Міністерство освіти і науки України

Національний технічний університет України

«Київський політехнічний інститут»

Кафедра КЕОА

**Лабораторна робота №1**

**з курсу: «Апаратні прискорювачі обчислень на мікросхемах програмованої логіки»**

Виконав:

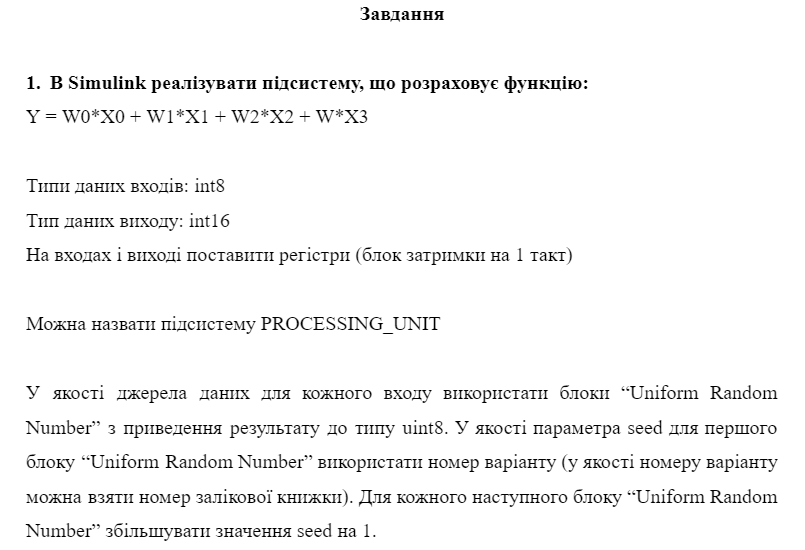
студент ІII-го курсу ФЕЛ

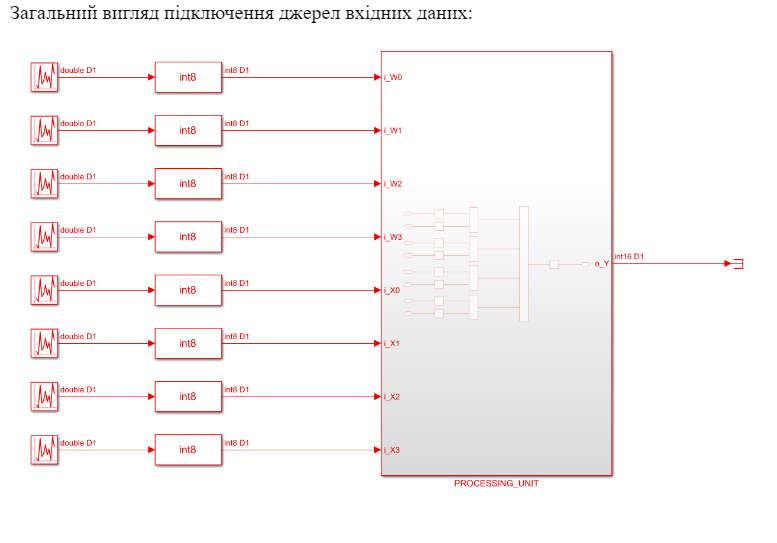
група ДК-02

Вакуленко М. В.

29.10.2022

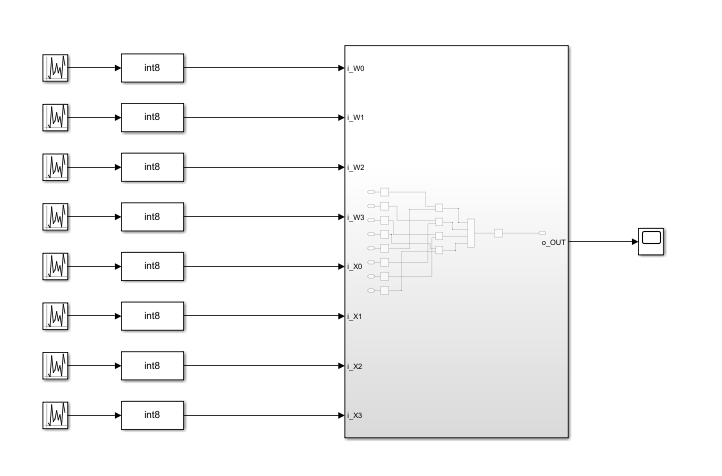
Київ-2022

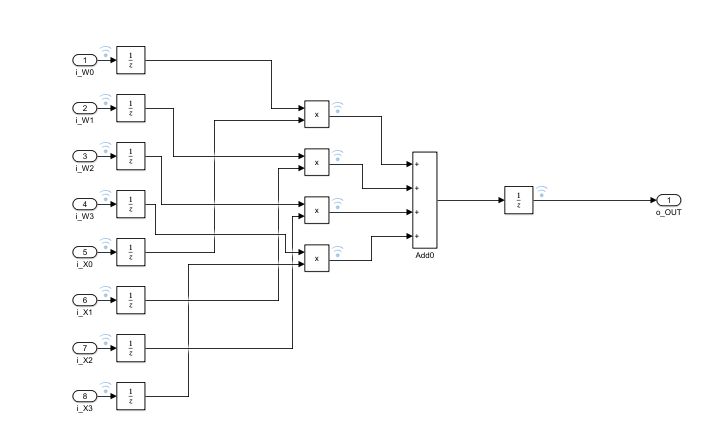
****

**хід роботи**

1. **Підсистема розрахунку функції в Simulink:**

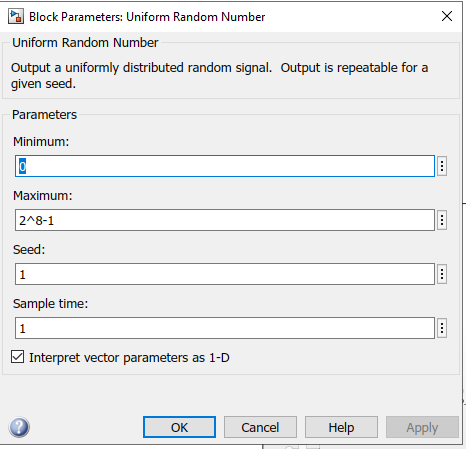
Схема(загальна):

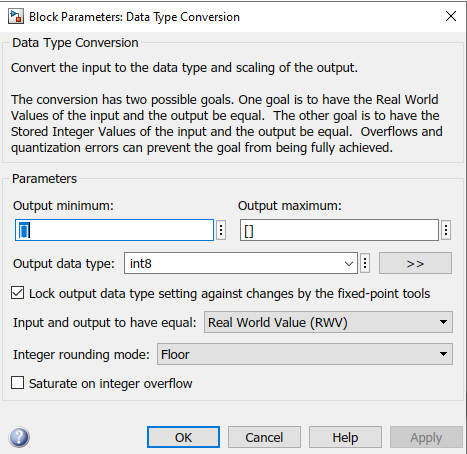
 Схема(підсистеми):

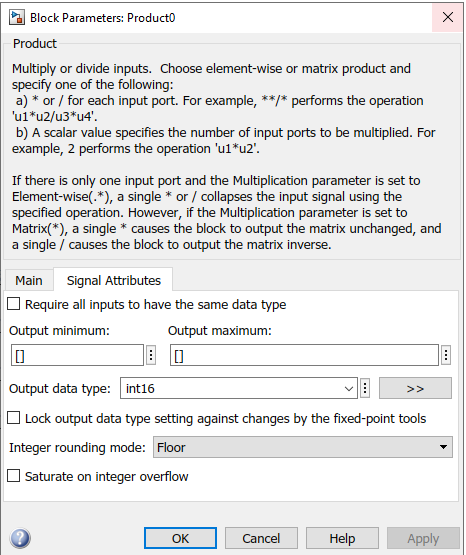
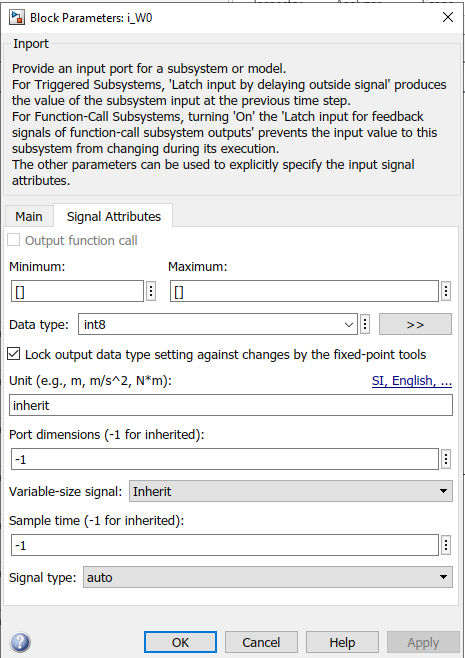
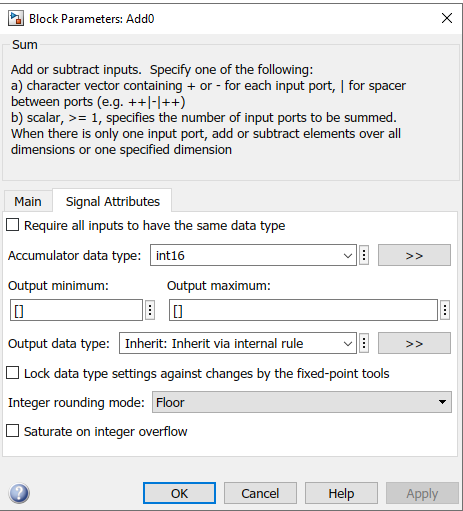
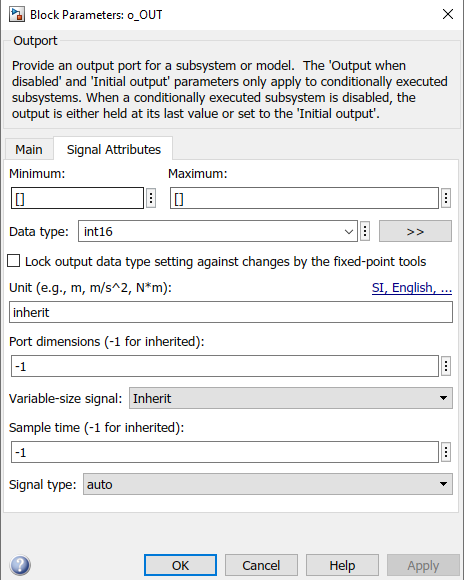


Налаштування компонентів:

Параметр seed відповідно 1 варіанту збільшується від 1 до для кожного наступного джерела випадкових чисел.

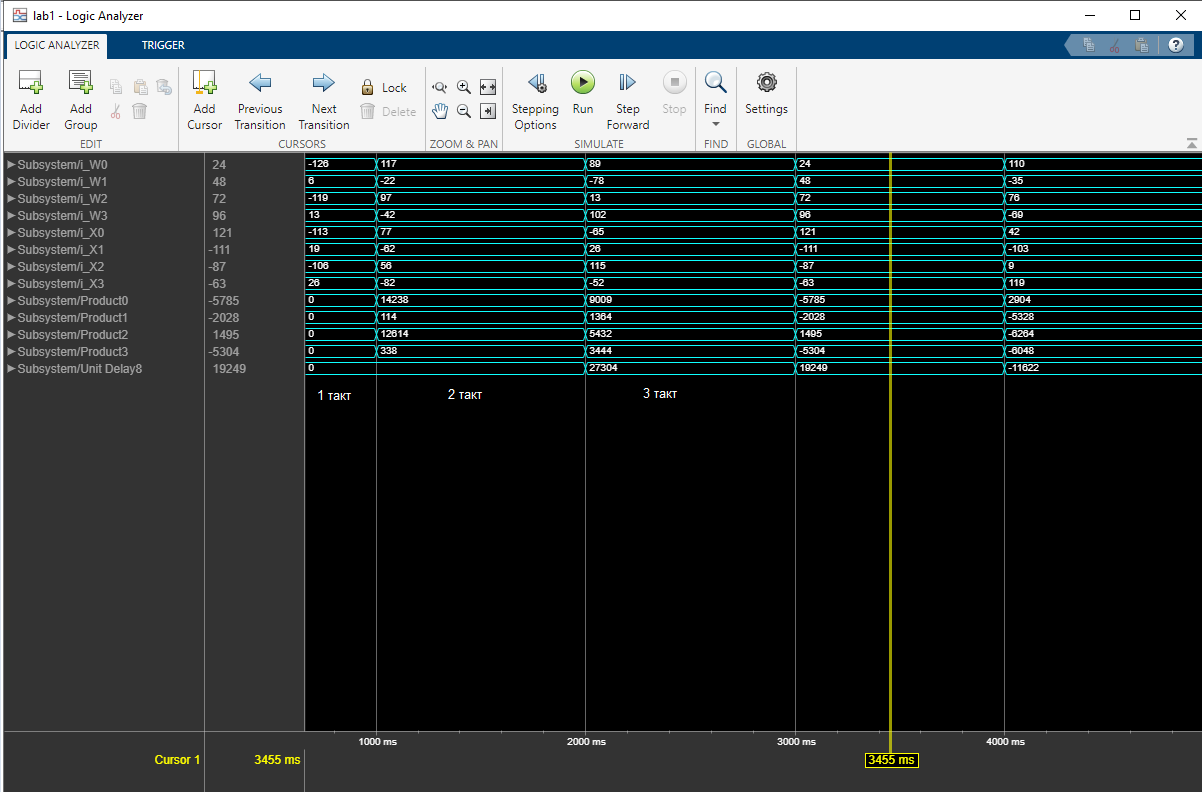




**2. В логічному аналазаторі переглянути дані на входах і на виході створеної підсистеми у знаковому десятковому поданні (форматі).**

Результат симуляції:



Як можна побачити сигнал на виході затримується на два такти відносно вхідного сигналу.

**3. Додати у звіт згенерований код на Verilog та результат синтезу згенерованого коду в Quartus для створеної підсистеми (звіт по апаратним витратам, результат виклику RTL Viewer).**

Згенерований Verilog код має наступний вигляд:

// -------------------------------------------------------------

//

// File Name: hdlsrc\lab1\Subsystem.v

// Created: 2022-10-18 11:40:09

//

// Generated by MATLAB 9.12 and HDL Coder 3.20

//

//

// -- -------------------------------------------------------------

// -- Rate and Clocking Details

// -- -------------------------------------------------------------

// Model base rate: 1

// Target subsystem base rate: 1

//

//

// Clock Enable Sample Time

// -- -------------------------------------------------------------

// ce\_out 1

// -- -------------------------------------------------------------

//

//

// Output Signal Clock Enable Sample Time

// -- -------------------------------------------------------------

// o\_OUT ce\_out 1

// -- -------------------------------------------------------------

//

// -------------------------------------------------------------

// -------------------------------------------------------------

//

// Module: Subsystem

// Source Path: lab1/Subsystem

// Hierarchy Level: 0

//

// -------------------------------------------------------------

`timescale 1 ns / 1 ns

module Subsystem

(i\_CLK,

i\_RST\_N,

i\_CLK\_EN,

i\_W0,

i\_W1,

i\_W2,

i\_W3,

i\_X0,

i\_X1,

i\_X2,

i\_X3,

ce\_out,

o\_OUT);

input i\_CLK;

input i\_RST\_N;

input i\_CLK\_EN;

input signed [7:0] i\_W0; // int8

input signed [7:0] i\_W1; // int8

input signed [7:0] i\_W2; // int8

input signed [7:0] i\_W3; // int8

input signed [7:0] i\_X0; // int8

input signed [7:0] i\_X1; // int8

input signed [7:0] i\_X2; // int8

input signed [7:0] i\_X3; // int8

output ce\_out;

output signed [15:0] o\_OUT; // int16

wire enb;

reg signed [7:0] Unit\_Delay\_out1; // int8

reg signed [7:0] Unit\_Delay2\_out1; // int8

reg signed [7:0] Unit\_Delay3\_out1; // int8

reg signed [7:0] Unit\_Delay4\_out1; // int8

reg signed [7:0] Unit\_Delay1\_out1; // int8

wire signed [15:0] Product0\_out1; // int16

reg signed [7:0] Unit\_Delay5\_out1; // int8

wire signed [15:0] Product1\_out1; // int16

wire signed [15:0] Add0\_stage2\_add\_temp; // sfix16

wire signed [16:0] Add0\_op\_stage1; // sfix17

reg signed [7:0] Unit\_Delay6\_out1; // int8

wire signed [15:0] Product2\_out1; // int16

wire signed [15:0] Add0\_stage3\_add\_cast; // sfix16

wire signed [15:0] Add0\_stage3\_add\_temp; // sfix16

wire signed [17:0] Add0\_op\_stage2; // sfix18

reg signed [7:0] Unit\_Delay7\_out1; // int8

wire signed [15:0] Product3\_out1; // int16

wire signed [15:0] Add0\_stage4\_add\_cast; // sfix16

wire signed [15:0] Add0\_out1; // int16

reg signed [15:0] Unit\_Delay8\_out1; // int16

assign enb = i\_CLK\_EN;

always @(posedge i\_CLK or posedge i\_RST\_N)

begin : Unit\_Delay\_process

if (i\_RST\_N == 1'b1) begin

Unit\_Delay\_out1 <= 8'sb00000000;

end

else begin

if (enb) begin

Unit\_Delay\_out1 <= i\_W0;

end

end

end

always @(posedge i\_CLK or posedge i\_RST\_N)

begin : Unit\_Delay2\_process

if (i\_RST\_N == 1'b1) begin

Unit\_Delay2\_out1 <= 8'sb00000000;

end

else begin

if (enb) begin

Unit\_Delay2\_out1 <= i\_W1;

end

end

end

always @(posedge i\_CLK or posedge i\_RST\_N)

begin : Unit\_Delay3\_process

if (i\_RST\_N == 1'b1) begin

Unit\_Delay3\_out1 <= 8'sb00000000;

end

else begin

if (enb) begin

Unit\_Delay3\_out1 <= i\_W2;

end

end

end

always @(posedge i\_CLK or posedge i\_RST\_N)

begin : Unit\_Delay4\_process

if (i\_RST\_N == 1'b1) begin

Unit\_Delay4\_out1 <= 8'sb00000000;

end

else begin

if (enb) begin

Unit\_Delay4\_out1 <= i\_W3;

end

end

end

always @(posedge i\_CLK or posedge i\_RST\_N)

begin : Unit\_Delay1\_process

if (i\_RST\_N == 1'b1) begin

Unit\_Delay1\_out1 <= 8'sb00000000;

end

else begin

if (enb) begin

Unit\_Delay1\_out1 <= i\_X0;

end

end

end

assign Product0\_out1 = Unit\_Delay\_out1 \* Unit\_Delay1\_out1;

always @(posedge i\_CLK or posedge i\_RST\_N)

begin : Unit\_Delay5\_process

if (i\_RST\_N == 1'b1) begin

Unit\_Delay5\_out1 <= 8'sb00000000;

end

else begin

if (enb) begin

Unit\_Delay5\_out1 <= i\_X1;

end

end

end

assign Product1\_out1 = Unit\_Delay2\_out1 \* Unit\_Delay5\_out1;

assign Add0\_stage2\_add\_temp = Product0\_out1 + Product1\_out1;

assign Add0\_op\_stage1 = {Add0\_stage2\_add\_temp[15], Add0\_stage2\_add\_temp};

always @(posedge i\_CLK or posedge i\_RST\_N)

begin : Unit\_Delay6\_process

if (i\_RST\_N == 1'b1) begin

Unit\_Delay6\_out1 <= 8'sb00000000;

end

else begin

if (enb) begin

Unit\_Delay6\_out1 <= i\_X2;

end

end

end

assign Product2\_out1 = Unit\_Delay3\_out1 \* Unit\_Delay6\_out1;

assign Add0\_stage3\_add\_cast = Add0\_op\_stage1[15:0];

assign Add0\_stage3\_add\_temp = Add0\_stage3\_add\_cast + Product2\_out1;

assign Add0\_op\_stage2 = {{2{Add0\_stage3\_add\_temp[15]}}, Add0\_stage3\_add\_temp};

always @(posedge i\_CLK or posedge i\_RST\_N)

begin : Unit\_Delay7\_process

if (i\_RST\_N == 1'b1) begin

Unit\_Delay7\_out1 <= 8'sb00000000;

end

else begin

if (enb) begin

Unit\_Delay7\_out1 <= i\_X3;

end

end

end

assign Product3\_out1 = Unit\_Delay4\_out1 \* Unit\_Delay7\_out1;

assign Add0\_stage4\_add\_cast = Add0\_op\_stage2[15:0];

assign Add0\_out1 = Add0\_stage4\_add\_cast + Product3\_out1;

always @(posedge i\_CLK or posedge i\_RST\_N)

begin : Unit\_Delay8\_process

if (i\_RST\_N == 1'b1) begin

Unit\_Delay8\_out1 <= 16'sb0000000000000000;

end

else begin

if (enb) begin

Unit\_Delay8\_out1 <= Add0\_out1;

end

end

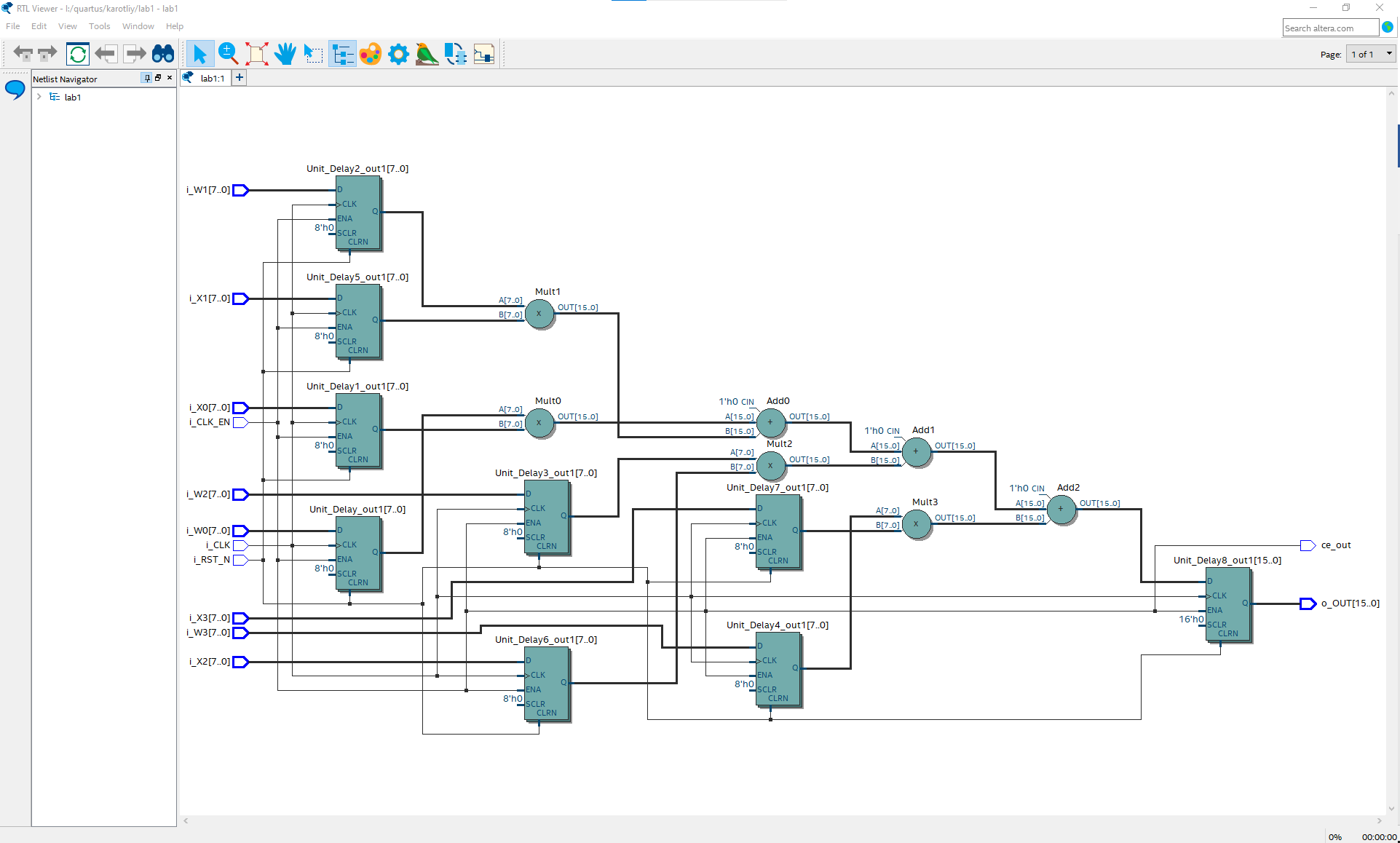
end

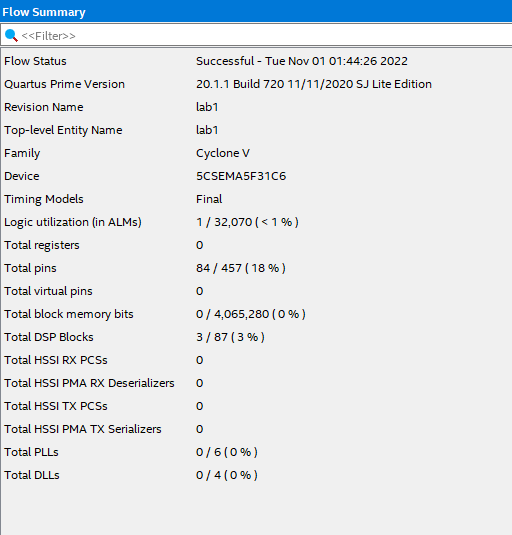
assign o\_OUT = Unit\_Delay8\_out1;

assign ce\_out = i\_CLK\_EN;

endmodule // Subsystem

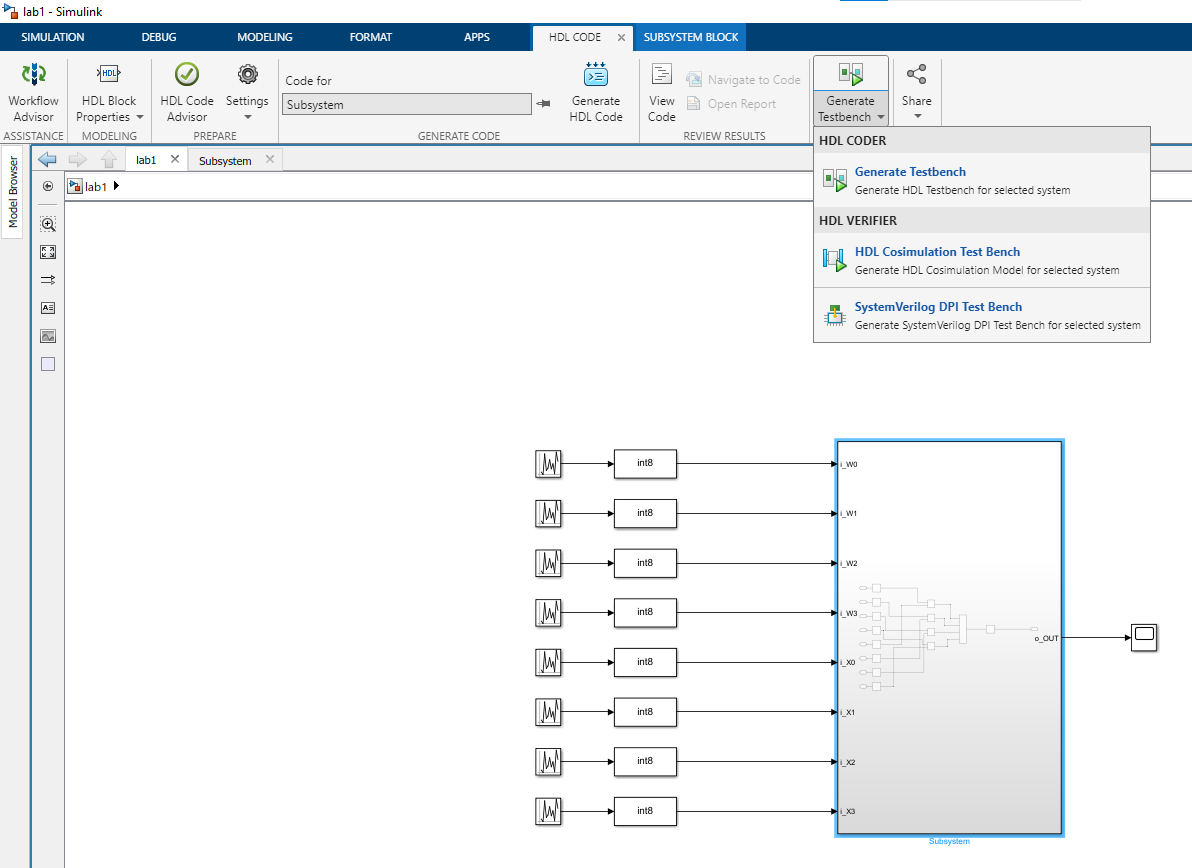
**4.Результат синтезу в RTL Viewer у квартусі та апаратні витрати плати:**



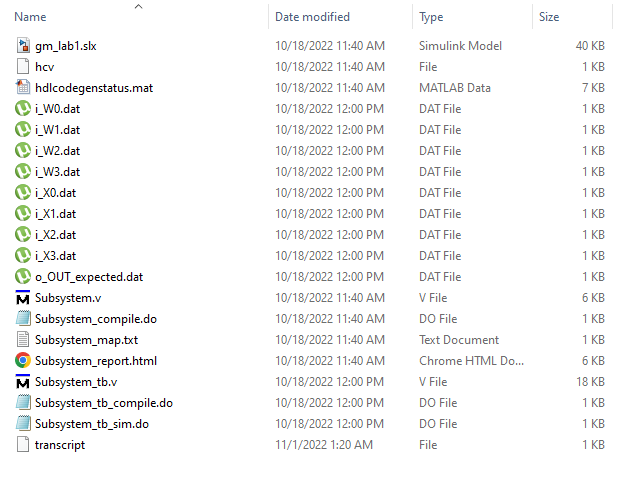
****

**5. Створити тестбенч в Matlab для створеної підсистеми і додати в звіт результат симуляції тестбенча в Modelsim/Questasim.**

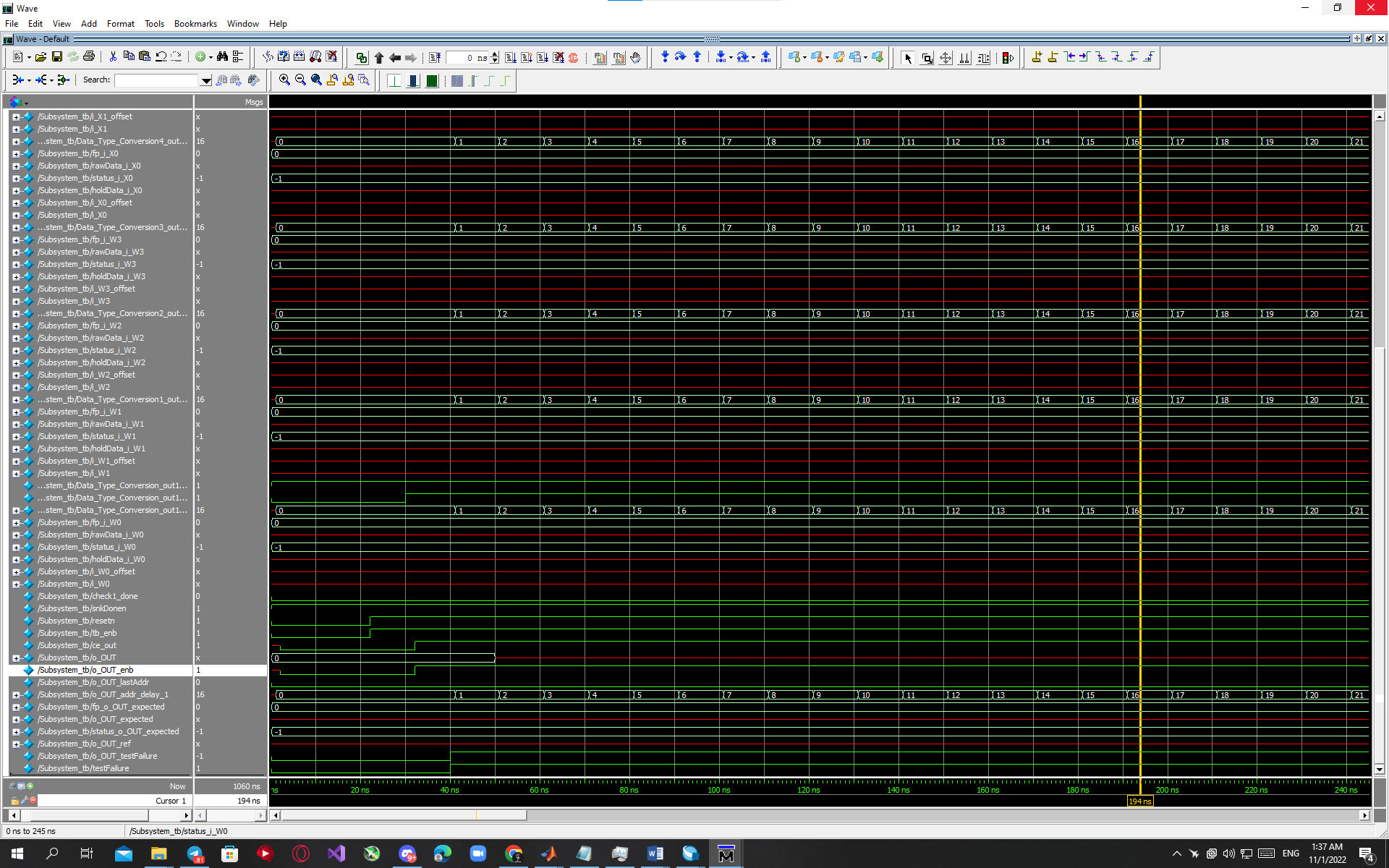
Створюємо тестбенч файл у сімулінку:

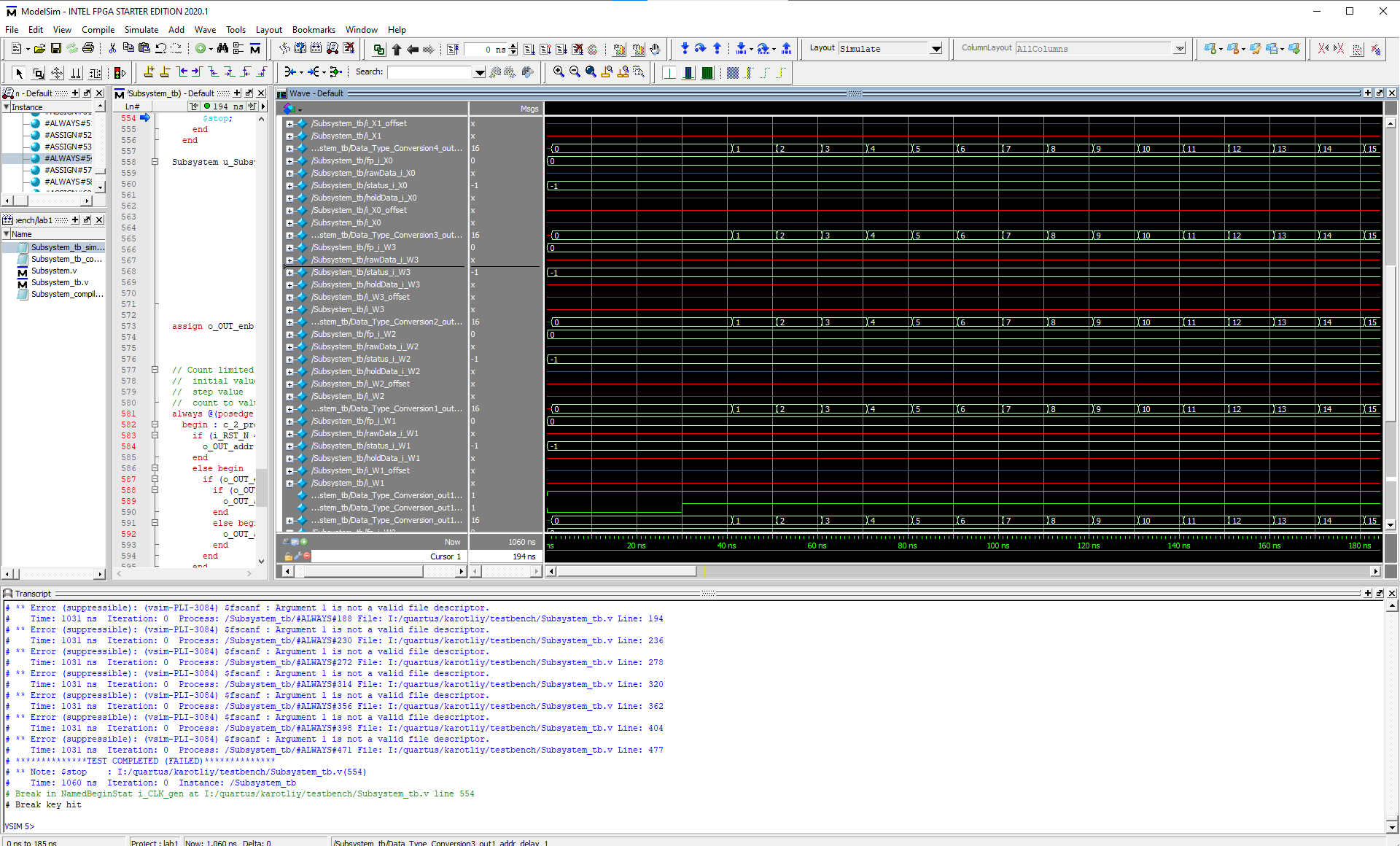
****

Бачимо створені файли:



Тестбенч файл запущений у ModelSim:

****



**Висновок:**

У ході виконання даної лабораторної робити ознайомилися з стандартними налаштування матлабу та сімулінку, навчилися складати та моделювати прості логічні схеми. Реалізували поставлене завдання створення підсистеми, подивилися коректність результатів у логічному аналізаторі. Створили тестбенч файл та відкрили його у моделсім. У квартусі синтезували схематичне позначення елементів по раніше згенерованому верілог коду підсистеми та скільки використали ресурсів плати.