Министерство образования и науки РФ

Санкт-Петербургский политехнический университет Петра Великого

Институт компьютерных наук и технологий

Высшая школа программной инженерии

ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ

по дисциплине «Микроэлектроника, схемотехника

и проектирование устройств вычислительной техники»

**Программная и аппаратная реализация алгоритма**

**вычисления определителя матрица 3 на 3**

Выполнили:

Студенты гр. 3530904/90002

Ли Ицзя, Мэн Цзянин, Го Синлкн

Проверил:

В. В. Амосов

Санкт-Петербург 2020 г.

# **Описание алгоритма**

Алгоритм:

Данный алгоритм позволяет посчитать определитель матрицы 3x3. Для начала запишем формулу, по которой вычисляется определитель:

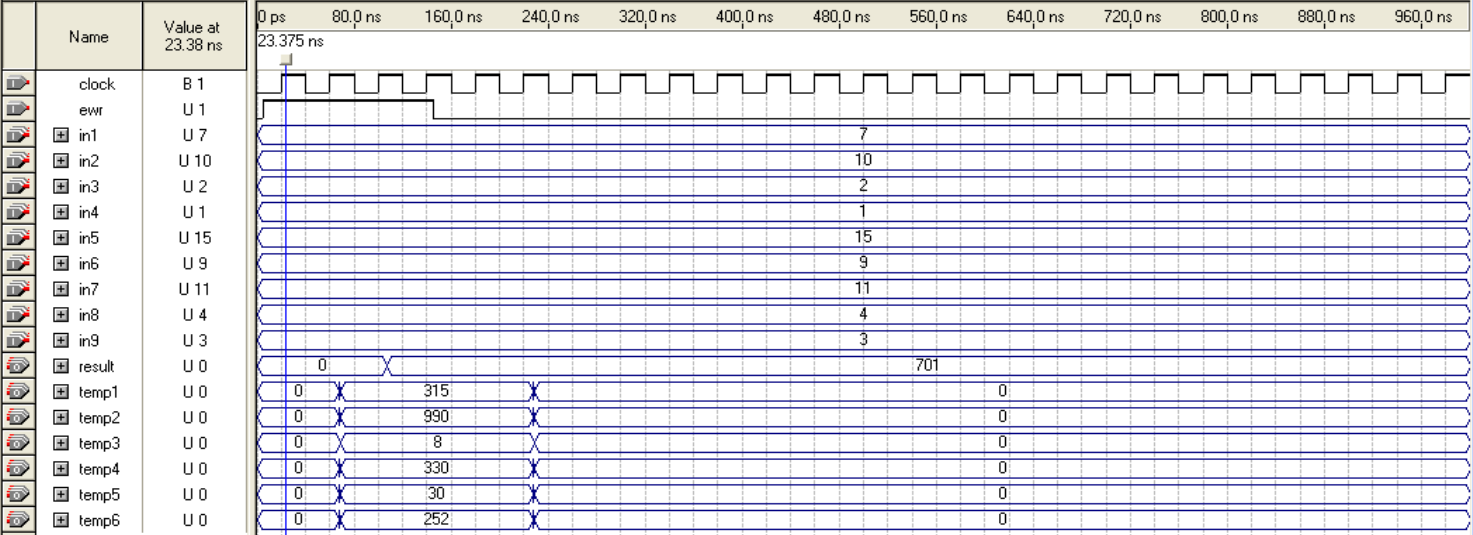
Очевидно, необходимо сначала вычислить произведения, а затем суммировать полученные значения.

## **Verilog HDL код**

|  |  |
| --- | --- |
|  | module plus\_mult(X, Y, Z,  tempResult, clock);  input[15:0] X, Y, Z ;  inout clock;  output [15:0] tempResult;  reg [15:0] product;  always @ (posedge clock)  begin  product = X \* Y \* Z;  end  assign tempResult = product;  endmodule  //get determinant  module determinant(result,  in1, in2, in3,  in4, in5, in6,  in7, in8, in9,  temp1, temp2, temp3, temp4, temp5, temp6,  ewr, clock);  output [15:0] result;  input [15:0] in1;  input [15:0] in2;  input [15:0] in3;  input [15:0] in4;  input [15:0] in5;  input [15:0] in6;  input [15:0] in7;  input [15:0] in8;  input [15:0] in9;  output [15:0] temp1;  output [15:0] temp2;  output [15:0] temp3;  output [15:0] temp4;  output [15:0] temp5;  output [15:0] temp6;  input ewr;  input clock;  reg [15:0] arr[9 : 1];  reg [15:0] tempResult[5 : 0];  reg [15:0] sumary;  //integer sumary;  integer i;  integer j;  plus\_mult (arr[1], arr[5], arr[9], tempResult[0], clock);  plus\_mult (arr[2], arr[6], arr[7], tempResult[1], clock);  plus\_mult (arr[4], arr[8], arr[3], tempResult[2], clock);  plus\_mult (arr[3], arr[5], arr[7], tempResult[3], clock);  plus\_mult (arr[2], arr[4], arr[9], tempResult[4], clock);  plus\_mult (arr[6], arr[8], arr[1], tempResult[5], clock);  always @ (posedge clock)  begin  if(ewr)  begin  arr[1] = in1;  arr[2] = in2;  arr[3] = in3;  arr[4] = in4;  arr[5] = in5;  arr[6] = in6;  arr[7] = in7;  arr[8] = in8;  arr[9] = in9;  sumary = tempResult[0] + tempResult[1] + tempResult[2] - tempResult[3] - tempResult[4] - tempResult[5];  end  else  begin  for (i=1;i<=9;i=i+1)  begin  for (j=0;j<=15;j=j+1)  arr[i][j] = 0;  end  end  end  assign result = sumary;  assign temp1 = tempResult[0];  assign temp2 = tempResult[1];  assign temp3 = tempResult[2];  assign temp4 = tempResult[3];  assign temp5 = tempResult[4];  assign temp6 = tempResult[5];  endmodule |

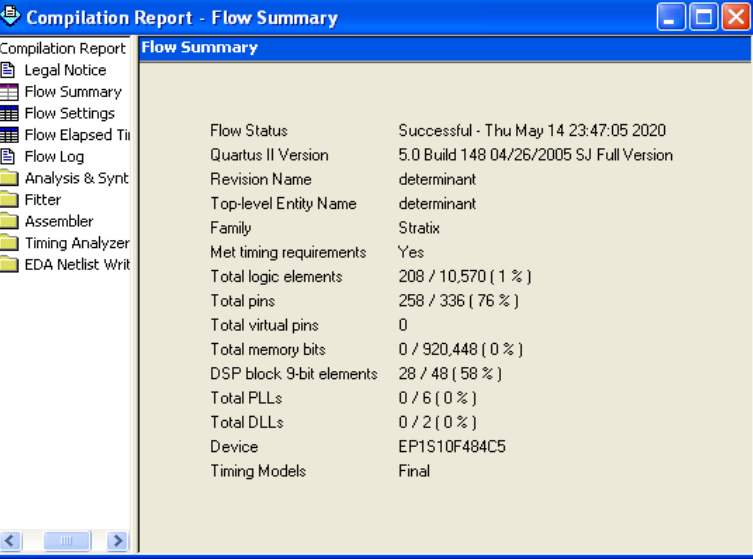
## **Симуляция (тестирование)**

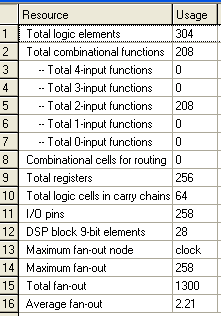
Waveform – диаграммы

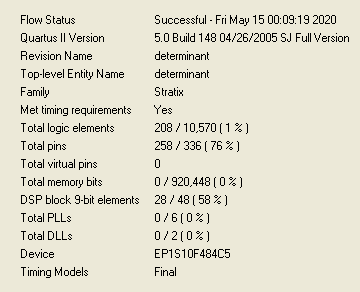


## **Проверка на синтезируемость**

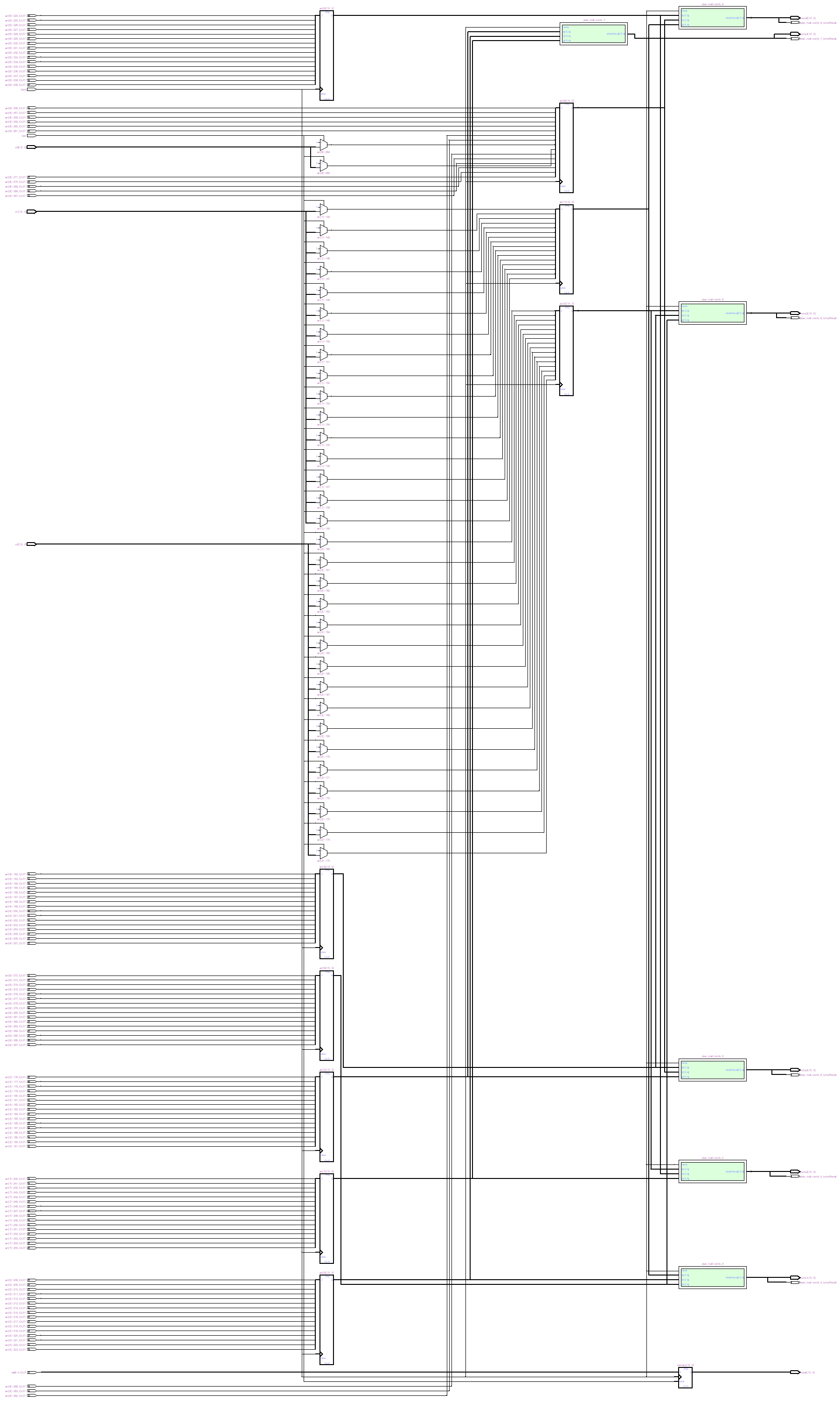
## **Отчёт в среде Quartus II**







## **Технологическая схема**



## **RTL схема**

