

В качестве библиотеки FP арифметики была использована:

<https://github.com/choukusepurva/Floating Point Unit Hardware Implementation Using SystemVerilog/tree/main/ADD SUB>

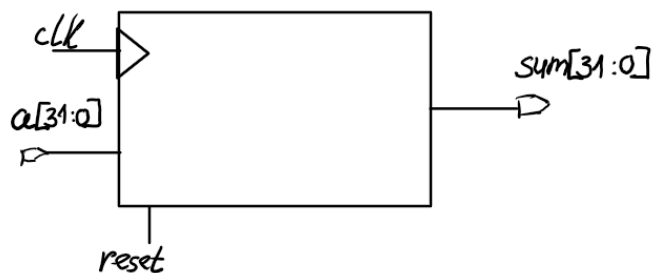
Симуляция проводилась в среде Vivado (имя проекта - project_4).

В рамках первого пункта был написан модуль, использующий библиотечный модуль сложения:

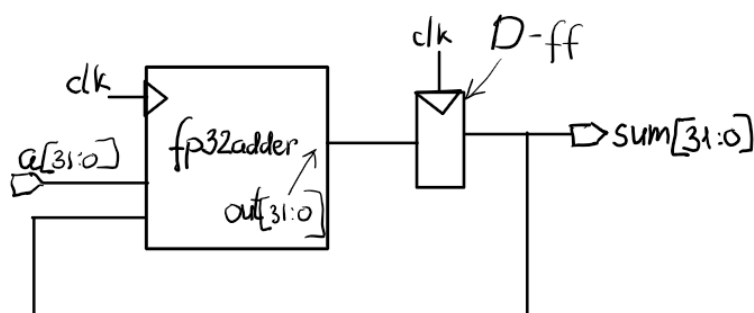
```
module add32float(  
    input logic [31:0] a_fpn,  
    input logic [31:0] b_fpn,  
    input clk, rst, signal,  
    output logic [31:0] out);  
  
    fp_add_sub calc(clk, rst, signal, a_fpn, b_fpn, out);  
endmodule
```

а также тестбенч по аналогии с библиотечным примером.

В рамках второго пункта (сумма ряда) была придумана схема, имеющая один информационный вход (a) и один информационный выход (sum), а также принимающая тактовый сигнал и reset



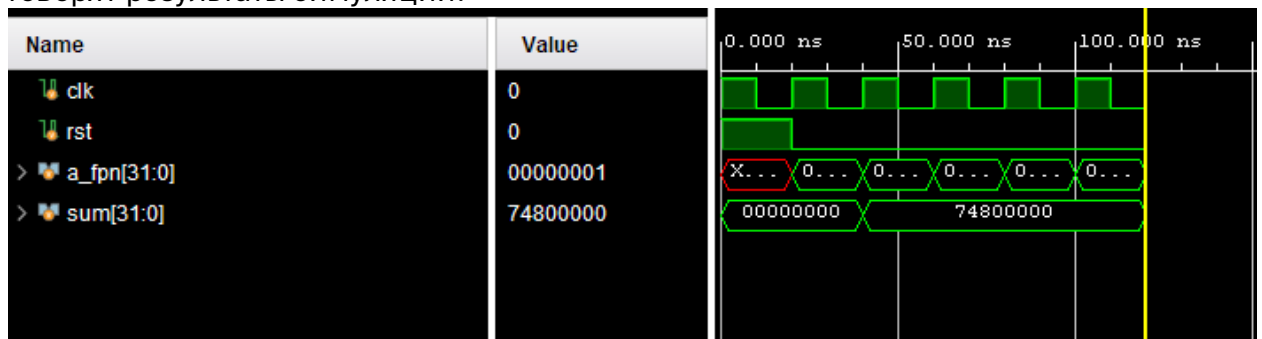
Реализация включает в себя используемый ранее модуль сложения, но с применением регистра, хранящего результат суммы, а вход b редуцирован, и вместо внешнего значения на него подается число, хранящееся в регистре:



Код модуля:

```
module float_series(  
    input logic [31:0] a_fpn,  
    input clk, rst,  
    output logic [31:0] sum  
);  
  
    logic [31:0] out;  
    logic [31:0] b_fpn;  
    assign signal = 1'b1; //только на сложение  
  
    always_ff @(posedge rst,posedge clk )  
    begin  
        if(rst) sum <= 0;  
        else sum <= out;  
    end  
    assign b_fpn = sum;  
  
    fp_add_sub  calc(clk, rst, signal, a_fpn, sum, out);  
  
endmodule
```

Также был написан тестбенч, однако не удалось реализовать функционал, о чем говорят результаты симуляции.



Пусть к коду в проекте project_4\project_4.srscs\sources_1\new