

Лабораторная работа № 5. Исследование регистров.

Цель работы: знакомство с работой параллельных и последовательных регистров. Для изучения предлагаются регистры: регистр сдвига на 8 разрядов последовательный, защелка на 4 разряда, а также параллельный 4-разрядный буферный регистр.

Задание на работу в лаборатории.

1. Создать проект. Открыть Verilog HDL файл и записать **программу 5.1**, отражающую функционирование последовательного регистра на 8 разрядов. Сохранить файл, установить его старшим в иерархии и откомпилировать.
2. Пользуясь «Приложением 2», получить диаграммы при **интервалах синхроимпульса – 20нс, данные – уровень «1» на интервалах от 10нс до 50 нс и от 130нс до 200нс.**
3. Открыть новый Verilog HDL файл и записать **программу 5.2**, отражающую функционирование регистра-защелки на 4 разряда. Сохранить файл под именем модуля, установить его старшим в иерархии и откомпилировать.
4. Получить временные диаграммы для регистра-защелки, задав следующие параметры: **синхронизация – 20нс; данные – d_in[0], d_in[2] – 35нс, d_in[1],d_in[3] – 50нс**
5. Открыть новый Verilog HDL файл и записать **программу 5.3**, отражающую функционирование параллельного буферного регистра на 4 разряда. Сохранить файл под именем модуля, установить его старшим в иерархии и откомпилировать.
6. Получить диаграмму работы устройства при следующих параметрах: **синхронизация – 20нс; данные – d_in[0], d_in[2] – 35нс, d_in[1],d_in[3] – 50нс; ena – 170нс; rd – 200нс.**

Отчет должен содержать программы работы исследуемых устройств и их временные диаграммы.

Программа 5.1

```
module shift_rgstr
#(parameter N=8)
(input clk,
Input d_in,
output d_out);
reg [N-1:0] d;
```

```

assign d_out = d[N-1];
always @ (posedge clk)
begin
    d[N-1:1] <= d[N-2:0];
    d[0] <= d_in;
end
endmodule

```

Программа 5.2

```

module latch_rgstr
#(parameter N=4)
(input [N-1:0] d_in,
input clk,
output [N-1:0]d_out);
reg [N-1:0]q;
assign d_out=q;
always@ (posedge clk)
begin
    q<=d_in;
end
endmodule

```

Программа 5.3

```

module buf_rgstr
#(parameter N=4)
(input [N-1:0] d_in,
input clk, ena, res, rd,
output [N-1:0] d_out);
reg [N-1:0]w;
assign d_out=(rd)? (w) : (0);
always@ (posedge clk, negedge res)
begin
    if (!res)
        w<=0;
    else if (clk)
        begin
            if (ena)
                w<=d_in;
            else
                w<=w;
        end
end
endmodule

```

