

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 III-го
курсу ФЕЛ
група ДК-02
Овдієнко П.К.
25.10.2022

Київ-2022

Хід роботи

1. В Simulink реалізувати підсистему, що розраховує функцію:

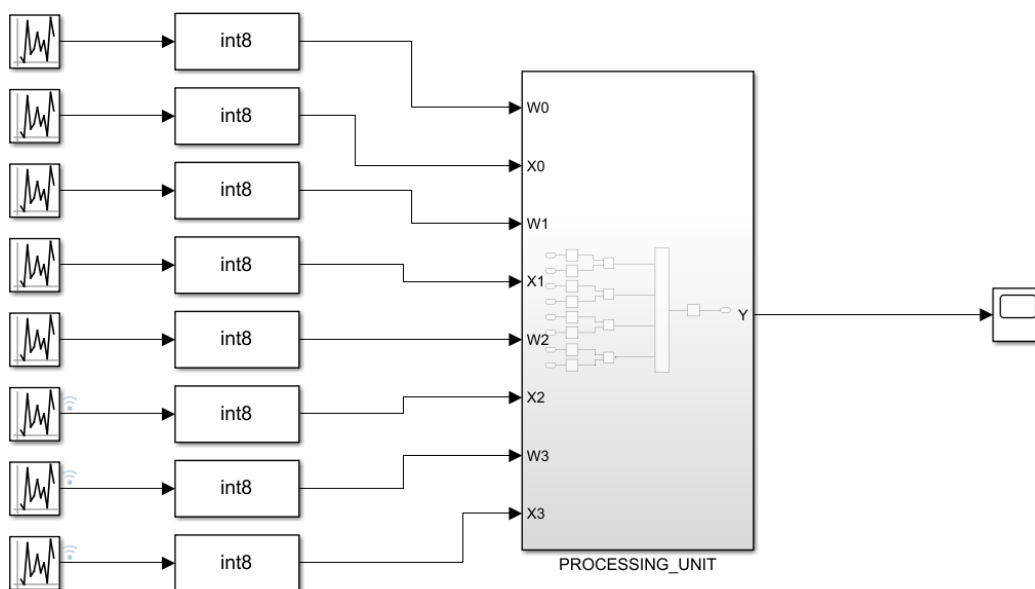
$$Y = W0*X0 + W1*X1 + W2*X2 + W*X3$$

Типи даних входів: int8

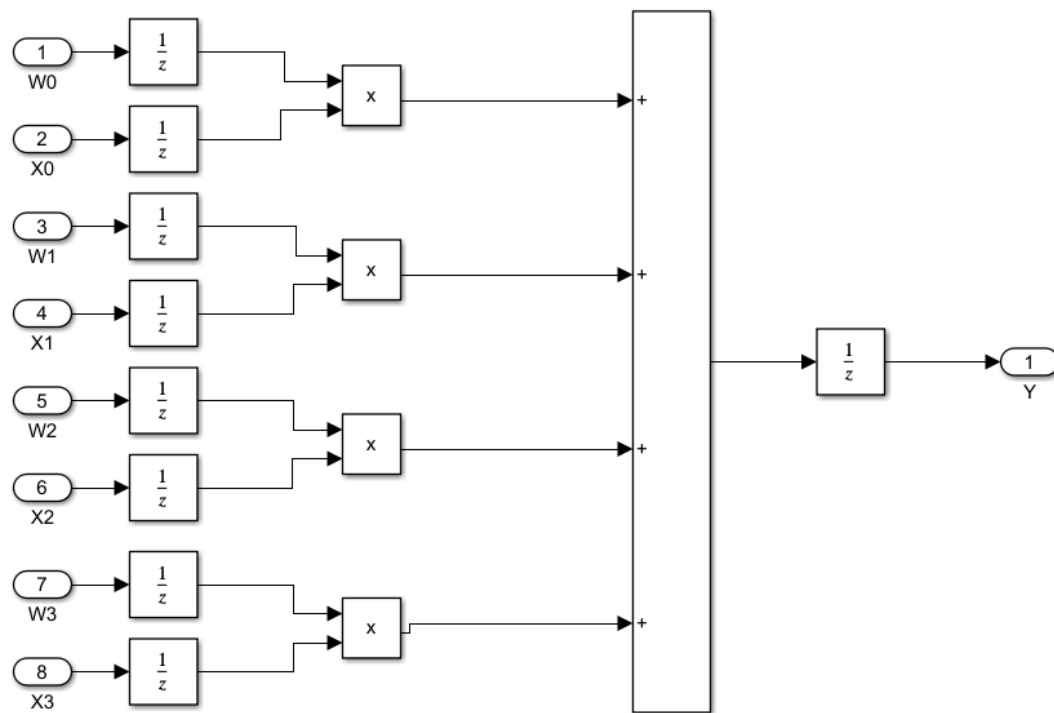
Тип даних виходу: int16

На входах і виході поставити регістри (блок затримки на 1 такт)

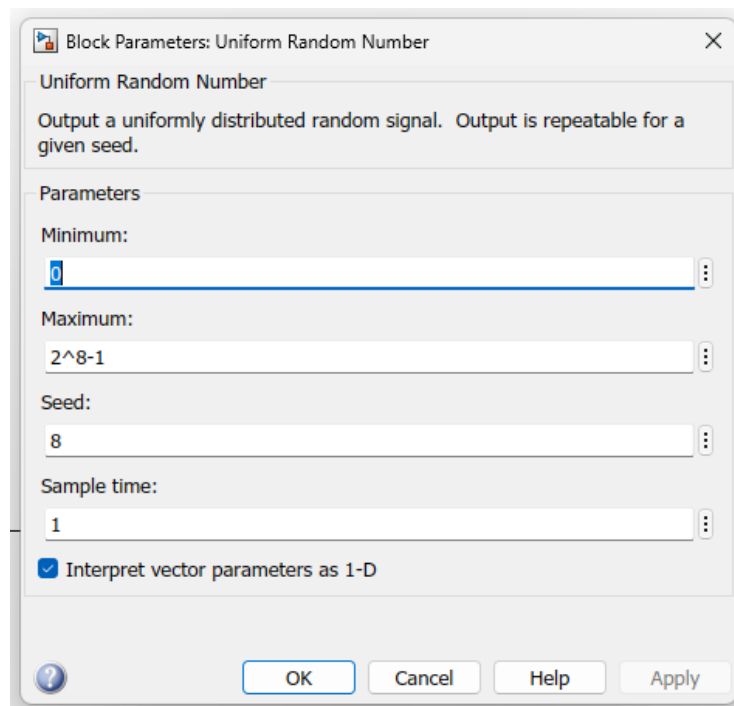
Схема має наступний вигляд:



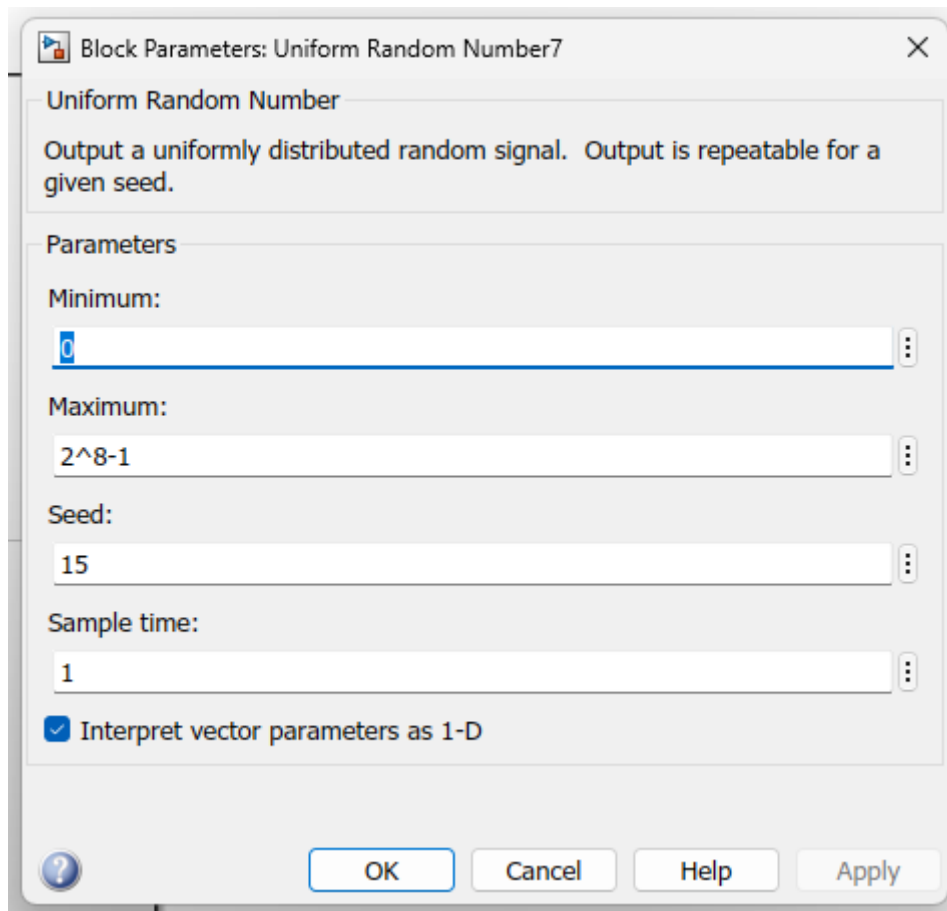
Вигляд всередині блоку PROCESSING_UNIT:



Налаштування першого Uniform Random number:



Налаштування останнього Uniform Random number:



Як можна побачити параметр seed починається з 8 (номер за списком) і далі збільшується з кількістю uniform random number. Всі інші налаштування були задані на основі параметрів які вказані в методичці і їх можна буде продивитися в надісланому проєкті.

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

Результат виглядає наступним чином:

lab1 - Logic Analyzer

LOGIC ANALYZER

TRIGGER

Add Divider

Add Group

Add Cursor

Previous Transition

Next Transition

Lock

Delete

Zoom & Pan

Stepping Options

Run

Step Forward

Stop

Find

Settings

EDIT

CURSORS

ZOOM & PAN

SIMULATE

FIND

GLOBAL

► PROCESSING_UNIT/W0	98	26	-32	-52	63	119	-85
► PROCESSING_UNIT/X0	-114	-99	38	39	-38	-26	97
► PROCESSING_UNIT/W1	-70	32	-102	-128	-14	85	23
► PROCESSING_UNIT/X1	-25	-93	16	-39	11	-60	-52
► PROCESSING_UNIT/W2	20	39	-122	52	35	52	-126
► PROCESSING_UNIT/X2	64	-88	-5	-115	60	-94	56
► PROCESSING_UNIT/W3	108	46	113	-26	84	18	-20
► PROCESSING_UNIT/X3	-104	-80	-25	65	108	-128	-94
► PROCESSING_UNIT/Unit_Delay8	-6336	0	-12584	-9799	-4706	13412	

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

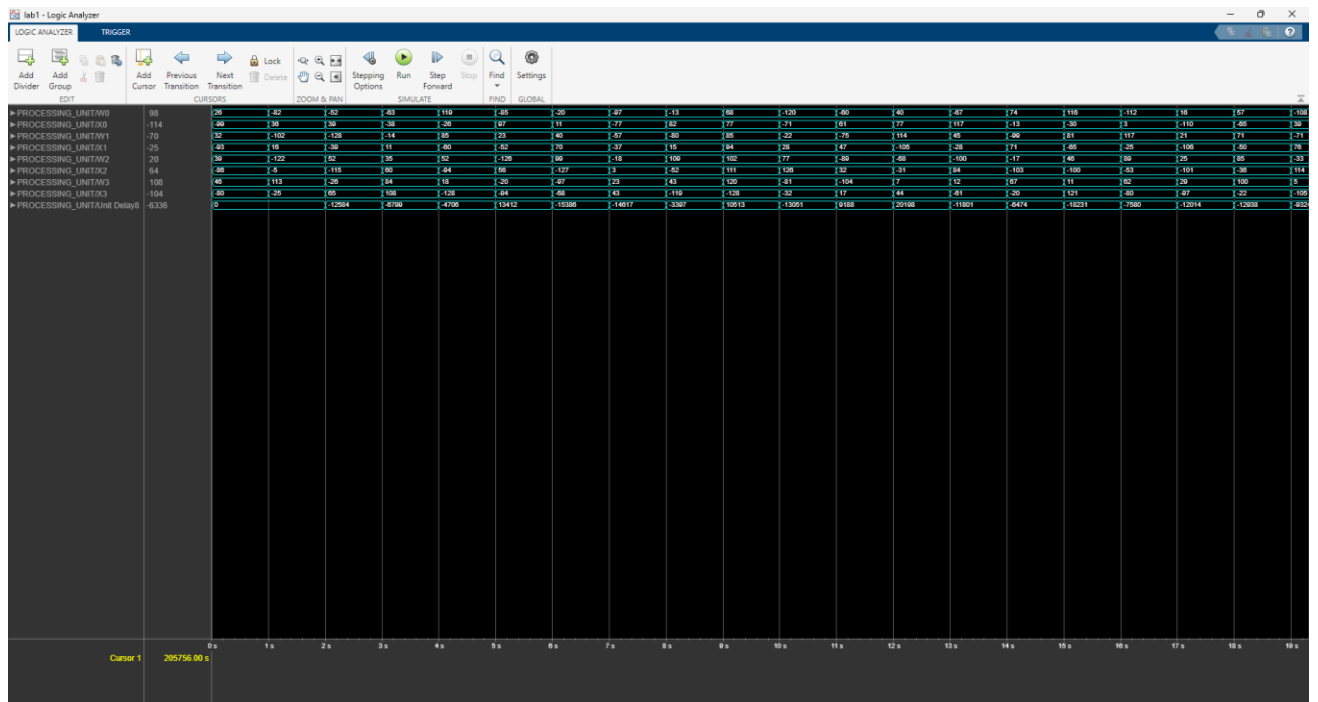
+

+

+

+

+</



Зробимо перевірку:

Візьмемо третій такт:

$(-52) * 39 + (-128) * (-39) + 52 * (-115) + (-26) * 65 = -4706$ і так як через затримка в два такти то на виході через два такти ми і можемо побачити це число -4706.

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:

```

lab1.v
2
3
4 File Name: hdlsrc\lab1\Subsystem.v
5 Created: 2022-10-18 11:40:18
6
7 Generated by MATLAB 9.12 and HDL Coder 3.20
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54

```

```

-- 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
Y 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,
    w0,
    x0,
    w1,
    x1,
    w2,
    x2,
    w3,
    x3,
    ce_out,
    y);

```

```

lab1.v
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105

```

```

ce_out,
y);

input i_CLK;
input i_RST_N;
input i_CLK_EN;
input signed [7:0] w0; // int8
input signed [7:0] x0; // int8
input signed [7:0] w1; // int8
input signed [7:0] x1; // int8
input signed [7:0] w2; // int8
input signed [7:0] x2; // int8
input signed [7:0] w3; // int8
input signed [7:0] x3; // int8
output ce_out;
output signed [15:0] y; // int16

wire enb;
reg signed [7:0] unit_delay_out1; // int8
reg signed [7:0] unit_delay2_out1; // int8
reg signed [7:0] unit_delay4_out1; // int8
reg signed [7:0] unit_delay6_out1; // int8
reg signed [7:0] unit_delay1_out1; // int8
wire signed [15:0] Product_out1; // int16
reg signed [7:0] unit_delay3_out1; // int8
wire signed [15:0] Product1_out1; // int16
wire signed [15:0] Add_stage2_add_temp; // sfixed16
wire signed [16:0] Add_op_stage1; // sfixed17
reg signed [7:0] unit_delay5_out1; // int8
wire signed [15:0] Product2_out1; // int16
wire signed [15:0] Add_stage3_add_cast; // sfixed16
wire signed [15:0] Add_stage3_add_temp; // sfixed16
wire signed [17:0] Add_op_stage2; // sfixed18
reg signed [7:0] unit_delay7_out1; // int8
wire signed [15:0] Product3_out1; // int16
wire signed [15:0] Add_stage4_add_cast; // sfixed16
wire signed [15:0] Add_out1; // int16
reg signed [15:0] unit_delay8_out1; // int16

assign enb = i_CLK_EN;

always @(posedge i_CLK or negedge i_RST_N)
begin : Unit_Delay_process
    if (i_RST_N == 1'b0) begin
        unit_delay_out1 <= 8'b00000000;
    end
    else begin
        if (enb) begin
            unit_delay_out1 <= w0;
        end
    end
end

```

```

lab1.v
103     unit_Delay_out1 <= w0;
104     end
105   end
106 end
107
108
109
110 always @(posedge i_CLK or negedge i_RST_N)
111   begin : unit_Delay2_process
112     if (i_RST_N == 1'b0) begin
113       unit_Delay2_out1 <= 8'sb000000000;
114     end
115     else begin
116       if (enb) begin
117         unit_Delay2_out1 <= w1;
118       end
119     end
120   end
121
122
123
124 always @(posedge i_CLK or negedge i_RST_N)
125   begin : unit_Delay4_process
126     if (i_RST_N == 1'b0) begin
127       unit_Delay4_out1 <= 8'sb000000000;
128     end
129     else begin
130       if (enb) begin
131         unit_Delay4_out1 <= w2;
132       end
133     end
134   end
135
136
137
138 always @(posedge i_CLK or negedge i_RST_N)
139   begin : unit_Delay6_process
140     if (i_RST_N == 1'b0) begin
141       unit_Delay6_out1 <= 8'sb000000000;
142     end
143     else begin
144       if (enb) begin
145         unit_Delay6_out1 <= w3;
146       end
147     end
148   end
149
150
151
152 always @(posedge i_CLK or negedge i_RST_N)
153   begin : unit_Delay1_process
154     if (i_RST_N == 1'b0) begin
155       unit_Delay1_out1 <= 8'sb000000000;
156     end

```

```

lab1.v
154     if (i_RST_N == 1'b0) begin
155       unit_Delay1_out1 <= 8'sb000000000;
156     end
157     else begin
158       if (enb) begin
159         unit_Delay1_out1 <= x0;
160       end
161     end
162   end
163
164
165
166 assign Product_out1 = Unit_Delay_out1 * Unit_Delay1_out1;
167
168
169
170 always @(posedge i_CLK or negedge i_RST_N)
171   begin : unit_Delay3_process
172     if (i_RST_N == 1'b0) begin
173       unit_Delay3_out1 <= 8'sb000000000;
174     end
175     else begin
176       if (enb) begin
177         unit_Delay3_out1 <= x1;
178       end
179     end
180   end
181
182
183
184 assign Product1_out1 = Unit_Delay2_out1 * Unit_Delay3_out1;
185
186
187 assign Add_stage2_add_temp = Product_out1 + Product1_out1;
188 assign Add_op_stage1 = {Add_stage2_add_temp[15], Add_stage2_add_temp};
189
190
191
192
193 always @(posedge i_CLK or negedge i_RST_N)
194   begin : unit_Delay5_process
195     if (i_RST_N == 1'b0) begin
196       unit_Delay5_out1 <= 8'sb000000000;
197     end
198     else begin
199       if (enb) begin
200         unit_Delay5_out1 <= x2;
201       end
202     end
203   end
204
205
206
207 assign Product2_out1 = Unit_Delay4_out1 * Unit_Delay5_out1;

```

```

206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259

assign Product2_out1 = Unit_Delay4_out1 * Unit_Delay5_out1;

assign Add_stage3_add_cast = Add_op_stage1[15:0];
assign Add_stage3_add_temp = Add_stage3_add_cast + Product2_out1;
assign Add_op_stage2 = {{2{Add_stage3_add_temp[15]}}, Add_stage3_add_temp};

always @(posedge i_CLK or negedge i_RST_N)
begin : Unit_Delay7_process
if (i_RST_N == 1'b0) begin
Unit_Delay7_out1 <= 8'sb000000000;
end
else begin
if (enb) begin
Unit_Delay7_out1 <= X3;
end
end
end

assign Product3_out1 = Unit_Delay6_out1 * Unit_Delay7_out1;

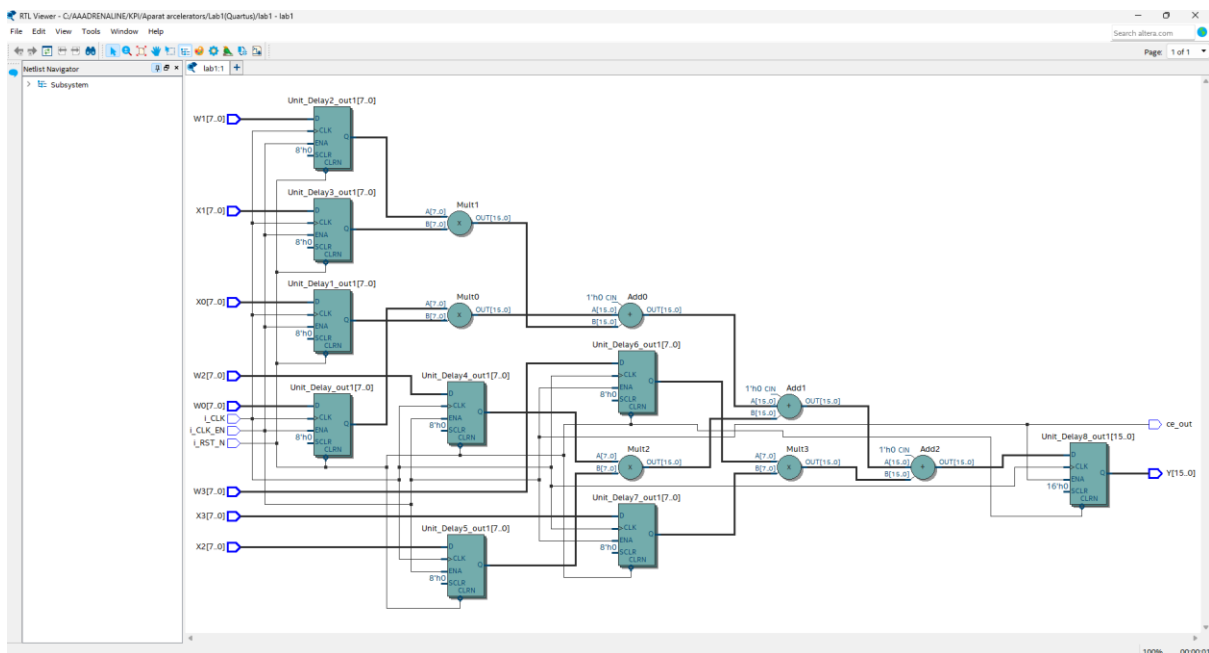
assign Add_stage4_add_cast = Add_op_stage2[15:0];
assign Add_out1 = Add_stage4_add_cast + Product3_out1;

always @(posedge i_CLK or negedge i_RST_N)
begin : Unit_Delay8_process
if (i_RST_N == 1'b0) begin
Unit_Delay8_out1 <= 16'sb0000000000000000;
end
else begin
if (enb) begin
Unit_Delay8_out1 <= Add_out1;
end
end
end

assign Y = Unit_Delay8_out1;
assign ce_out = i_CLK_EN;
endmodule // Subsystem

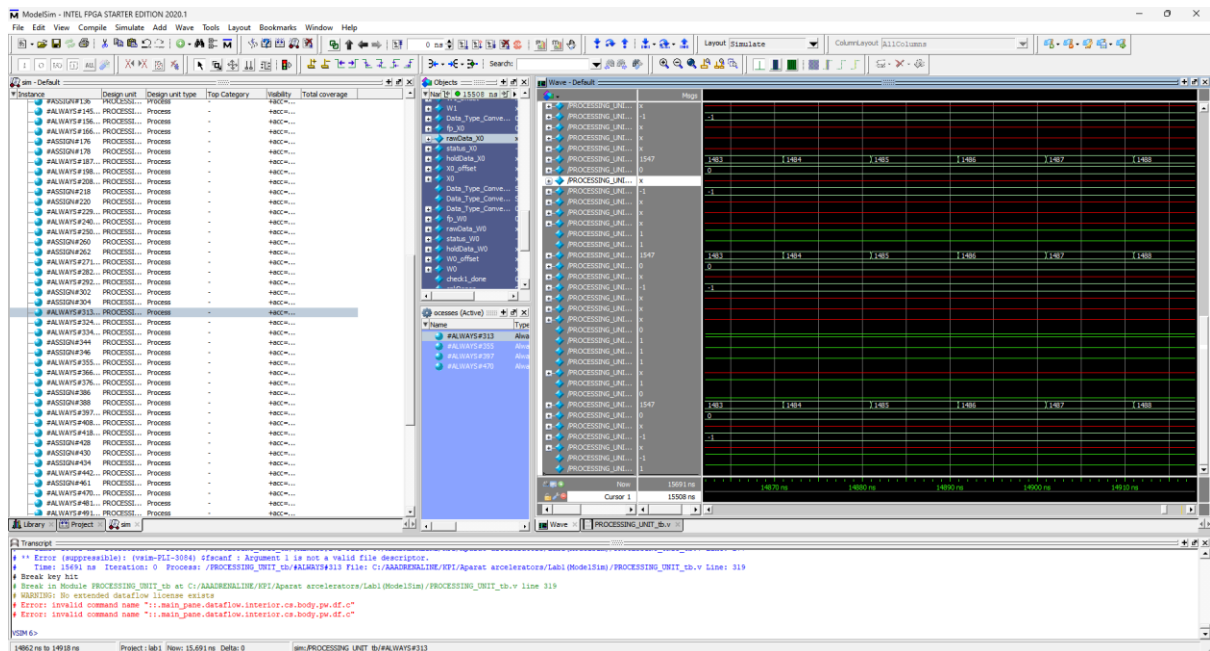
```

Результат синтезу в RTL Viewer:



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

Результат симуляції створеного тестбенчу:



Висновок: під час виконання лабораторної роботи завдяки відео зі знайомством в середовищі Matlab вдалося розробити підсистему і перевірити її роботу. Можна сказати, що симуляція в Matlab та синтез в Quartus Prime мають задовільний результат, у той час як симуляція в середовищі ModelSim має сумніви та невпевненість, що може бути пов'язано з певною при налаштуванні та симуляції проекту.