**Отчёт по лабораторной работе**

по дисциплине

«Языки моделирования и описания цифровой аппаратуры»

**Программная и аппаратная реализация алгоритма сортировки вставками**

Студент гр. 5130904/30008 Ребдев П.А

Студент гр. 5130904/30008 Мунгой Шеллер Валмиро Да Линда

Проверил: Амосов В.В

Оглавление

[1. Техническое задание 3](#__RefHeading___Toc163_1041169204)

[2. Описание алгоритма 3](#__RefHeading___Toc175_1041169204)

[3. Аппаратная реализация 4](#__RefHeading___Toc177_1041169204)

[3.1 Блок-схема 4](#__RefHeading___Toc179_1041169204)

[3.2 Verilog код 4](#__RefHeading___Toc181_1041169204)

[3.3 Симуляция (тестирование) 6](#__RefHeading___Toc183_1041169204)

[3.4 Технологическая схема 7](#__RefHeading___Toc185_1041169204)

[6. Ручной расчёт 9](#__RefHeading___Toc199_1041169204)

## 1. Техническое задание

1) Разработать аппаратную реализацию сортировки вставсками массива на языке Verilog

2) Произвести симуляцию Verilog кода

3) Создать RTL схему в среде Quartus II

4) Создать технологическую схему

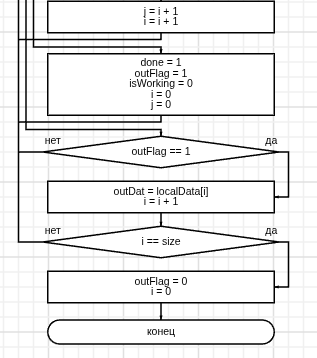
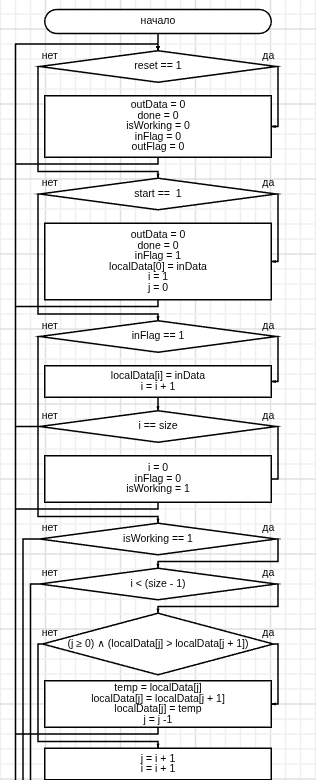
5) Сделать ручной расчёт сортировки

## 2. Описание алгоритма

На вход алгоритма подаётся последовательность n чисел. Алгоритм состоит из прохода по массиву. На каждом шагу следующий элемент сравнивается с текущим, если порядок неверный, то следующий элемент вставляется в отсортированную часть. Проход по массиву проводиться однократно, на каждом шаге отсортированная часть увеличивается, а не отсортированная уменьшается

## 3. Аппаратная реализация

### 3.1 Блок-схема



### 3.2 Verilog код

module insert\_sort(clk, start, reset, inData, done, outData);

parameter size = 8;

input clk, start, reset;

input [31 : 0] inData;

output reg done;

output reg [31 : 0] outData;

integer localData[(size - 1) : 0];

integer i = 0, j = 0, temp = 0;

reg isWorking = 0, inFlag = 0, outFlag = 0;

always @(posedge clk)

begin

if (reset)

begin

outData = 0;

done = 0;

isWorking = 0;

inFlag = 0;

outFlag = 0;

end

else if (start)

begin

outData = 0;

done = 0;

inFlag = 1;

localData[0] = inData;

i = 1;

j = 0;

end

else if (inFlag)

begin

localData[i] = inData;

i = i + 1;

if (i == size)

begin

i = 0;

inFlag = 0;

isWorking = 1;

end

end

else if (isWorking == 1)

begin

if (i < (size - 1))

begin

if ((j >= 0) && (localData[j] > localData[j + 1]))

begin

temp = localData[j];

localData[j] = localData[j + 1];

localData[j + 1] = temp;

j = j - 1;

end

else

begin

j = i + 1;

i = i + 1;

end

end

else

begin

done = 1;

outFlag = 1;

isWorking = 0;

i = 0;

j = 0;

end

end

else if (outFlag)

begin

outData = localData[i];

i = i + 1;

if (i == size)

begin

outFlag = 0;

i = 0;

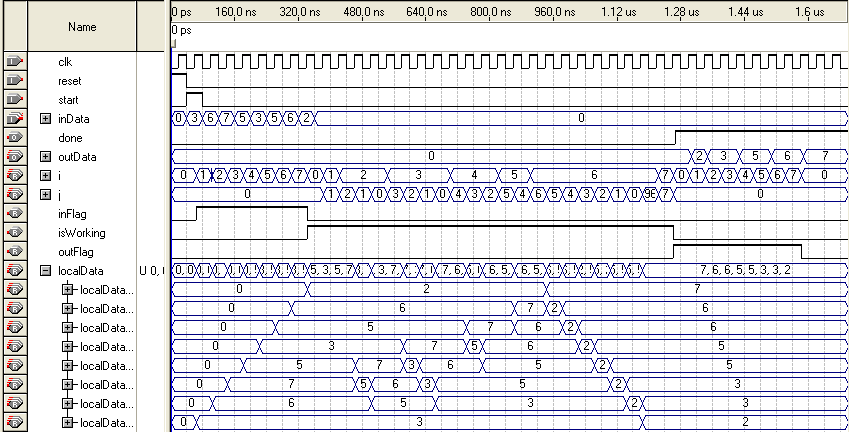
end

end

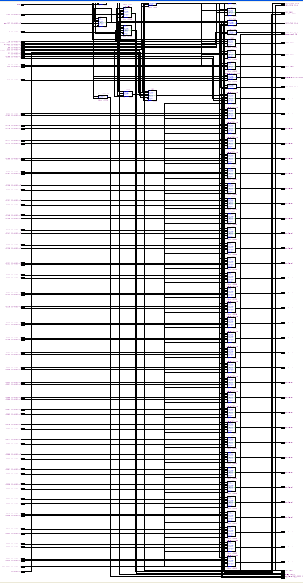
end

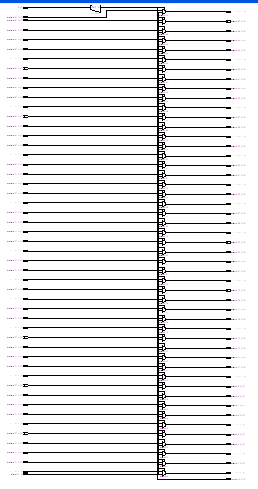
endmodule

### 3.3 Симуляция (тестирование)

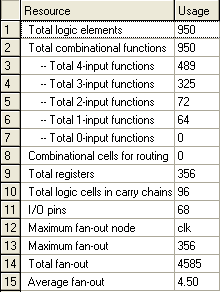
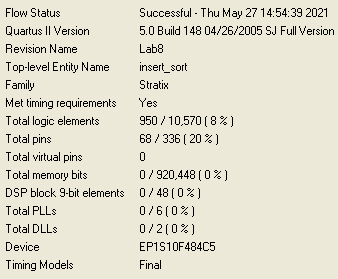


### 3.4 Технологическая схема

**3.5 RTL схема**

****

**3.6 Отчёт в среде Quartus**



## 6. Ручной расчёт

Возьмём 8 случайно сгенерированых тестовых чисел: 3 6 7 5 3 5 6 2

