

Задание №1

Инвертор без задержки

Задачи

- написание тестового окружения;
- моделирование;
- верификация

```
// Инвертор
module inverter (x, y);
    input x;
    output y;

    assign y = ~x;
endmodule

// Тестовое окружение
module tb_inverter;
    reg x;
    wire y;

    inverter i1(x, y);

    initial begin
        x = 0;
        #10 x = 1;
        #10 x = 0;
        #10 x = 1;
        #10 $finish;
    end
endmodule
```

Задание №2

4-битный счётчик с асинхронным сбросом

Задачи

- написание тестового окружения;
- моделирование;
- верификация;
- цифровой синтез;
- верификация

```
module counter(input clk, input rst, output reg [3:0] out);
    always @ (negedge clk or posedge rst) begin
        if (rst)
            out <= 0;
        else
            out <= out + 1;
    end
endmodule

module tb_counter;
    reg clk, rst;
    wire [3:0] out;

    counter dut1(clk, rst, out);

    initial begin
        clk = 0;
        rst = 0;
        #5 rst = 1;
        #10 rst = 0;
        #85 $finish;
    end

    always
        #10 clk = ~clk;
endmodule
```

Задание №3

Инвертор с задержкой

Задачи

- написание тестового окружения;
- моделирование;
- верификация

```

`timescale 1ns/1ns

// Inverter
module inverter (x, y);
    input x;
    output y;

    assign #2 y = ~x;
endmodule

```

Задание №4

Вентиль 2-И-НЕ

Задачи

- написание тестового окружения;
- моделирование;
- верификация;

```

// NAND2
module nand2 (x1, x2, y);
    input x1, x2;
    output y;

    assign y = x1 & x2;
endmodule

```

Задание №5

Статический

D-триггер

Задачи

- написание тестового окружения;
- моделирование;
- верификация;

```

// DFF
module dff (clk, d, q, nq);
    input clk, d;
    output reg q, nq;

    always @(negedge clk) begin
        q <= d;
        nq <= ~d;
    end
endmodule

```

Задание №5

Динамический

D-триггер с

асинхронным сбросом

Задачи

- написание тестового окружения;
- моделирование;
- верификация;

```

// DFF with reset
module dff (clk, d, r, q, nq);
    input clk, d, r;
    output reg q, nq;

    always @(negedge clk or posedge r) begin
        if (r == 1) begin
            q <= 1;
            nq <= 0;
        end
        else begin
            q <= d;
            nq <= ~d;
        end;
    end
endmodule

```