

Міністерство освіти і науки України
Національний технічний університет України
«Київський політехнічний інститут»

Кафедра КЕОА

Лабораторна робота №2
з курсу: «Апаратні прискорювачі обчислень на мікросхемах
програмованої логіки»

Виконав:
студент III-го
курсу ФЕЛ
група ДК-02
Овдієнко П.К.
13.01.2023

Київ-2023

Мета: в Simulink реалізувати підсистему, що розраховує модуль і аргумент комплексного числа для вхідних даних у форматах з фіксованою комою і плаваючою комою

Хід роботи

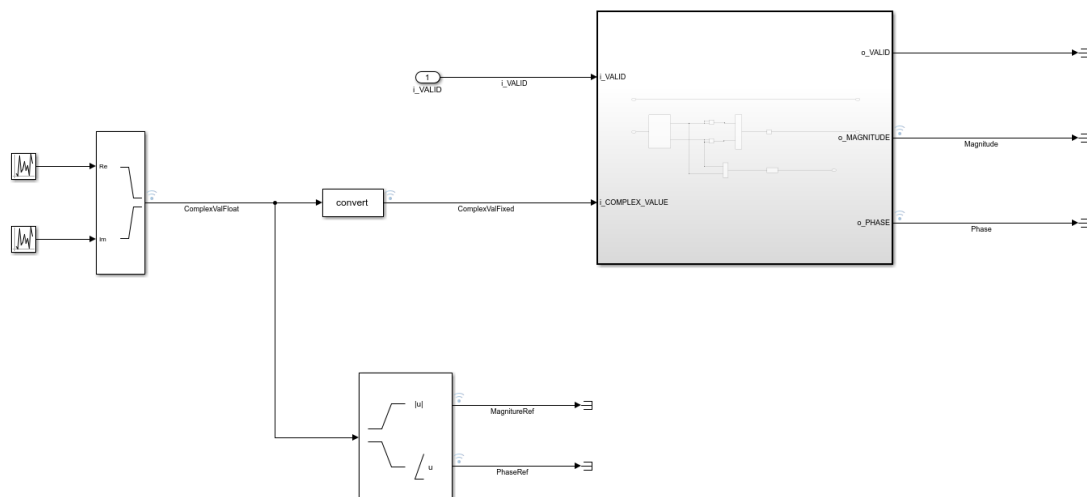
Варіант

Овдієнко Павло

23

1. В Simulink побудувати блок схеми обчислювачів модуля і аргументу комплексного числа для вхідного аргументу з фіксованої комою і плаваючою комою. Обчислювачі для вхідних даних з фіксованою комою і плаваючою комою будувати в окремих моделях Simulink.

Вигляд схеми з фіксованою комою:



Параметри “Uniform random number”:

Block Parameters: Uniform Random Number

Uniform Random Number

Output a uniformly distributed random signal. Output is repeatable for a given seed.

Parameters

Minimum:

Maximum:

Seed:

Sample time:

☒ Interpret vector parameters as 1-D

OK Cancel Help Apply

Block Parameters: Uniform Random Number1

Uniform Random Number

Output a uniformly distributed random signal. Output is repeatable for a given seed.

Parameters

Minimum:

Maximum:

Seed:

Sample time:

☒ Interpret vector parameters as 1-D

OK Cancel Help Apply

Параметр seed налаштований відповідно до номеру варіанту (для другого генератора випадкових чисел $23+1=24$).

Значення мінімуму, максимуму були обрані наступним чином:

Відомо з лекції що, ціла частина знакового числа Fixed Point буде знаковою і приймає значення з діапазону: $-2^{(N-1)} \dots 2^{(N-1)}-1$.

Дробова частина завжди додатня і приймає значення з діапазону $0 \dots 1-2^{(-M)}$.

Розрядність цілої частини $N=K=23$ (номер варіанту), розрядність дробової частини $M=32-23=9$. (Вираз **N-1** використовується тому що в цілій частині в знакових числах виділяємо найстарший біт під знак.)

Розрахуємо мінімальне та максимальне значення:

Мінімальне (при цілій частині $-2^{(N-1)}$ та дробовій частині 0) :

$$-2^{(N-1)}+0 = \underline{-2^{(23-1)}} = -2^{22} = -4194304$$

Максимальне (ціла частина $2^{(N-1)}-1$, дробова $1-2^{(-M)}$) :

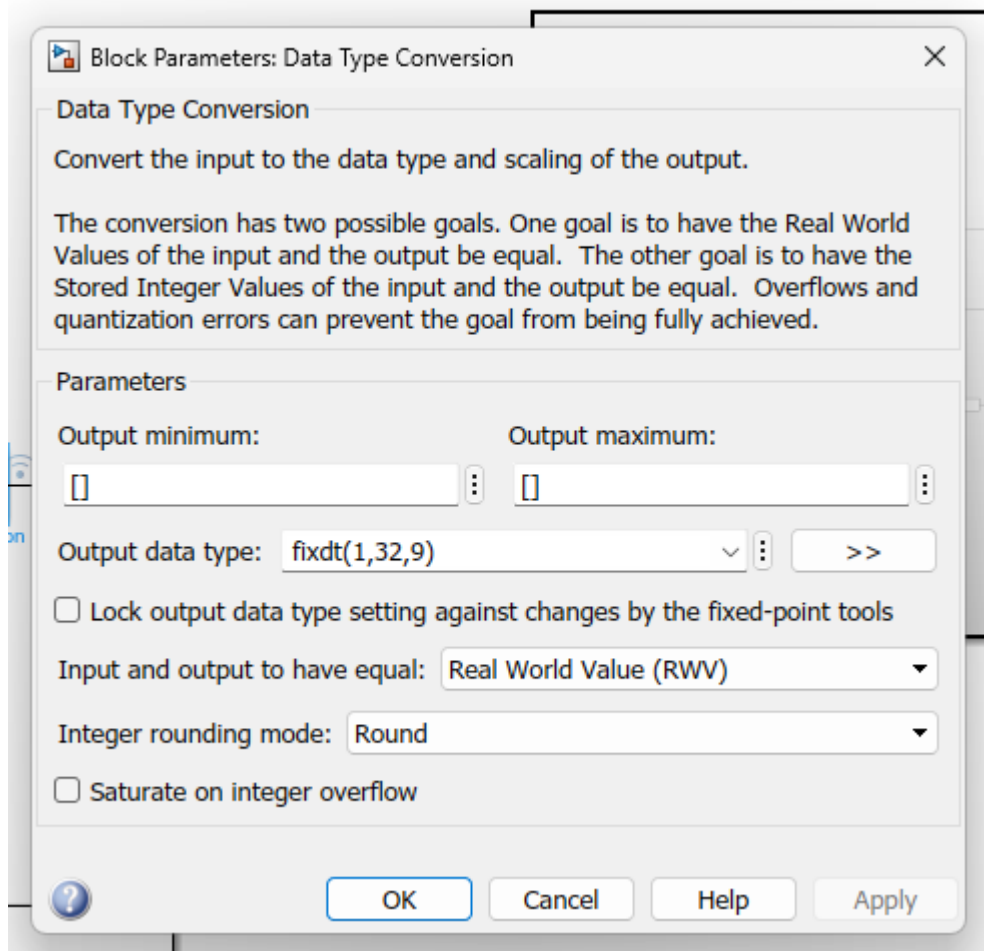
$$2^{(N-1)}-1+1-2^{(-M)} = 2^{(N-1)}-2^{(-M)} = \underline{2^{(23-1)}-2^{(-9)}} = \\ = 4194304 - (1/512) = 4194304 - 0,00195312.$$

Виділені значення і є тим що було підставлено в мінімум та максимум.

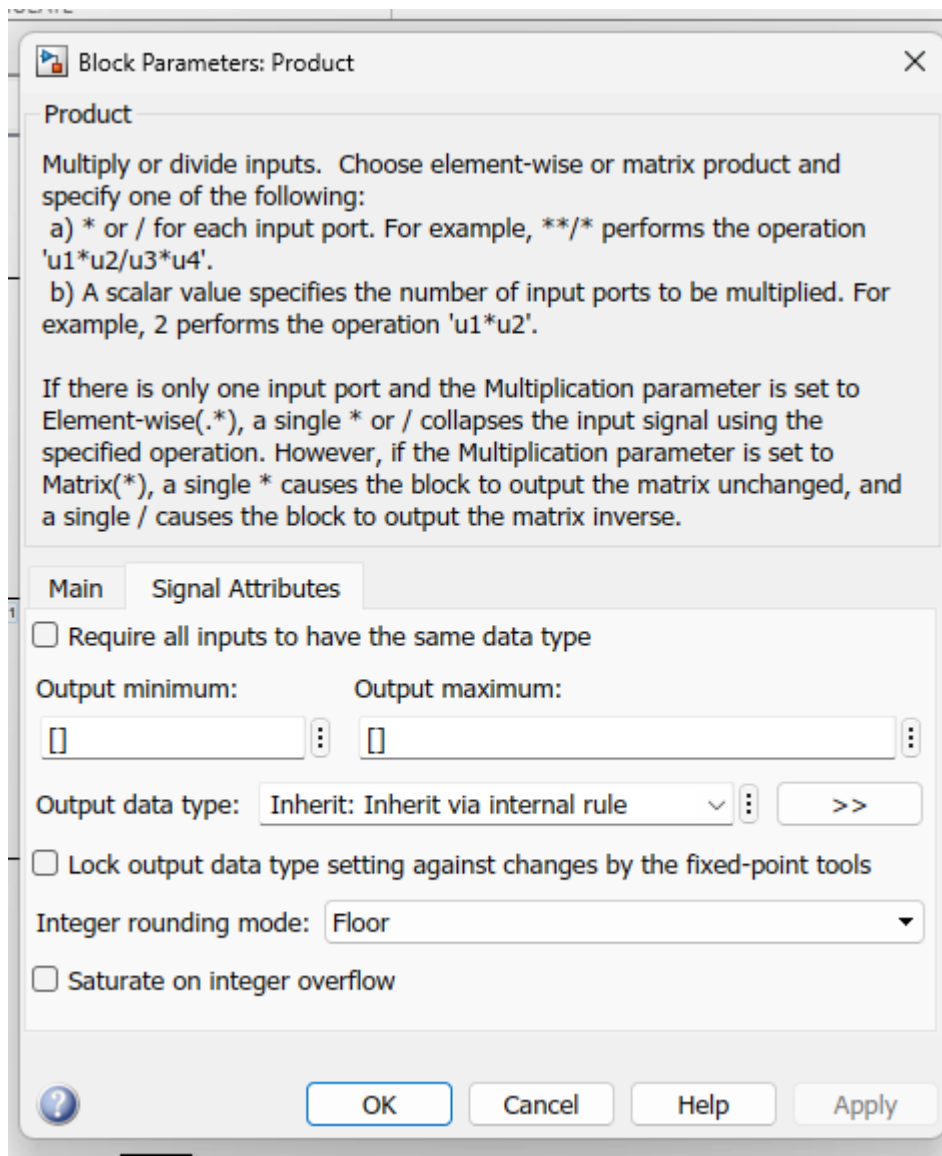
Інші налаштування:

Параметри блоку “data type conversion”:

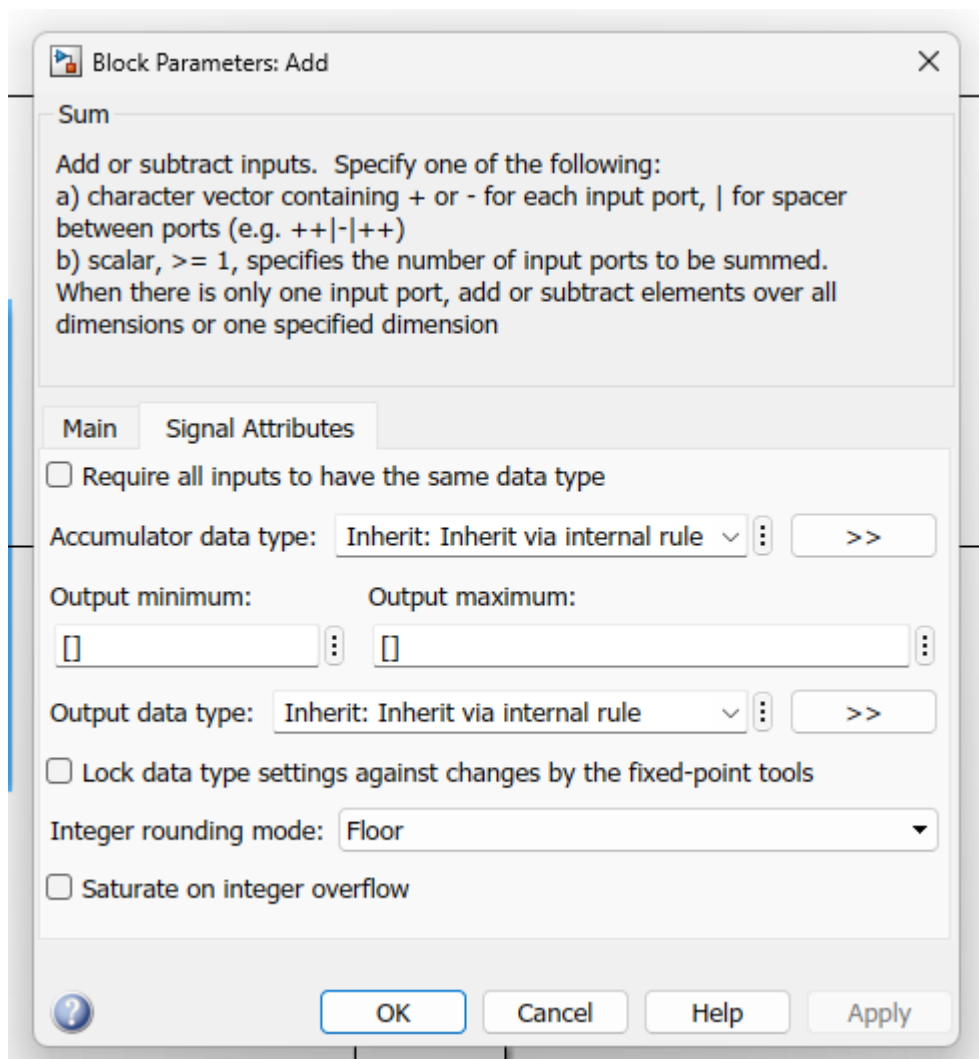




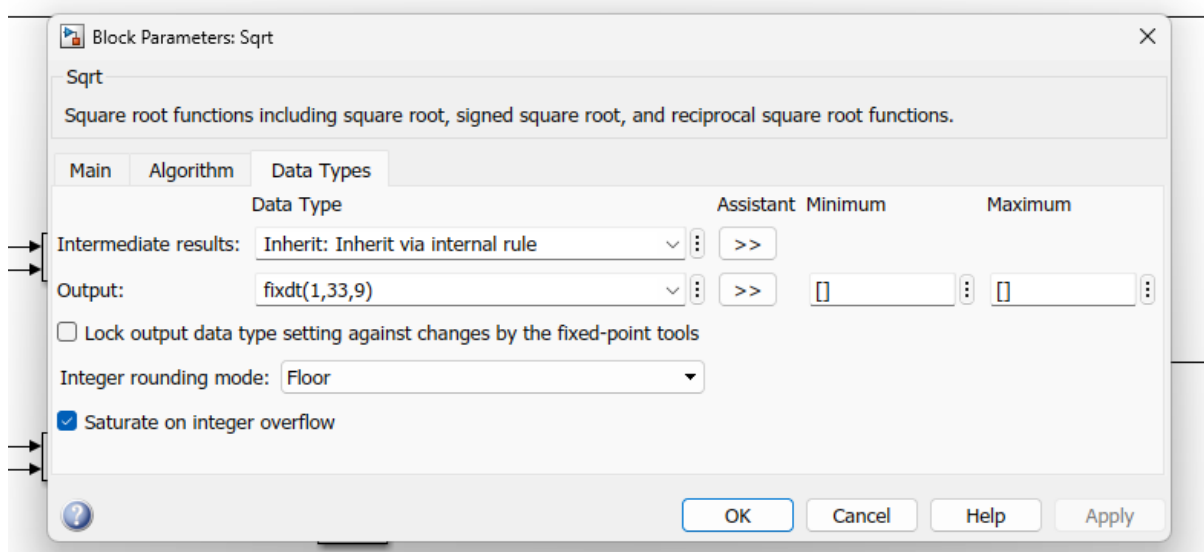
Параметри блоків “product” (налаштування за замовчуванням):



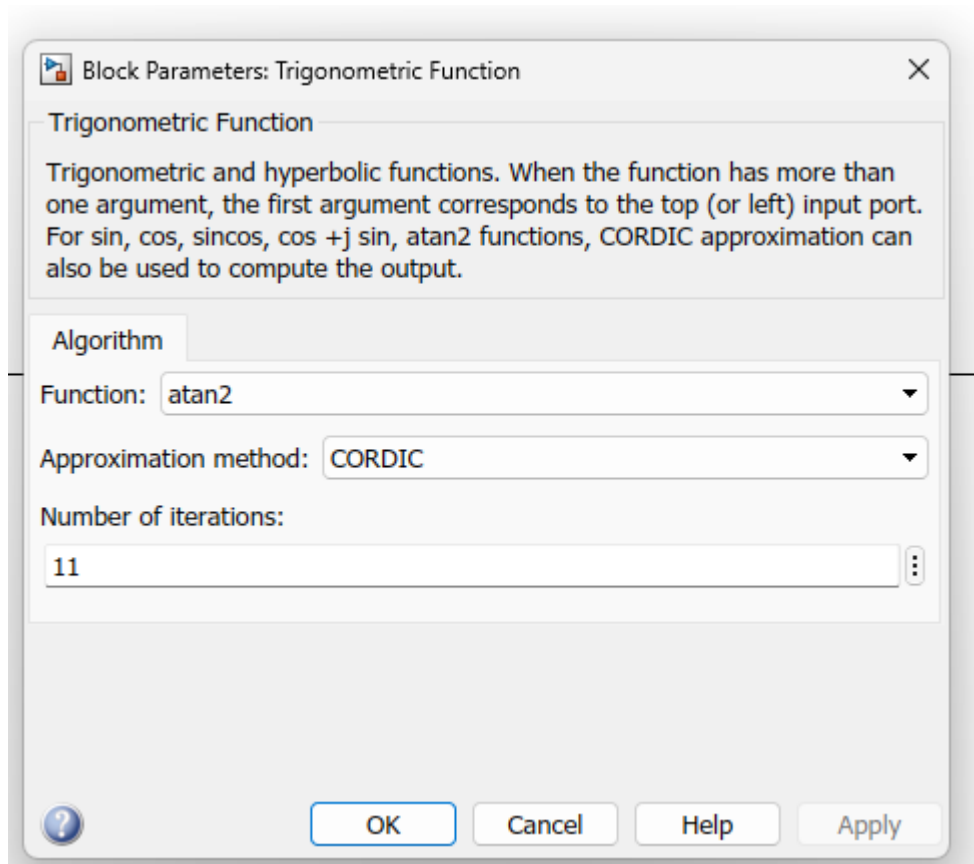
Параметри блоку “add” (налаштування теж за замовчуванням):




Параметри блоку “sqrt”:



Параметри блоку “Trigonometric Function”:



Параметри блоку “data type conversion” для виходу o_Phase:

Block Parameters: Data Type Conversion

×

Data Type Conversion

Convert the input to the data type and scaling of the output.

The conversion has two possible goals. One goal is to have the Real World Values of the input and the output be equal. The other goal is to have the Stored Integer Values of the input and the output be equal. Overflows and quantization errors can prevent the goal from being fully achieved.

Parameters

Output minimum:

Output maximum:

Output data type:

fixdt(1,16,13)

▼

⋮

>>

☐

Lock output data type setting against changes by the fixed-point tools

Input and output to have equal:

Real World Value (RWV)

▼

Integer rounding mode:

Floor

▼

☐

Saturate on integer overflow

?

OK

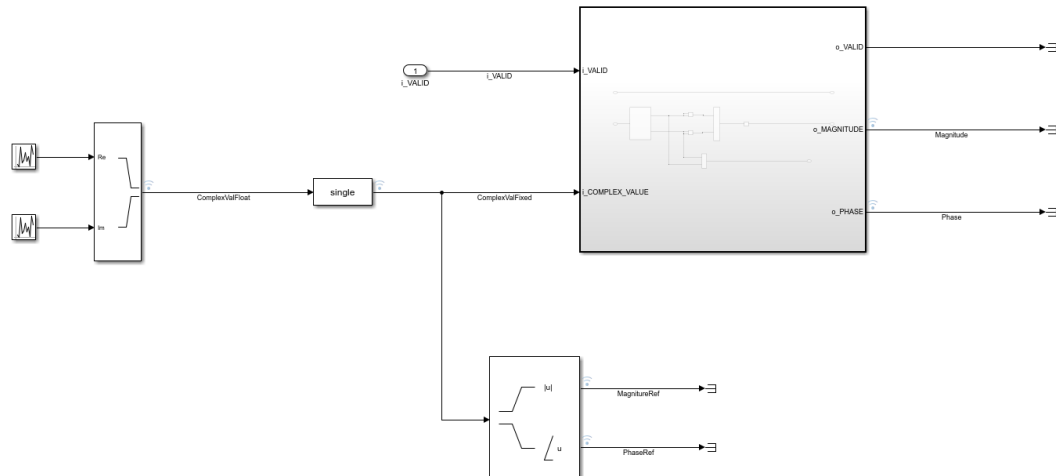
Cancel

Help

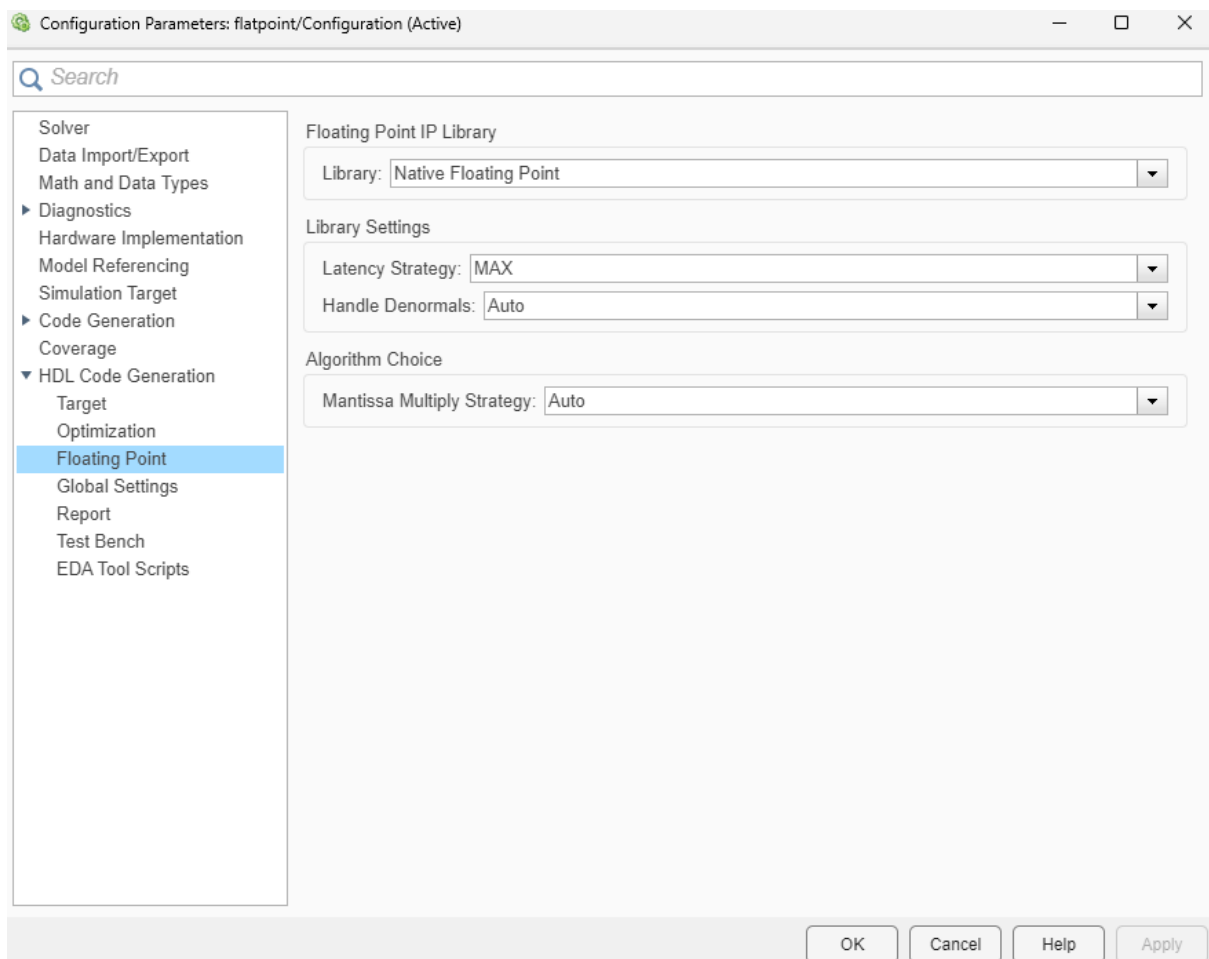
Apply

Зміни для схеми з плаваючою комою:

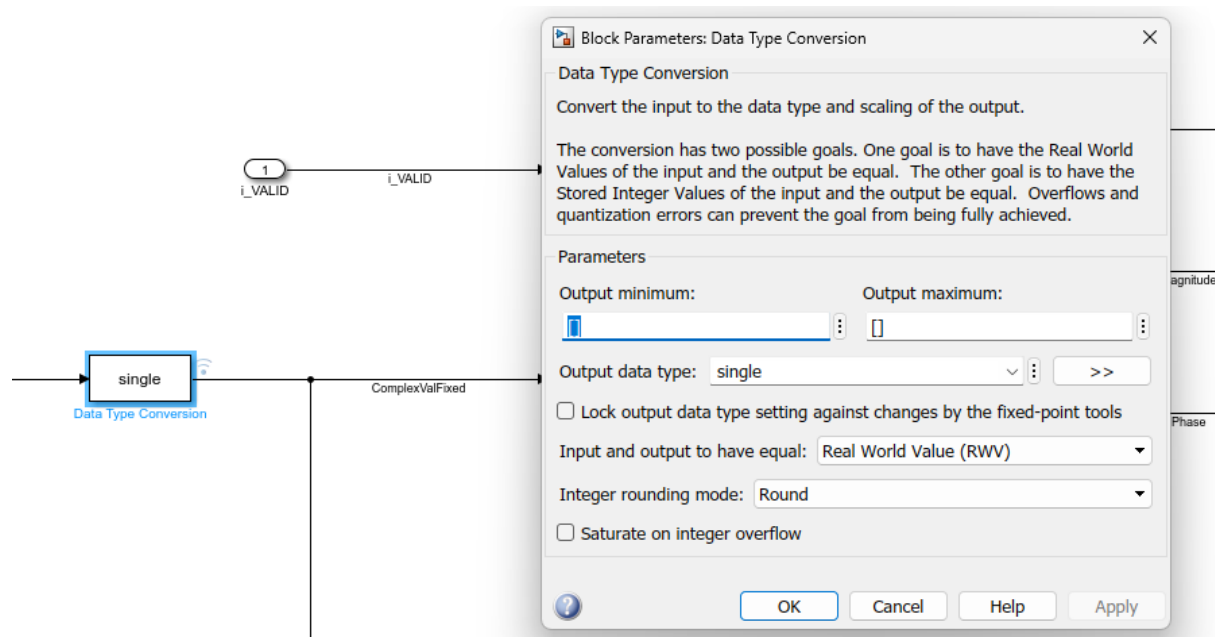
Вигляд схеми:



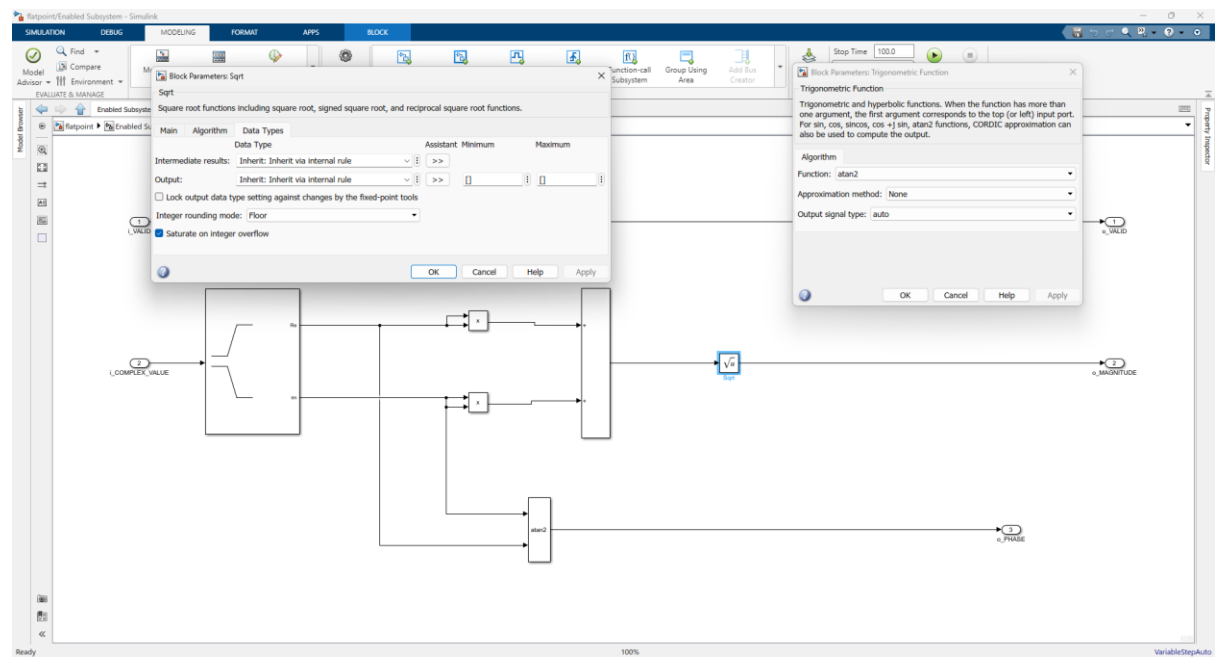
Налаштування Simulink:



Параметри блоку “data type conversion”:



Зміни в submodule:



Налаштування аналізатора:



Значення для схеми floatpoint:



Як можна побачити з графіків значення для обох схем, що в першому що в другому графіку, еталонні значення майже однакові з розрахованими значеннями і вони відрізняються на невеличку похибку.

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).

Для **FixedPoint**:

```

1 //-----
2 //
3 // File Name: hdlsrc\fixedpoint\Enabled_Subsystem.v
4 // Created: 2023-01-13 04:38:39
5 //
6 // Generated by MATLAB 9.12 and HDL Coder 3.20
7 //
8 //-----
9 //
10 // -- Rate and Clocking Details
11 //-----
12 // Model base rate: 1
13 // Target subsystem base rate: 1
14 //
15 //-----
16 // Clock Enable Sample Time
17 //-----
18 // ce_out      1
19 //-----
20 //
21 //-----
22 // Output Signal      Clock Enable Sample Time
23 //-----
24 // o_VALID            ce_out      1
25 // o_MAGNITUDE        ce_out      1
26 // o_PHASE            ce_out      1
27 //-----
28 //
29 //-----
30 //
31 //-----
32 //
33 // Module: Enabled_Subsystem
34 // Source Path: fixedpoint/Enabled_Subsystem
35 // Hierarchy Level: 0
36 //-----
37 //
38 //-----
39 //
40 // `timescale 1 ns / 1 ns
41 //
42 // module Enabled_Subsystem
43 // (clk,
44 //  reset,
45 //  clk_enable,

```

```

43         (clk,
44          reset,
45          clk_enable,
46          i_VALID,
47          i_COMPLEX_VALUE_re,
48          i_COMPLEX_VALUE_im,
49          ce_out,
50          o_VALID,
51          o_MAGNITUDE,
52          o_PHASE);
53
54
55     input    clk;
56     input    reset;
57     input    clk_enable;
58     input [63:0] i_VALID; // double
59     input signed [31:0] i_COMPLEX_VALUE_re; // sfix32_En9
60     input signed [31:0] i_COMPLEX_VALUE_im; // sfix32_En9
61     output    ce_out;
62     output [63:0] o_VALID; // double
63     output signed [32:0] o_MAGNITUDE; // sfix33_En9
64     output signed [15:0] o_PHASE; // sfix16_En13
65
66
67     wire enb;
68     reg [63:0] delayMatch_reg [0:33]; // ufix64 [34]
69     wire [63:0] delayMatch_reg_next [0:33]; // ufix64 [34]
70     wire [63:0] i_VALID_1; // ufix64
71     wire signed [63:0] Product_out1; // sfix64_En18
72     wire signed [63:0] Product1_out1; // sfix64_En18
73     wire signed [63:0] Add_out1; // sfix64_En18
74     wire signed [32:0] Sqrt_out1; // sfix33_En9
75     wire signed [31:0] Trigonometric_Function_out1; // sfix32_En29
76     wire signed [15:0] Data_Type_Conversion_out1; // sfix16_En13
77     reg signed [15:0] delayMatch1_reg [0:19]; // sfix16 [20]
78     wire signed [15:0] delayMatch1_reg_next [0:19]; // sfix16_En13 [20]
79     wire signed [15:0] Data_Type_Conversion_out1_1; // sfix16_En13
80
81
82     assign enb = clk_enable;
83
84     always @(posedge clk or posedge reset)
85     begin : delayMatch_process
86         if (reset == 1'b1) begin
87             delayMatch_reg[0] <= 64'h0000000000000000;

```

```

80
81
82
83     assign enb = clk_enable;
84
85     always @(posedge clk or posedge reset)
86     begin : delayMatch_process
87         if (reset == 1'b1) begin
88             delayMatch_reg[0] <= 64'h0000000000000000;
89             delayMatch_reg[1] <= 64'h0000000000000000;
90             delayMatch_reg[2] <= 64'h0000000000000000;
91             delayMatch_reg[3] <= 64'h0000000000000000;
92             delayMatch_reg[4] <= 64'h0000000000000000;
93             delayMatch_reg[5] <= 64'h0000000000000000;
94             delayMatch_reg[6] <= 64'h0000000000000000;
95             delayMatch_reg[7] <= 64'h0000000000000000;
96             delayMatch_reg[8] <= 64'h0000000000000000;
97             delayMatch_reg[9] <= 64'h0000000000000000;
98             delayMatch_reg[10] <= 64'h0000000000000000;
99             delayMatch_reg[11] <= 64'h0000000000000000;
100            delayMatch_reg[12] <= 64'h0000000000000000;
101            delayMatch_reg[13] <= 64'h0000000000000000;
102            delayMatch_reg[14] <= 64'h0000000000000000;
103            delayMatch_reg[15] <= 64'h0000000000000000;
104            delayMatch_reg[16] <= 64'h0000000000000000;
105            delayMatch_reg[17] <= 64'h0000000000000000;
106            delayMatch_reg[18] <= 64'h0000000000000000;
107            delayMatch_reg[19] <= 64'h0000000000000000;
108            delayMatch_reg[20] <= 64'h0000000000000000;
109            delayMatch_reg[21] <= 64'h0000000000000000;
110            delayMatch_reg[22] <= 64'h0000000000000000;
111            delayMatch_reg[23] <= 64'h0000000000000000;
112            delayMatch_reg[24] <= 64'h0000000000000000;
113            delayMatch_reg[25] <= 64'h0000000000000000;
114            delayMatch_reg[26] <= 64'h0000000000000000;
115            delayMatch_reg[27] <= 64'h0000000000000000;
116            delayMatch_reg[28] <= 64'h0000000000000000;
117            delayMatch_reg[29] <= 64'h0000000000000000;
118            delayMatch_reg[30] <= 64'h0000000000000000;
119            delayMatch_reg[31] <= 64'h0000000000000000;
120            delayMatch_reg[32] <= 64'h0000000000000000;
121            delayMatch_reg[33] <= 64'h0000000000000000;
122        end
    else begin

```

```

121     end
122 else begin
123     if (enb) begin
124         delayMatch_reg[0] <= delayMatch_reg_next[0];
125         delayMatch_reg[1] <= delayMatch_reg_next[1];
126         delayMatch_reg[2] <= delayMatch_reg_next[2];
127         delayMatch_reg[3] <= delayMatch_reg_next[3];
128         delayMatch_reg[4] <= delayMatch_reg_next[4];
129         delayMatch_reg[5] <= delayMatch_reg_next[5];
130         delayMatch_reg[6] <= delayMatch_reg_next[6];
131         delayMatch_reg[7] <= delayMatch_reg_next[7];
132         delayMatch_reg[8] <= delayMatch_reg_next[8];
133         delayMatch_reg[9] <= delayMatch_reg_next[9];
134         delayMatch_reg[10] <= delayMatch_reg_next[10];
135         delayMatch_reg[11] <= delayMatch_reg_next[11];
136         delayMatch_reg[12] <= delayMatch_reg_next[12];
137         delayMatch_reg[13] <= delayMatch_reg_next[13];
138         delayMatch_reg[14] <= delayMatch_reg_next[14];
139         delayMatch_reg[15] <= delayMatch_reg_next[15];
140         delayMatch_reg[16] <= delayMatch_reg_next[16];
141         delayMatch_reg[17] <= delayMatch_reg_next[17];
142         delayMatch_reg[18] <= delayMatch_reg_next[18];
143         delayMatch_reg[19] <= delayMatch_reg_next[19];
144         delayMatch_reg[20] <= delayMatch_reg_next[20];
145         delayMatch_reg[21] <= delayMatch_reg_next[21];
146         delayMatch_reg[22] <= delayMatch_reg_next[22];
147         delayMatch_reg[23] <= delayMatch_reg_next[23];
148         delayMatch_reg[24] <= delayMatch_reg_next[24];
149         delayMatch_reg[25] <= delayMatch_reg_next[25];
150         delayMatch_reg[26] <= delayMatch_reg_next[26];
151         delayMatch_reg[27] <= delayMatch_reg_next[27];
152         delayMatch_reg[28] <= delayMatch_reg_next[28];
153         delayMatch_reg[29] <= delayMatch_reg_next[29];
154         delayMatch_reg[30] <= delayMatch_reg_next[30];
155         delayMatch_reg[31] <= delayMatch_reg_next[31];
156         delayMatch_reg[32] <= delayMatch_reg_next[32];
157         delayMatch_reg[33] <= delayMatch_reg_next[33];
158     end
159 end
160 end
161

```

```

161
162
163 assign i_VALID_1 = delayMatch_reg[33];
164 assign delayMatch_reg_next[0] = i_VALID;
165 assign delayMatch_reg_next[1] = delayMatch_reg[0];
166 assign delayMatch_reg_next[2] = delayMatch_reg[1];
167 assign delayMatch_reg_next[3] = delayMatch_reg[2];
168 assign delayMatch_reg_next[4] = delayMatch_reg[3];
169 assign delayMatch_reg_next[5] = delayMatch_reg[4];
170 assign delayMatch_reg_next[6] = delayMatch_reg[5];
171 assign delayMatch_reg_next[7] = delayMatch_reg[6];
172 assign delayMatch_reg_next[8] = delayMatch_reg[7];
173 assign delayMatch_reg_next[9] = delayMatch_reg[8];
174 assign delayMatch_reg_next[10] = delayMatch_reg[9];
175 assign delayMatch_reg_next[11] = delayMatch_reg[10];
176 assign delayMatch_reg_next[12] = delayMatch_reg[11];
177 assign delayMatch_reg_next[13] = delayMatch_reg[12];
178 assign delayMatch_reg_next[14] = delayMatch_reg[13];
179 assign delayMatch_reg_next[15] = delayMatch_reg[14];
180 assign delayMatch_reg_next[16] = delayMatch_reg[15];
181 assign delayMatch_reg_next[17] = delayMatch_reg[16];
182 assign delayMatch_reg_next[18] = delayMatch_reg[17];
183 assign delayMatch_reg_next[19] = delayMatch_reg[18];
184 assign delayMatch_reg_next[20] = delayMatch_reg[19];
185 assign delayMatch_reg_next[21] = delayMatch_reg[20];
186 assign delayMatch_reg_next[22] = delayMatch_reg[21];
187 assign delayMatch_reg_next[23] = delayMatch_reg[22];
188 assign delayMatch_reg_next[24] = delayMatch_reg[23];
189 assign delayMatch_reg_next[25] = delayMatch_reg[24];
190 assign delayMatch_reg_next[26] = delayMatch_reg[25];
191 assign delayMatch_reg_next[27] = delayMatch_reg[26];
192 assign delayMatch_reg_next[28] = delayMatch_reg[27];
193 assign delayMatch_reg_next[29] = delayMatch_reg[28];
194 assign delayMatch_reg_next[30] = delayMatch_reg[29];
195 assign delayMatch_reg_next[31] = delayMatch_reg[30];
196 assign delayMatch_reg_next[32] = delayMatch_reg[31];
197
198
199
200 assign Product_out1 = i_COMPLEX_VALUE_re * i_COMPLEX_VALUE_re;
201
202
203

```



```

199
200 assign Product_out1 = i_COMPLEX_VALUE_re * i_COMPLEX_VALUE_re;
201
202
203
204 assign Product1_out1 = i_COMPLEX_VALUE_im * i_COMPLEX_VALUE_im;
205
206
207
208 assign Add_out1 = Product_out1 + Product1_out1;
209
210
211
212 Sqrt u_Sqrt (.clk(clk),
213             .reset(reset),
214             .enb(clk_enable),
215             .din(Add_out1), // sfix64_En18
216             .dout(Sqrt_out1) // sfix33_En9
217             );
218
219 assign o_MAGNITUDE = Sqrt_out1;
220
221 atan2_cordic_nw u_Trigonometric_Function_inst (.clk(clk),
222                                                .reset(reset),
223                                                .enb(clk_enable),
224                                                .y_in(i_COMPLEX_VALUE_im), // sfix32_En9
225                                                .x_in(i_COMPLEX_VALUE_re), // sfix32_En9
226                                                .angle(Trigonometric_Function_out1) // sfix32_En29
227                                                );
228
229 assign Data_Type_Conversion_out1 = Trigonometric_Function_out1[31:16];
230
231
232

```

```

229 assign Data_Type_Conversion_out1 = Trigonometric_Function_out1[31:16];
230
231
232
233
234 always @(posedge clk or posedge reset)
235 begin : delayMatch1_process
236 if (reset == 1'b1) begin
237 delayMatch1_reg[0] <= 16'sb0000000000000000;
238 delayMatch1_reg[1] <= 16'sb0000000000000000;
239 delayMatch1_reg[2] <= 16'sb0000000000000000;
240 delayMatch1_reg[3] <= 16'sb0000000000000000;
241 delayMatch1_reg[4] <= 16'sb0000000000000000;
242 delayMatch1_reg[5] <= 16'sb0000000000000000;
243 delayMatch1_reg[6] <= 16'sb0000000000000000;
244 delayMatch1_reg[7] <= 16'sb0000000000000000;
245 delayMatch1_reg[8] <= 16'sb0000000000000000;
246 delayMatch1_reg[9] <= 16'sb0000000000000000;
247 delayMatch1_reg[10] <= 16'sb0000000000000000;
248 delayMatch1_reg[11] <= 16'sb0000000000000000;
249 delayMatch1_reg[12] <= 16'sb0000000000000000;
250 delayMatch1_reg[13] <= 16'sb0000000000000000;
251 delayMatch1_reg[14] <= 16'sb0000000000000000;
252 delayMatch1_reg[15] <= 16'sb0000000000000000;
253 delayMatch1_reg[16] <= 16'sb0000000000000000;
254 delayMatch1_reg[17] <= 16'sb0000000000000000;
255 delayMatch1_reg[18] <= 16'sb0000000000000000;
256 delayMatch1_reg[19] <= 16'sb0000000000000000;
257 end
258 else begin
259 if (enb) begin
260 delayMatch1_reg[0] <= delayMatch1_reg_next[0];
261 delayMatch1_reg[1] <= delayMatch1_reg_next[1];
262 delayMatch1_reg[2] <= delayMatch1_reg_next[2];
263 delayMatch1_reg[3] <= delayMatch1_reg_next[3];
264 delayMatch1_reg[4] <= delayMatch1_reg_next[4];
265 delayMatch1_reg[5] <= delayMatch1_reg_next[5];
266 delayMatch1_reg[6] <= delayMatch1_reg_next[6];
267 delayMatch1_reg[7] <= delayMatch1_reg_next[7];
268 delayMatch1_reg[8] <= delayMatch1_reg_next[8];
269 delayMatch1_reg[9] <= delayMatch1_reg_next[9];
270 delayMatch1_reg[10] <= delayMatch1_reg_next[10];
271 delayMatch1_reg[11] <= delayMatch1_reg_next[11];
272 delayMatch1_reg[12] <= delayMatch1_reg_next[12];
273 delayMatch1_reg[13] <= delayMatch1_reg_next[13];
274 delayMatch1_reg[14] <= delayMatch1_reg_next[14];

```

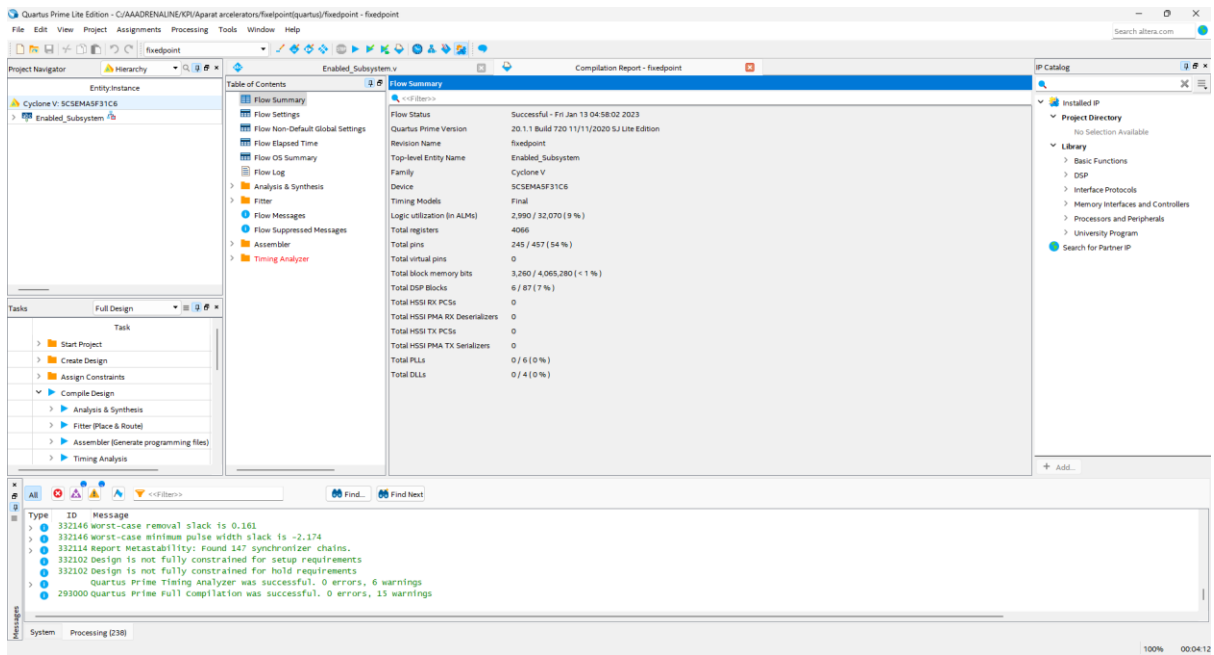


```

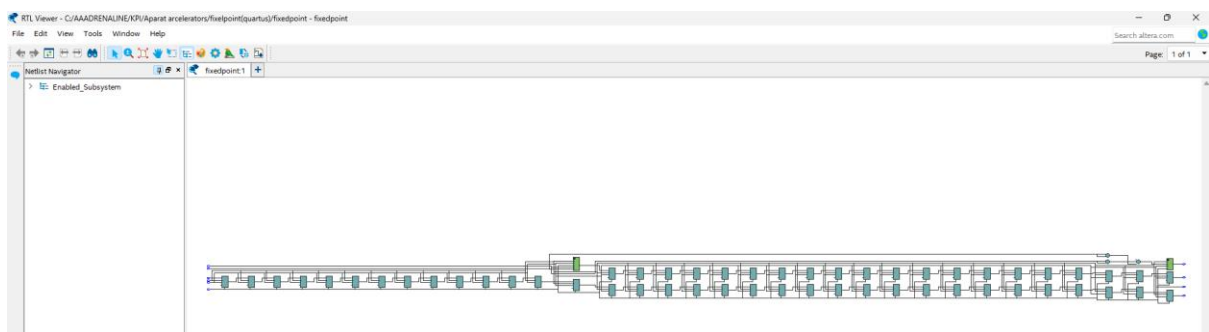
256     end
257   else begin
258     if (enb) begin
259       delayMatch1_reg[0] <= delayMatch1_reg_next[0];
260       delayMatch1_reg[1] <= delayMatch1_reg_next[1];
261       delayMatch1_reg[2] <= delayMatch1_reg_next[2];
262       delayMatch1_reg[3] <= delayMatch1_reg_next[3];
263       delayMatch1_reg[4] <= delayMatch1_reg_next[4];
264       delayMatch1_reg[5] <= delayMatch1_reg_next[5];
265       delayMatch1_reg[6] <= delayMatch1_reg_next[6];
266       delayMatch1_reg[7] <= delayMatch1_reg_next[7];
267       delayMatch1_reg[8] <= delayMatch1_reg_next[8];
268       delayMatch1_reg[9] <= delayMatch1_reg_next[9];
269       delayMatch1_reg[10] <= delayMatch1_reg_next[10];
270       delayMatch1_reg[11] <= delayMatch1_reg_next[11];
271       delayMatch1_reg[12] <= delayMatch1_reg_next[12];
272       delayMatch1_reg[13] <= delayMatch1_reg_next[13];
273       delayMatch1_reg[14] <= delayMatch1_reg_next[14];
274       delayMatch1_reg[15] <= delayMatch1_reg_next[15];
275       delayMatch1_reg[16] <= delayMatch1_reg_next[16];
276       delayMatch1_reg[17] <= delayMatch1_reg_next[17];
277       delayMatch1_reg[18] <= delayMatch1_reg_next[18];
278       delayMatch1_reg[19] <= delayMatch1_reg_next[19];
279     end
280   end
281 end

282
283 assign Data_Type_Conversion_out1_1 = delayMatch1_reg[19];
284 assign delayMatch1_reg_next[0] = Data_Type_Conversion_out1_1;
285 assign delayMatch1_reg_next[1] = delayMatch1_reg[0];
286 assign delayMatch1_reg_next[2] = delayMatch1_reg[1];
287 assign delayMatch1_reg_next[3] = delayMatch1_reg[2];
288 assign delayMatch1_reg_next[4] = delayMatch1_reg[3];
289 assign delayMatch1_reg_next[5] = delayMatch1_reg[4];
290 assign delayMatch1_reg_next[6] = delayMatch1_reg[5];
291 assign delayMatch1_reg_next[7] = delayMatch1_reg[6];
292 assign delayMatch1_reg_next[8] = delayMatch1_reg[7];
293 assign delayMatch1_reg_next[9] = delayMatch1_reg[8];
294 assign delayMatch1_reg_next[10] = delayMatch1_reg[9];
295 assign delayMatch1_reg_next[11] = delayMatch1_reg[10];
296 assign delayMatch1_reg_next[12] = delayMatch1_reg[11];
297 assign delayMatch1_reg_next[13] = delayMatch1_reg[12];
298 assign delayMatch1_reg_next[14] = delayMatch1_reg[13];
299 assign delayMatch1_reg_next[15] = delayMatch1_reg[14];
300 assign delayMatch1_reg_next[16] = delayMatch1_reg[15];
301 assign delayMatch1_reg_next[17] = delayMatch1_reg[16];
302 assign delayMatch1_reg_next[18] = delayMatch1_reg[17];
303 assign delayMatch1_reg_next[19] = delayMatch1_reg[18];
304
305
306
307 assign o_PHASE = Data_Type_Conversion_out1_1;
308
309 assign ce_out = clk_enable;
310
311 assign o_VALID = i_VALID_1;
312
313 endmodule // Enabled_Subsystem
314
315

```



RTL Viewer:



Для FloatPoint:



```

40 timescale 1 ns / 1 ns
41
42 module EnabledSubsystem
43 (
44     clk,
45     reset,
46     clk_enable,
47     i_VALID,
48     i_COMPLEX_VALUE_re,
49     i_COMPLEX_VALUE_im,
50     ce_out,
51     o_VALID,
52     o_MAGNITUDE,
53     o_PHASE);
54
55 input clk;
56 input reset;
57 input clk_enable;
58 input [63:0] i_VALID; // double
59 input [31:0] i_COMPLEX_VALUE_re; // single
60 input [31:0] i_COMPLEX_VALUE_im; // single
61 output ce_out;
62 output [63:0] o_VALID; // double
63 output [31:0] o_MAGNITUDE; // single
64 output [31:0] o_PHASE; // single
65
66
67 wire enb;
68 reg [63:0] delayMatch_reg [0:46]; // ufix64 [47]
69 wire [63:0] delayMatch_reg_next [0:46]; // ufix64 [47]
70 wire [63:0] i_VALID_1; // ufix64
71 wire [31:0] Product_out1; // ufix32
72 wire [31:0] Product1_out1; // ufix32
73 wire [31:0] Add_out1; // ufix32
74 wire [31:0] Sqrt_out1; // ufix32
75 wire [31:0] Trigonometric_Function_out1; // ufix32
76 reg [31:0] delayMatch1_reg [0:4]; // ufix32 [5]
77 wire [31:0] delayMatch1_reg_next [0:4]; // ufix32 [5]
78 wire [31:0] Trigonometric_Function_out1_1; // ufix32
79
80
81 assign enb = clk_enable;
82
83 always @(posedge clk or posedge reset)
84 begin : delayMatch_process
85 if (reset == 'b1') begin

```

```

64 output [31:0] o_PHASE; // single
65
66
67 wire enb;
68 reg [63:0] delayMatch_reg [0:46]; // ufix64 [47]
69 wire [63:0] delayMatch_reg_next [0:46]; // ufix64 [47]
70 wire [63:0] i_VALID_1; // ufix64
71 wire [31:0] Product_out1; // ufix32
72 wire [31:0] Product1_out1; // ufix32
73 wire [31:0] Add_out1; // ufix32
74 wire [31:0] Sqrt_out1; // ufix32
75 wire [31:0] Trigonometric_Function_out1; // ufix32
76 reg [31:0] delayMatch1_reg [0:4]; // ufix32 [5]
77 wire [31:0] delayMatch1_reg_next [0:4]; // ufix32 [5]
78 wire [31:0] Trigonometric_Function_out1_1; // ufix32
79
80
81 assign enb = clk_enable;
82
83 always @(posedge clk or posedge reset)
84 begin : delayMatch_process
85 if (reset == 'b1') begin
86 delayMatch_reg[0] <= 64'h0000000000000000;
87 delayMatch_reg[1] <= 64'h0000000000000000;
88 delayMatch_reg[2] <= 64'h0000000000000000;
89 delayMatch_reg[3] <= 64'h0000000000000000;
90 delayMatch_reg[4] <= 64'h0000000000000000;
91 delayMatch_reg[5] <= 64'h0000000000000000;
92 delayMatch_reg[6] <= 64'h0000000000000000;
93 delayMatch_reg[7] <= 64'h0000000000000000;
94 delayMatch_reg[8] <= 64'h0000000000000000;
95 delayMatch_reg[9] <= 64'h0000000000000000;
96 delayMatch_reg[10] <= 64'h0000000000000000;
97 delayMatch_reg[11] <= 64'h0000000000000000;
98 delayMatch_reg[12] <= 64'h0000000000000000;
99 delayMatch_reg[13] <= 64'h0000000000000000;
100 delayMatch_reg[14] <= 64'h0000000000000000;
101 delayMatch_reg[15] <= 64'h0000000000000000;
102 delayMatch_reg[16] <= 64'h0000000000000000;
103 delayMatch_reg[17] <= 64'h0000000000000000;
104 delayMatch_reg[18] <= 64'h0000000000000000;
105 delayMatch_reg[19] <= 64'h0000000000000000;
106 delayMatch_reg[20] <= 64'h0000000000000000;
107 delayMatch_reg[21] <= 64'h0000000000000000;
108 delayMatch_reg[22] <= 64'h0000000000000000;
109 delayMatch_reg[23] <= 64'h0000000000000000;

```

```

88 delayMatch_reg[2] <= 64'h0000000000000000;
89 delayMatch_reg[3] <= 64'h0000000000000000;
90 delayMatch_reg[4] <= 64'h0000000000000000;
91 delayMatch_reg[5] <= 64'h0000000000000000;
92 delayMatch_reg[6] <= 64'h0000000000000000;
93 delayMatch_reg[7] <= 64'h0000000000000000;
94 delayMatch_reg[8] <= 64'h0000000000000000;
95 delayMatch_reg[9] <= 64'h0000000000000000;
96 delayMatch_reg[10] <= 64'h0000000000000000;
97 delayMatch_reg[11] <= 64'h0000000000000000;
98 delayMatch_reg[12] <= 64'h0000000000000000;
99 delayMatch_reg[13] <= 64'h0000000000000000;
100 delayMatch_reg[14] <= 64'h0000000000000000;
101 delayMatch_reg[15] <= 64'h0000000000000000;
102 delayMatch_reg[16] <= 64'h0000000000000000;
103 delayMatch_reg[17] <= 64'h0000000000000000;
104 delayMatch_reg[18] <= 64'h0000000000000000;
105 delayMatch_reg[19] <= 64'h0000000000000000;
106 delayMatch_reg[20] <= 64'h0000000000000000;
107 delayMatch_reg[21] <= 64'h0000000000000000;
108 delayMatch_reg[22] <= 64'h0000000000000000;
109 delayMatch_reg[23] <= 64'h0000000000000000;
110 delayMatch_reg[24] <= 64'h0000000000000000;
111 delayMatch_reg[25] <= 64'h0000000000000000;
112 delayMatch_reg[26] <= 64'h0000000000000000;
113 delayMatch_reg[27] <= 64'h0000000000000000;
114 delayMatch_reg[28] <= 64'h0000000000000000;
115 delayMatch_reg[29] <= 64'h0000000000000000;
116 delayMatch_reg[30] <= 64'h0000000000000000;
117 delayMatch_reg[31] <= 64'h0000000000000000;
118 delayMatch_reg[32] <= 64'h0000000000000000;
119 delayMatch_reg[33] <= 64'h0000000000000000;
120 delayMatch_reg[34] <= 64'h0000000000000000;
121 delayMatch_reg[35] <= 64'h0000000000000000;
122 delayMatch_reg[36] <= 64'h0000000000000000;
123 delayMatch_reg[37] <= 64'h0000000000000000;
124 delayMatch_reg[38] <= 64'h0000000000000000;
125 delayMatch_reg[39] <= 64'h0000000000000000;
126 delayMatch_reg[40] <= 64'h0000000000000000;
127 delayMatch_reg[41] <= 64'h0000000000000000;
128 delayMatch_reg[42] <= 64'h0000000000000000;
129 delayMatch_reg[43] <= 64'h0000000000000000;
130 delayMatch_reg[44] <= 64'h0000000000000000;
131 delayMatch_reg[45] <= 64'h0000000000000000;
132 delayMatch_reg[46] <= 64'h0000000000000000;
133 end

```



```

187 assign i_VALID_1 = delayMatch_reg[46];
188 assign delayMatch_reg_next[0] = i_VALID;
189 assign delayMatch_reg_next[1] = delayMatch_reg[0];
190 assign delayMatch_reg_next[2] = delayMatch_reg[1];
191 assign delayMatch_reg_next[3] = delayMatch_reg[2];
192 assign delayMatch_reg_next[4] = delayMatch_reg[3];
193 assign delayMatch_reg_next[5] = delayMatch_reg[4];
194 assign delayMatch_reg_next[6] = delayMatch_reg[5];
195 assign delayMatch_reg_next[7] = delayMatch_reg[6];
196 assign delayMatch_reg_next[8] = delayMatch_reg[7];
197 assign delayMatch_reg_next[9] = delayMatch_reg[8];
198 assign delayMatch_reg_next[10] = delayMatch_reg[9];
199 assign delayMatch_reg_next[11] = delayMatch_reg[10];
200 assign delayMatch_reg_next[12] = delayMatch_reg[11];
201 assign delayMatch_reg_next[13] = delayMatch_reg[12];
202 assign delayMatch_reg_next[14] = delayMatch_reg[13];
203 assign delayMatch_reg_next[15] = delayMatch_reg[14];
204 assign delayMatch_reg_next[16] = delayMatch_reg[15];
205 assign delayMatch_reg_next[17] = delayMatch_reg[16];
206 assign delayMatch_reg_next[18] = delayMatch_reg[17];
207 assign delayMatch_reg_next[19] = delayMatch_reg[18];
208 assign delayMatch_reg_next[20] = delayMatch_reg[19];
209 assign delayMatch_reg_next[21] = delayMatch_reg[20];
210 assign delayMatch_reg_next[22] = delayMatch_reg[21];
211 assign delayMatch_reg_next[23] = delayMatch_reg[22];
212 assign delayMatch_reg_next[24] = delayMatch_reg[23];
213 assign delayMatch_reg_next[25] = delayMatch_reg[24];
214 assign delayMatch_reg_next[26] = delayMatch_reg[25];
215 assign delayMatch_reg_next[27] = delayMatch_reg[26];
216 assign delayMatch_reg_next[28] = delayMatch_reg[27];
217 assign delayMatch_reg_next[29] = delayMatch_reg[28];
218 assign delayMatch_reg_next[30] = delayMatch_reg[29];
219 assign delayMatch_reg_next[31] = delayMatch_reg[30];
220 assign delayMatch_reg_next[32] = delayMatch_reg[31];
221 assign delayMatch_reg_next[33] = delayMatch_reg[32];
222 assign delayMatch_reg_next[34] = delayMatch_reg[33];
223 assign delayMatch_reg_next[35] = delayMatch_reg[34];
224 assign delayMatch_reg_next[36] = delayMatch_reg[35];
225 assign delayMatch_reg_next[37] = delayMatch_reg[36];
226 assign delayMatch_reg_next[38] = delayMatch_reg[37];
227 assign delayMatch_reg_next[39] = delayMatch_reg[38];
228 assign delayMatch_reg_next[40] = delayMatch_reg[39];
229 assign delayMatch_reg_next[41] = delayMatch_reg[40];
230 assign delayMatch_reg_next[42] = delayMatch_reg[41];
231 assign delayMatch_reg_next[43] = delayMatch_reg[42];
232 assign delayMatch_reg_next[44] = delayMatch_reg[43];
233 assign delayMatch_reg_next[45] = delayMatch_reg[44];
234 assign delayMatch_reg_next[46] = delayMatch_reg[45];
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
260
261
262
263
264
265

```

```

220 assign delayMatch_reg_next[32] = delayMatch_reg[31];
221 assign delayMatch_reg_next[33] = delayMatch_reg[32];
222 assign delayMatch_reg_next[34] = delayMatch_reg[33];
223 assign delayMatch_reg_next[35] = delayMatch_reg[34];
224 assign delayMatch_reg_next[36] = delayMatch_reg[35];
225 assign delayMatch_reg_next[37] = delayMatch_reg[36];
226 assign delayMatch_reg_next[38] = delayMatch_reg[37];
227 assign delayMatch_reg_next[39] = delayMatch_reg[38];
228 assign delayMatch_reg_next[40] = delayMatch_reg[39];
229 assign delayMatch_reg_next[41] = delayMatch_reg[40];
230 assign delayMatch_reg_next[42] = delayMatch_reg[41];
231 assign delayMatch_reg_next[43] = delayMatch_reg[42];
232 assign delayMatch_reg_next[44] = delayMatch_reg[43];
233 assign delayMatch_reg_next[45] = delayMatch_reg[44];
234 assign delayMatch_reg_next[46] = delayMatch_reg[45];
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
260
261
262
263
264
265

```

```

nfp_mul_single u_nfp_mul_comp (.clk(clk),
                             .reset(reset),
                             .enb(clk_enable),
                             .nfp_in1(i_COMPLEX_VALUE_re), // single
                             .nfp_in2(i_COMPLEX_VALUE_im), // single
                             .nfp_out(Product_out1) // single
);

nfp_mul_single u_nfp_mul_comp_1 (.clk(clk),
                                .reset(reset),
                                .enb(clk_enable),
                                .nfp_in1(i_COMPLEX_VALUE_im), // single
                                .nfp_in2(i_COMPLEX_VALUE_re), // single
                                .nfp_out(Product1_out1) // single
);

nfp_add_single u_nfp_add_comp (.clk(clk),
                              .reset(reset),
                              .enb(clk_enable),
                              .nfp_in1(Product_out1), // single
                              .nfp_in2(Product1_out1), // single
                              .nfp_out(Add_out1) // single
);

nfp_sqrt_single u_nfp_sqrt_comp (.clk(clk),
                                .reset(reset),
                                .enb(clk_enable),
                                .nfp_in(Add_out1) // single
);

```

```

262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305

```

```

nfp_sqrt_single u_nfp_sqrt_comp (.clk(clk),
                                .reset(reset),
                                .enb(clk_enable),
                                .nfp_in(Add_out1), // single
                                .nfp_out(Sqrt_out1) // single
);

nfp_atan2_single u_nfp_atan2_comp (.clk(clk),
                                  .reset(reset),
                                  .enb(clk_enable),
                                  .nfp_in1(i_COMPLEX_VALUE_im), // single
                                  .nfp_in2(i_COMPLEX_VALUE_re), // single
                                  .nfp_out(Trigonometric_Function_out1) // single
);

always @(posedge clk or posedge reset)
begin : delayMatch1_process
if (reset == 1'b1) begin
delayMatch1_reg[0] <= 32'h00000000;
delayMatch1_reg[1] <= 32'h00000000;
delayMatch1_reg[2] <= 32'h00000000;
delayMatch1_reg[3] <= 32'h00000000;
delayMatch1_reg[4] <= 32'h00000000;
end
else begin
if (enb) begin
delayMatch1_reg[0] <= delayMatch1_reg_next[0];
delayMatch1_reg[1] <= delayMatch1_reg_next[1];
delayMatch1_reg[2] <= delayMatch1_reg_next[2];
delayMatch1_reg[3] <= delayMatch1_reg_next[3];
delayMatch1_reg[4] <= delayMatch1_reg_next[4];
end
end
end

assign Trigonometric_Function_out1_1 = delayMatch1_reg[4];
assign delayMatch1_reg_next[0] = Trigonometric_Function_out1;
assign delayMatch1_reg_next[1] = delayMatch1_reg[0];
assign delayMatch1_reg_next[2] = delayMatch1_reg[1];
assign delayMatch1_reg_next[3] = delayMatch1_reg[2];
assign delayMatch1_reg_next[4] = delayMatch1_reg[3];

```

```

277 always @(posedge clk or posedge reset)
278 begin : delayMatch1_process
279 if (reset == 1'b1) begin
280 delayMatch1_reg[0] <= 32'h00000000;
281 delayMatch1_reg[1] <= 32'h00000000;
282 delayMatch1_reg[2] <= 32'h00000000;
283 delayMatch1_reg[3] <= 32'h00000000;
284 delayMatch1_reg[4] <= 32'h00000000;
285 end
286 else begin
287 if (enb) begin
288 delayMatch1_reg_next[0] <= delayMatch1_reg_next[0];
289 delayMatch1_reg_next[1] <= delayMatch1_reg_next[1];
290 delayMatch1_reg_next[2] <= delayMatch1_reg_next[2];
291 delayMatch1_reg_next[3] <= delayMatch1_reg_next[3];
292 delayMatch1_reg_next[4] <= delayMatch1_reg_next[4];
293 end
294 end
295 end
296
297 assign Trigonometric_Function_out1_1 = delayMatch1_reg[4];
298 assign delayMatch1_reg_next[0] = Trigonometric_Function_out1;
299 assign delayMatch1_reg_next[1] = delayMatch1_reg[0];
300 assign delayMatch1_reg_next[2] = delayMatch1_reg[1];
301 assign delayMatch1_reg_next[3] = delayMatch1_reg[2];
302 assign delayMatch1_reg_next[4] = delayMatch1_reg[3];
303
304
305
306 assign ce_out = clk_enable;
307
308 assign o_VALID = i_VALID_1;
309
310 assign o_MAGNITUDE = Sqrt_out1;
311
312 assign o_PHASE = Trigonometric_Function_out1_1;
313
314 endmodule // Enabled_Subsystem
315
316

```

The screenshot displays the Quartus Prime IDE interface. The main window shows the 'Compilation Report - floatpoint' for a project named 'floatpoint'. The report is dated 'Fri Jan 13 05:17:17 2023' and indicates a 'Successful' compilation. The device is identified as 'Cyclone V' (SC5EM4SF31C6). The report includes sections for 'Flow Status', 'Quartus Prime Version', 'Revision Name', 'Top-level Entity Name', 'Family', 'Device', 'Timing Models', 'Logic utilization (in ALMs)', 'Total registers', 'Total pins', 'Total virtual pins', 'Total block memory bits', 'Total DSP blocks', 'Total HSSI RX PCs', 'Total HSSI PM4 RX Deserializers', 'Total HSSI TX PCs', 'Total HSSI PM4 TX Serializers', 'Total PLLs', and 'Total DLLs'. The Messages window at the bottom shows a warning about setup requirements for the timing analyzer.

File Edit View Project Assignments Processing Tools Window Help

Search altium.com

Project Navigator

Files

- floatpoint
 - floatpoint_single.v
 - floatpoint_single.v
 - floatpoint_single.v
 - floatpoint_single.v
 - floatpoint_single.v
 - Enabled_Subsystem.v

Table of Contents

- Flow Summary
- Flow Settings
- Flow Non-Default Global Settings
- Flow Elapsed Time
- Flow OS Summary
- Flow Log
- Analysis & Synthesis
 - Filter
 - Flow Messages
 - Flow Suppressed Messages
- Assembler
- Timing Analyzer

Flow Summary

Flow Status: Successful - Fri Jan 13 05:17:17 2023

Quartus Prime Version: 20.1 Build 720 11/11/2020 SJ Lite Edition

Revision Name: floatpoint

Top-level Entity Name: Enabled_Subsystem

Family: Cyclone V

Device: SC5EM4SF31C6

Timing Models: Final

Logic utilization (in ALMs): 2,215 / 32,070 (7 %)

Total registers: 4169

Total pins: 260 / 437 (57 %)

Total virtual pins: 0

Total block memory bits: 5,933 / 4,065,280 (<1 %)

Total DSP blocks: 10 / 87 (11 %)

Total HSSI RX PCs: 0

Total HSSI PM4 RX Deserializers: 0

Total HSSI TX PCs: 0

Total HSSI PM4 TX Serializers: 0

Total PLLs: 0 / 6 (0 %)

Total DLLs: 0 / 4 (0 %)

Tasks

Compilation

Task

- Compile Design
 - Analysis & Synthesis
 - Filter Place & Route
 - Assembler (Generate programming files)
 - Timing Analysis
 - EDA Netlist writer
 - Edit Settings
 - Program Device (Open Programmer)

Messages

Type ID Message

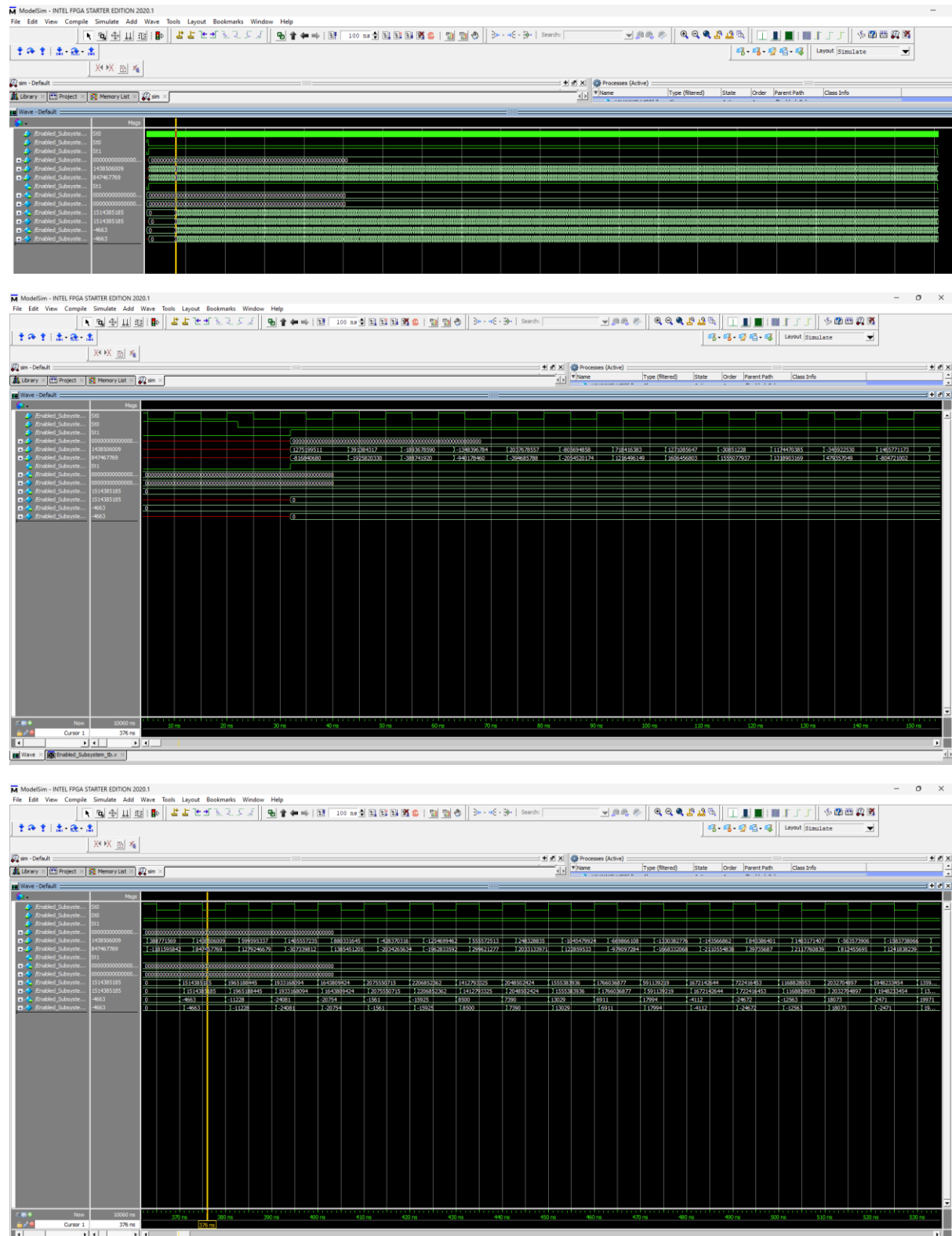
- 3321246 worst-case removal slack is 0.161
- 3321446 worst-case minimum pulse width slack is -2.225
- 332114 Report Metastability: Found 212 synchronizer chains.
- 3321202 design is not fully constrained for setup requirements
- 3321202 design is not fully constrained for hold requirements
- Quartus Prime Timing Analyzer was successful. 0 errors, 6 warnings
- 293000 Quartus Prime full compilation was successful. 0 errors, 14 warnings

System (1) Processing (214)

100% 00:03



Для FixedPoint:



Для FloatPoint:

