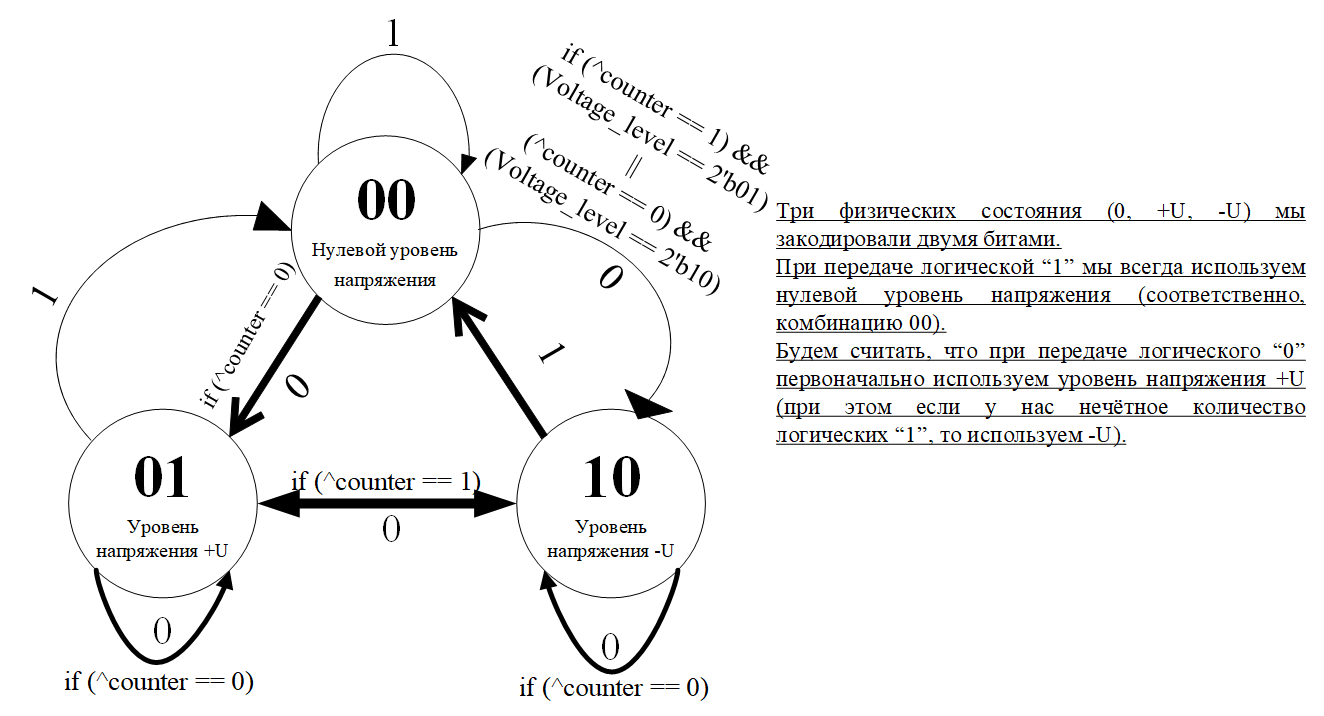
Студент группы ИВТ-32

Голев Андрей Дмитриевич

Москва, Зеленоград, 2022 г.

***Цель работы*:** изучение, реализация и функциональное тестирование методов битового кодирования информации.

**Задание №1.** Разработка графа состояний процедуры битового кодирования



**Задание №2.** Разработка кодера на языке Verilog

module Coder (

input bit\_i, clk\_i,

output reg [1:0] code\_o

);

reg [31:0] counter = 0;

reg [1:0] voltage\_level = 2'b01;

always @( posedge clk\_i ) begin

if ( bit\_i ) begin

code\_o <= 2'b00;

counter <= counter + 1;

if ( counter == 32'd4294967295 ) counter <= 0;

end else begin

if ( ^counter ) code\_o <= ~voltage\_level;

else code\_o <= voltage\_level;

voltage\_level <= code\_o;

counter <= 0;

end

end

endmodule

**Задание №3.** Разработка Testbench для проверки работы модуля кодера

`timescale 1ns / 1ps //

`define N 23

module Coder\_tb();

//////////////////////////////

reg CLK\_tb;

parameter PERIOD = 10;

always begin

CLK\_tb = 1'b0;

#( PERIOD/2 ) CLK\_tb = 1'b1;

#( PERIOD/2 );

end

//////////////////////////////

reg [`N - 1:0] bit\_sequence\_tb; // Последовательность битов

reg bit\_tb;

wire [1:0] code\_tb;

reg [1:0] encoded\_sequence\_tb [0:22]; // Закодированная последовательность битов

reg [22:0] decoded\_sequence\_tb; // Декодированная последовательность битов (для проверки)

integer i; // Счётчик для цикла for

Coder CD ( .clk\_i(CLK\_tb), .bit\_i(bit\_tb), .code\_o(code\_tb) );

initial begin

bit\_sequence\_tb = `N'd8201481;

$display ( "\n////\nКодирование числа %d", bit\_sequence\_tb );

// Цикл кодирования битовой последовательности

for ( i = 0; i < 23; i = i + 1 ) begin

bit\_tb = bit\_sequence\_tb[i];

@( posedge CLK\_tb ); #1; // Ждём сигнал CLOCK и ещё 1 нс (для иммитации реальной задержки в схеме)

encoded\_sequence\_tb[i] = code\_tb;

end

// Цикл декодирования

for ( i = 0; i < 23; i = i + 1 ) begin

if ( encoded\_sequence\_tb[i] )

decoded\_sequence\_tb[i] = 0;

else

decoded\_sequence\_tb[i] = 1;

end

$display ( "\nРезультат декодирования: %d\n////\n", decoded\_sequence\_tb );

$finish;

end

endmodule