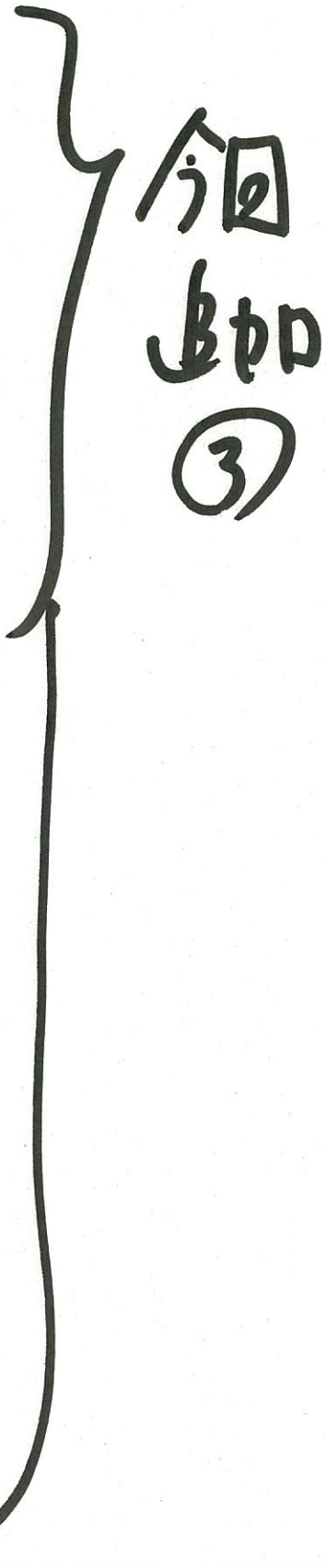
```
1   ----------------------------------------------------------------------------------
2   -- Company:
3   -- Engineer:
4   --
5   -- Create Date:    18:09:54 01/24/2012
6   -- Design Name:
7   -- Module Name:    fft_circuitA - Behavioral
8   -- Project Name:
9   -- Target Devices:
10  -- Tool versions:
11  -- Description:
12  --
13  -- Dependencies:
14  --
15  -- Revision:
16  -- Revision 0.01 - File Created
17  -- Additional Comments:
18  --
19  ----------------------------------------------------------------------------------
20  library IEEE;
21  use IEEE.STD_LOGIC_1164.ALL;
22  use IEEE.STD_LOGIC_SIGNED.ALL; -- TSUIKA
23
24  -- Uncomment the following library declaration if using
25  -- arithmetic functions with Signed or Unsigned values
26  --use IEEE.NUMERIC_STD.ALL;
27
28  -- Uncomment the following library declaration if instantiating
29  -- any Xilinx primitives in this code.
30  --library UNISIM;
31  --use UNISIM.VComponents.all;
32
33  entity fft_circuitA is
34      Port ( s_re0 : in  STD_LOGIC_VECTOR (15 downto 0);
35             s_im0 : in  STD_LOGIC_VECTOR (15 downto 0);
36             s_re1 : in  STD_LOGIC_VECTOR (15 downto 0);
37             s_im1 : in  STD_LOGIC_VECTOR (15 downto 0);
38             s_re2 : in  STD_LOGIC_VECTOR (15 downto 0);
39             s_im2 : in  STD_LOGIC_VECTOR (15 downto 0);
40             s_re3 : in  STD_LOGIC_VECTOR (15 downto 0);
41             s_im3 : in  STD_LOGIC_VECTOR (15 downto 0);
42             s_re4 : in  STD_LOGIC_VECTOR (15 downto 0);
43             s_im4 : in  STD_LOGIC_VECTOR (15 downto 0);
44             s_re5 : in  STD_LOGIC_VECTOR (15 downto 0);
45             s_im5 : in  STD_LOGIC_VECTOR (15 downto 0);
46             s_re6 : in  STD_LOGIC_VECTOR (15 downto 0);
47             s_im6 : in  STD_LOGIC_VECTOR (15 downto 0);
48             s_re7 : in  STD_LOGIC_VECTOR (15 downto 0);
49             s_im7 : in  STD_LOGIC_VECTOR (15 downto 0);
50             G_re0 : out  STD_LOGIC_VECTOR (15 downto 0);
51             G_im0 : out  STD_LOGIC_VECTOR (15 downto 0);
52             G_re1 : out  STD_LOGIC_VECTOR (15 downto 0);
53             G_im1 : out  STD_LOGIC_VECTOR (15 downto 0);
54             G_re2 : out  STD_LOGIC_VECTOR (15 downto 0);
55             G_im2 : out  STD_LOGIC_VECTOR (15 downto 0);
56             G_re3 : out  STD_LOGIC_VECTOR (15 downto 0);
57             G_im3 : out  STD_LOGIC_VECTOR (15 downto 0);
58             G_re4 : out  STD_LOGIC_VECTOR (15 downto 0);
59             G_im4 : out  STD_LOGIC_VECTOR (15 downto 0);
60             G_re5 : out  STD_LOGIC_VECTOR (15 downto 0);
61             G_im5 : out  STD_LOGIC_VECTOR (15 downto 0);
62             G_re6 : out  STD_LOGIC_VECTOR (15 downto 0);
```

完成版 8FFT 回路 記述

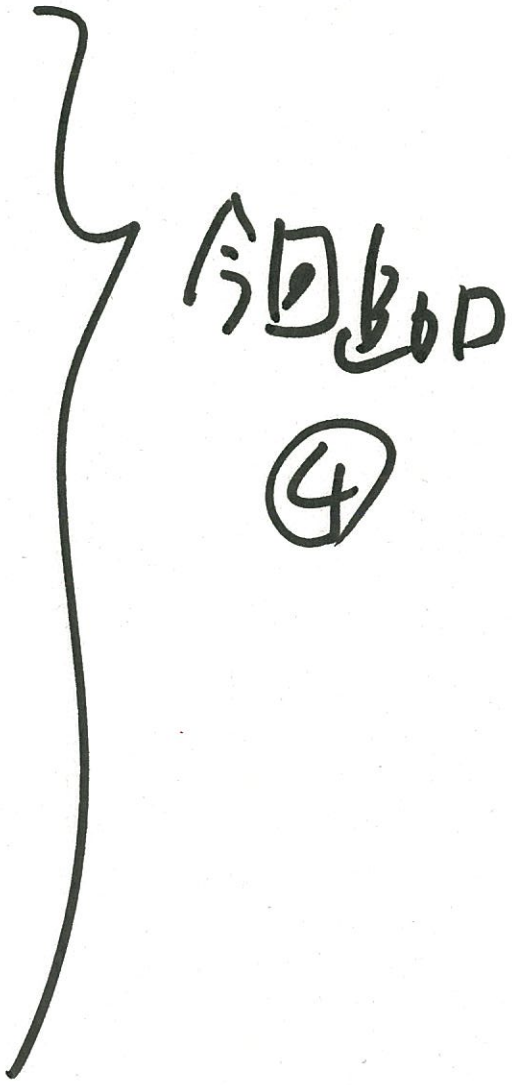
```
63              G_im6 : out  STD_LOGIC_VECTOR (15 downto 0);
64              G_re7 : out  STD_LOGIC_VECTOR (15 downto 0);
65              G_im7 : out  STD_LOGIC_VECTOR (15 downto 0));
66  end fft_circuitA;
67
68  architecture Behavioral of fft_circuitA is
69
70  --s1 signals
71  signal s1_re0 :  STD_LOGIC_VECTOR (15 downto 0);
72  signal s1_im0 :  STD_LOGIC_VECTOR (15 downto 0);
73  signal s1_re1 :  STD_LOGIC_VECTOR (15 downto 0);
74  signal s1_im1 :  STD_LOGIC_VECTOR (15 downto 0);
75  signal s1_re2 :  STD_LOGIC_VECTOR (15 downto 0);
76  signal s1_im2 :  STD_LOGIC_VECTOR (15 downto 0);
77  signal s1_re3 :  STD_LOGIC_VECTOR (15 downto 0);
78  signal s1_im3 :  STD_LOGIC_VECTOR (15 downto 0);
79  signal s1_re4 :  STD_LOGIC_VECTOR (15 downto 0);
80  signal s1_im4 :  STD_LOGIC_VECTOR (15 downto 0);
81  signal s1_re5 :  STD_LOGIC_VECTOR (15 downto 0);
82  signal s1_im5 :  STD_LOGIC_VECTOR (15 downto 0);
83  signal s1_re6 :  STD_LOGIC_VECTOR (15 downto 0);
84  signal s1_im6 :  STD_LOGIC_VECTOR (15 downto 0);
85  signal s1_re7 :  STD_LOGIC_VECTOR (15 downto 0);
86  signal s1_im7 :  STD_LOGIC_VECTOR (15 downto 0);
87  --
88  signal W8_re1 :  STD_LOGIC_VECTOR (9 downto 0) := "0101101010"; -- +0.7071 in
    <10,0,t>
89  signal W8_im1 :  STD_LOGIC_VECTOR (9 downto 0) := "1010010110"; -- -0.7071 in
    <10,0,t>
90  --
91  signal tmp_s1_re5 : STD_LOGIC_VECTOR (25 downto 0);
92  signal tmp_s1_im5 : STD_LOGIC_VECTOR (25 downto 0);
93  signal tmp_s1_re7 : STD_LOGIC_VECTOR (25 downto 0);
94  signal tmp_s1_im7 : STD_LOGIC_VECTOR (25 downto 0);
95
96  -- *** H24/2/1 ***
97  signal s2_re0 :  STD_LOGIC_VECTOR (15 downto 0);
98  signal s2_im0 :  STD_LOGIC_VECTOR (15 downto 0);
99  signal s2_re1 :  STD_LOGIC_VECTOR (15 downto 0);
100 signal s2_im1 :  STD_LOGIC_VECTOR (15 downto 0);
101 signal s2_re2 :  STD_LOGIC_VECTOR (15 downto 0);
102 signal s2_im2 :  STD_LOGIC_VECTOR (15 downto 0);
103 signal s2_re3 :  STD_LOGIC_VECTOR (15 downto 0);
104 signal s2_im3 :  STD_LOGIC_VECTOR (15 downto 0);
105 signal s2_re4 :  STD_LOGIC_VECTOR (15 downto 0);
106 signal s2_im4 :  STD_LOGIC_VECTOR (15 downto 0);
107 signal s2_re5 :  STD_LOGIC_VECTOR (15 downto 0);
108 signal s2_im5 :  STD_LOGIC_VECTOR (15 downto 0);
109 signal s2_re6 :  STD_LOGIC_VECTOR (15 downto 0);
110 signal s2_im6 :  STD_LOGIC_VECTOR (15 downto 0);
111 signal s2_re7 :  STD_LOGIC_VECTOR (15 downto 0);
112 signal s2_im7 :  STD_LOGIC_VECTOR (15 downto 0);
113
114 signal s3_re0 :  STD_LOGIC_VECTOR (15 downto 0);
115 signal s3_im0 :  STD_LOGIC_VECTOR (15 downto 0);
116 signal s3_re1 :  STD_LOGIC_VECTOR (15 downto 0);
117 signal s3_im1 :  STD_LOGIC_VECTOR (15 downto 0);
118 signal s3_re2 :  STD_LOGIC_VECTOR (15 downto 0);
119 signal s3_im2 :  STD_LOGIC_VECTOR (15 downto 0);
120 signal s3_re3 :  STD_LOGIC_VECTOR (15 downto 0);
121 signal s3_im3 :  STD_LOGIC_VECTOR (15 downto 0);
122 signal s3_re4 :  STD_LOGIC_VECTOR (15 downto 0);
```

```
123   signal s3_im4 :  STD_LOGIC_VECTOR (15 downto 0);
124   signal s3_re5 :  STD_LOGIC_VECTOR (15 downto 0);
125   signal s3_im5 :  STD_LOGIC_VECTOR (15 downto 0);
126   signal s3_re6 :  STD_LOGIC_VECTOR (15 downto 0);
127   signal s3_im6 :  STD_LOGIC_VECTOR (15 downto 0);
128   signal s3_re7 :  STD_LOGIC_VECTOR (15 downto 0);
129   signal s3_im7 :  STD_LOGIC_VECTOR (15 downto 0);
130
131   begin
132   -- STAGE1
133   -- s1(0) = s(0) + s(4)
134   --s1_re(0) <= s_re(0) + s_re(4);
135   --s1_im(0) <= s_im(0) + s_im(4);
136   s1_re0 <= s_re0 + s_re4;
137   s1_im0 <= s_im0 + s_im4;
138
139   -- s1(4) = {s(0) - s(4)}*W8**0
140   --s1_re(4) <= s_re(0) - s_re(4);
141   --s1_im(4) <= s_im(0) - s_im(4);
142   s1_re4 <= s_re0 - s_re4;
143   s1_im4 <= s_im0 - s_im4;
144
145   -- s1(1) = s(1) + s(5)
146   --s1_re(1) <= s_re(1) + s_re(5);
147   --s1_im(1) <= s_im(1) + s_im(5);
148   s1_re1 <= s_re1 + s_re5;
149   s1_im1 <= s_im1 + s_im5;
150
151   -- s1(5) = {s(1) - s(5)}*W8**1
152   --t_re      := s_re(1) - s_re(5);
153   --t_im      := s_im(1) - s_im(5);
154   --s1_re(5) <= W8_re(1) * t_re - W8_im(1) * t_im;
155   --s1_im(5) <= W8_im(1) * t_re + W8_re(1) * t_im;
156   tmp_s1_re5 <= (W8_re1 * (s_re1 - s_re5)) - (W8_im1 * (s_im1 - s_im5));
157   tmp_s1_im5 <= (W8_im1 * (s_re1 - s_re5)) + (W8_re1 * (s_im1 - s_im5));
158   s1_re5 <= tmp_s1_re5(24 downto 9);
159   s1_im5 <= tmp_s1_im5(24 downto 9);
160
161   -- s1(2) = s(2) + s(6)
162   --s1_re(2) <= s_re(2) + s_re(6);
163   --s1_im(2) <= s_im(2) + s_im(6);
164   s1_re2 <= s_re2 + s_re6;
165   s1_im2 <= s_im2 + s_im6;
166
167   -- s1(6) = {s(2) - s(6)}*W8**2
168   --W8_re(2) <=  0.0;    W8_im(2) <= -1.0;
169   --t_re      := s_re(2) - s_re(6);
170   --t_im      := s_im(2) - s_im(6);
171   --s1_re(6) <= W8_re(2) * t_re - W8_im(2) * t_im =  t_im;
172   --s1_im(6) <= W8_im(2) * t_re + W8_re(2) * t_im = -t_re;
173   s1_re6 <= s_im2 - s_im6;
174   s1_im6 <= s_re6 - s_re2;
175
176   -- s1(3) = s(3) + s(7)
177   --s1_re(3) <= s_re(3) + s_re(7);
178   --s1_im(3) <= s_im(3) + s_im(7);
179   s1_re3 <= s_re3 + s_re7;
180   s1_im3 <= s_im3 + s_im7;
181
182   -- s1(7) = {s(3) - s(7)}*W8**3
183   --W8_re(3) <= -0.7071; W8_im(3) <= -0.7071;
184   --t_re      := s_re(3) - s_re(7);
```

```
185   --t_im     := s_im(3) - s_im(7);
186   --s1_re(7) <= W8_re(3) * t_re - W8_im(3) * t_im=W8_im1 * t_re - W8_im1 * t_im;
187   --s1_im(7) <= W8_im(3) * t_re + W8_re(3) * t_im=W8_im1 * t_re + W8_im1 * t_im;
188   tmp_s1_re7 <= W8_im1 * (s_re3 - s_re7 - s_im3 + s_im7);
189   tmp_s1_im7 <= W8_im1 * (s_re3 - s_re7 + s_im3 - s_im7);
190   s1_re7 <= tmp_s1_re7(24 downto 9);
191   s1_im7 <= tmp_s1_im7(24 downto 9);
192
193   -- *** H24/2/1 ***
194   --STAGE2
195   ----   s2(0)=s1(0)+s1(2)
196   --s2_re(0) <= s1_re(0) + s1_re(2);
197   --s2_im(0) <= s1_im(0) + s1_im(2);
198   s2_re0 <= s1_re0 + s1_re2;
199   s2_im0 <= s1_im0 + s1_im2;
200   ----   s2(2)=(s1(0)-s1(2))*W4**0
201   --s2_re(2) <= s1_re(0) - s1_re(2);
202   --s2_im(2) <= s1_im(0) - s1_im(2);
203   s2_re2 <= s1_re0 - s1_re2;
204   s2_im2 <= s1_im0 - s1_im2;
205   ---- s2(1)=s1(1)+s2(3)
206   --s2_re(1) <= s1_re(1) + s1_re(3);
207   --s2_im(1) <= s1_im(1) + s1_im(3);
208   s2_re1 <= s1_re1 + s1_re3;
209   s2_im1 <= s1_im1 + s1_im3;
210   ---- s2(3)=(s1(1)-s2(3))*W4**1
211   --t_re      := s1_re(1) - s1_re(3);
212   --t_im      := s1_im(1) - s1_im(3);
213   --s2_re(3) <= W8_re(2) * t_re - W8_im(2) * t_im;
214   --s2_im(3) <= W8_im(2) * t_re + W8_re(2) * t_im;
215   -- W8_re(2)=0, W8_im(2)=-1
216   s2_re3 <= s1_im1 - s1_im3;
217   s2_im3 <= s1_re3 - s1_re1;
218   ---- s2(4)=s1(4)+s1(6)
219   --s2_re(4) <= s1_re(4) + s1_re(6);
220   --s2_im(4) <= s1_im(4) + s1_im(6);
221   s2_re4 <= s1_re4 + s1_re6;
222   s2_im4 <= s1_im4 + s1_im6;
223   ---- s2(6)=(s1(4)-s1(6))*W4**0
224   --s2_re(6) <= s1_re(4) - s1_re(6);
225   --s2_im(6) <= s1_im(4) - s1_im(6);
226   s2_re6 <= s1_re4 - s1_re6;
227   s2_im6 <= s1_im4 - s1_im6;
228   ---- s2(5)=s1(5)+s1(7)
229   --s2_re(5) <= s1_re(5) + s1_re(7);
230   --s2_im(5) <= s1_im(5) + s1_im(7);
231   s2_re5 <= s1_re5 + s1_re7;
232   s2_im5 <= s1_im5 + s1_im7;
233   ---- s2(7)=(s1(5)-s1(7))*W4**1
234   --t_re      := s1_re(5) - s1_re(7);
235   --t_im      := s1_im(5) - s1_im(7);
236   --s2_re(7) <= W8_re(2) * t_re - W8_im(2) * t_im;
237   --s2_im(7) <= W8_im(2) * t_re + W8_re(2) * t_im;
238   -- W8_re(2)=0, W8_im(2)=-1
239   s2_re7 <= s1_im5 - s1_im7;
240   s2_im7 <= s1_re7 - s1_re5;
241
242   -- STAGE3
243   ---- s3(0)=s2(0)+s2(1)
244   --s3_re(0) <= s2_re(0) + s2_re(1);
245   --s3_im(0) <= s2_im(0) + s2_im(1);
246   s3_re0 <= s2_re0 + s2_re1;
```



```
247   s3_im0 <= s2_im0 + s2_im1;
248   ---- s3(1)=s2(0)-s2(1)
249   --s3_re(1) <= s2_re(0) - s2_re(1);
250   --s3_im(1) <= s2_im(0) - s2_im(1);
251   s3_re1 <= s2_re0 - s2_re1;
252   s3_im1 <= s2_im0 - s2_im1;
253   ---- s3(2)=s2(2)+s2(3)
254   --s3_re(2) <= s2_re(2) + s2_re(3);
255   --s3_im(2) <= s2_im(2) + s2_im(3);
256   s3_re2 <= s2_re2 + s2_re3;
257   s3_im2 <= s2_im2 + s2_im3;
258   ---- s3(3)=s3(2)-s2(3)
259   --s3_re(3) <= s2_re(2) - s2_re(3);
260   --s3_im(3) <= s2_im(2) - s2_im(3);
261   s3_re3 <= s2_re2 - s2_re3;
262   s3_im3 <= s2_im2 - s2_im3;
263   ---- s3(4)=s3(4)+s3(5)
264   --s3_re(4) <= s2_re(4) + s2_re(5);
265   --s3_im(4) <= s2_im(4) + s2_im(5);
266   s3_re4 <= s2_re4 + s2_re5;
267   s3_im4 <= s2_im4 + s2_im5;
268   ---- s3(5)=s3(4)-s3(5)
269   --s3_re(5) <= s2_re(4) - s2_re(5);
270   --s3_im(5) <= s2_im(4) - s2_im(5);
271   s3_re5 <= s2_re4 - s2_re5;
272   s3_im5 <= s2_im4 - s2_im5;
273   ---- s3(6)=s3(6)+s3(7)
274   --s3_re(6) <= s2_re(6) + s2_re(7);
275   --s3_im(6) <= s2_im(6) + s2_im(7);
276   s3_re6 <= s2_re6 + s2_re7;
277   s3_im6 <= s2_im6 + s2_im7;
278   ---- s3(7)=s3(6)-s3(7)
279   --s3_re(7) <= s2_re(6) - s2_re(7);
280   --s3_im(7) <= s2_im(6) - s2_im(7);
281   s3_re7 <= s2_re6 - s2_re7;
282   s3_im7 <= s2_im6 - s2_im7;
283
284   -- REORDER: DECRIBE G OUTPUT
285   ---- G(0) = s3(0)
286   G_re0 <= s3_re0;
287   G_im0 <= s3_im0;
288   ---- G(1) = s3(4)
289   G_re1 <= s3_re4;
290   G_im1 <= s3_im4;
291   ---- G(2) = s3(2)
292   G_re2 <= s3_re2;
293   G_im2 <= s3_im2;
294   ---- G(3) = s3(6)
295   G_re3 <= s3_re6;
296   G_im3 <= s3_im6;
297   ---- G(4) = s3(1)
298   G_re4 <= s3_re1;
299   G_im4 <= s3_im1;
300   ---- G(5) = s3(5)
301   G_re5 <= s3_re5;
302   G_im5 <= s3_im5;
303   ---- G(6) = s3(3)
304   G_re6 <= s3_re3;
305   G_im6 <= s3_im3;
306   ---- G(7) = s3(7)
307   G_re7 <= s3_re7;
308   G_im7 <= s3_im7;
```



```
309
310
311   end Behavioral;
312
313
```

```
1   --------------------------------------------------------------------------------
2   -- Company:
3   -- Engineer:
4   --
5   -- Create Date:   18:52:33 01/24/2012
6   -- Design Name:
7   -- Module Name:   C:/Users/WADA/Xilinx/fft_circuitA/test_fft_circuitA.vhd
8   -- Project Name:  fft_circuitA
9   -- Target Device:
10  -- Tool versions:
11  -- Description:
12  --
13  -- VHDL Test Bench Created by ISE for module: fft_circuitA
14  --
15  -- Dependencies:
16  --
17  -- Revision:
18  -- Revision 0.01 - File Created
19  -- Additional Comments:
20  --
21  -- Notes:
22  -- This testbench has been automatically generated using types std_logic and
23  -- std_logic_vector for the ports of the unit under test.  Xilinx recommends
24  -- that these types always be used for the top-level I/O of a design in order
25  -- to guarantee that the testbench will bind correctly to the post-implementation
26  -- simulation model.
27  --------------------------------------------------------------------------------
28  LIBRARY ieee;
29  USE ieee.std_logic_1164.ALL;
30  USE ieee.std_logic_signed.ALL; -- FOR CONV_INTEGER
31
32  -- Uncomment the following library declaration if using
33  -- arithmetic functions with Signed or Unsigned values
34  --USE ieee.numeric_std.ALL;
35
36  ENTITY test_fft_circuitA IS
37  END test_fft_circuitA;
38
39  ARCHITECTURE behavior OF test_fft_circuitA IS
40
41      -- Component Declaration for the Unit Under Test (UUT)
42
43      COMPONENT fft_circuitA
44      PORT(
45           s_re0 : IN  std_logic_vector(15 downto 0);
46           s_im0 : IN  std_logic_vector(15 downto 0);
47           s_re1 : IN  std_logic_vector(15 downto 0);
48           s_im1 : IN  std_logic_vector(15 downto 0);
49           s_re2 : IN  std_logic_vector(15 downto 0);
50           s_im2 : IN  std_logic_vector(15 downto 0);
51           s_re3 : IN  std_logic_vector(15 downto 0);
52           s_im3 : IN  std_logic_vector(15 downto 0);
53           s_re4 : IN  std_logic_vector(15 downto 0);
54           s_im4 : IN  std_logic_vector(15 downto 0);
55           s_re5 : IN  std_logic_vector(15 downto 0);
56           s_im5 : IN  std_logic_vector(15 downto 0);
57           s_re6 : IN  std_logic_vector(15 downto 0);
58           s_im6 : IN  std_logic_vector(15 downto 0);
59           s_re7 : IN  std_logic_vector(15 downto 0);
60           s_im7 : IN  std_logic_vector(15 downto 0);
61           G_re0 : OUT  std_logic_vector(15 downto 0);
62           G_im0 : OUT  std_logic_vector(15 downto 0);
```

テストベンチ

完成版

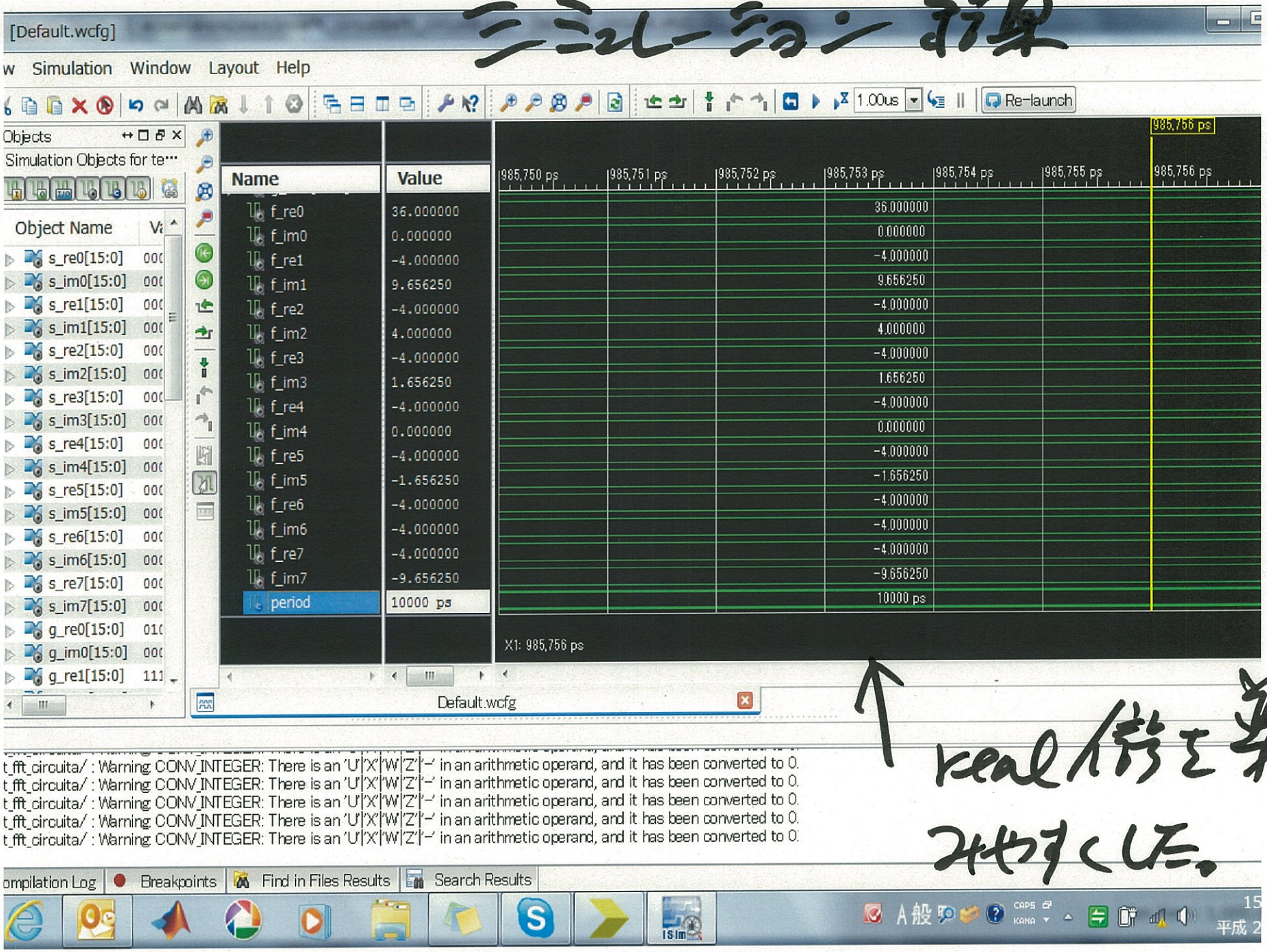
← 今回追加の

```
63              G_re1 : OUT  std_logic_vector(15 downto 0);
64              G_im1 : OUT  std_logic_vector(15 downto 0);
65              G_re2 : OUT  std_logic_vector(15 downto 0);
66              G_im2 : OUT  std_logic_vector(15 downto 0);
67              G_re3 : OUT  std_logic_vector(15 downto 0);
68              G_im3 : OUT  std_logic_vector(15 downto 0);
69              G_re4 : OUT  std_logic_vector(15 downto 0);
70              G_im4 : OUT  std_logic_vector(15 downto 0);
71              G_re5 : OUT  std_logic_vector(15 downto 0);
72              G_im5 : OUT  std_logic_vector(15 downto 0);
73              G_re6 : OUT  std_logic_vector(15 downto 0);
74              G_im6 : OUT  std_logic_vector(15 downto 0);
75              G_re7 : OUT  std_logic_vector(15 downto 0);
76              G_im7 : OUT  std_logic_vector(15 downto 0)
77            );
78        END COMPONENT;
79
80
81        --Inputs
82        signal s_re0 : std_logic_vector(15 downto 0) := (others => '0');
83        signal s_im0 : std_logic_vector(15 downto 0) := (others => '0');
84        signal s_re1 : std_logic_vector(15 downto 0) := (others => '0');
85        signal s_im1 : std_logic_vector(15 downto 0) := (others => '0');
86        signal s_re2 : std_logic_vector(15 downto 0) := (others => '0');
87        signal s_im2 : std_logic_vector(15 downto 0) := (others => '0');
88        signal s_re3 : std_logic_vector(15 downto 0) := (others => '0');
89        signal s_im3 : std_logic_vector(15 downto 0) := (others => '0');
90        signal s_re4 : std_logic_vector(15 downto 0) := (others => '0');
91        signal s_im4 : std_logic_vector(15 downto 0) := (others => '0');
92        signal s_re5 : std_logic_vector(15 downto 0) := (others => '0');
93        signal s_im5 : std_logic_vector(15 downto 0) := (others => '0');
94        signal s_re6 : std_logic_vector(15 downto 0) := (others => '0');
95        signal s_im6 : std_logic_vector(15 downto 0) := (others => '0');
96        signal s_re7 : std_logic_vector(15 downto 0) := (others => '0');
97        signal s_im7 : std_logic_vector(15 downto 0) := (others => '0');
98
99        --Outputs
100       signal G_re0 : std_logic_vector(15 downto 0);
101       signal G_im0 : std_logic_vector(15 downto 0);
102       signal G_re1 : std_logic_vector(15 downto 0);
103       signal G_im1 : std_logic_vector(15 downto 0);
104       signal G_re2 : std_logic_vector(15 downto 0);
105       signal G_im2 : std_logic_vector(15 downto 0);
106       signal G_re3 : std_logic_vector(15 downto 0);
107       signal G_im3 : std_logic_vector(15 downto 0);
108       signal G_re4 : std_logic_vector(15 downto 0);
109       signal G_im4 : std_logic_vector(15 downto 0);
110       signal G_re5 : std_logic_vector(15 downto 0);
111       signal G_im5 : std_logic_vector(15 downto 0);
112       signal G_re6 : std_logic_vector(15 downto 0);
113       signal G_im6 : std_logic_vector(15 downto 0);
114       signal G_re7 : std_logic_vector(15 downto 0);
115       signal G_im7 : std_logic_vector(15 downto 0);
116       -- No clocks detected in port list. Replace <clock> below with
117       -- appropriate port name
118
119       constant period : time := 10 ns;
120
121       -- H24/2/1
122       signal F_re0 : real;
123       signal F_im0 : real;
124       signal F_re1 : real;
```

```
125       signal F_im1 : real;
126       signal F_re2 : real;
127       signal F_im2 : real;
128       signal F_re3 : real;
129       signal F_im3 : real;
130       signal F_re4 : real;
131       signal F_im4 : real;
132       signal F_re5 : real;
133       signal F_im5 : real;
134       signal F_re6 : real;
135       signal F_im6 : real;
136       signal F_re7 : real;
137       signal F_im7 : real;
138
139    BEGIN
140
141       -- Instantiate the Unit Under Test (UUT)
142       uut: fft_circuitA PORT MAP (
143             s_re0 => s_re0,
144             s_im0 => s_im0,
145             s_re1 => s_re1,
146             s_im1 => s_im1,
147             s_re2 => s_re2,
148             s_im2 => s_im2,
149             s_re3 => s_re3,
150             s_im3 => s_im3,
151             s_re4 => s_re4,
152             s_im4 => s_im4,
153             s_re5 => s_re5,
154             s_im5 => s_im5,
155             s_re6 => s_re6,
156             s_im6 => s_im6,
157             s_re7 => s_re7,
158             s_im7 => s_im7,
159             G_re0 => G_re0,
160             G_im0 => G_im0,
161             G_re1 => G_re1,
162             G_im1 => G_im1,
163             G_re2 => G_re2,
164             G_im2 => G_im2,
165             G_re3 => G_re3,
166             G_im3 => G_im3,
167             G_re4 => G_re4,
168             G_im4 => G_im4,
169             G_re5 => G_re5,
170             G_im5 => G_im5,
171             G_re6 => G_re6,
172             G_im6 => G_im6,
173             G_re7 => G_re7,
174             G_im7 => G_im7
175          );
176
177       -- H24/2/1
178       F_re0 <= real ( CONV_INTEGER(G_re0) ) / 512.0;
179       F_im0 <= real ( CONV_INTEGER(G_im0) ) / 512.0;
180       F_re1 <= real ( CONV_INTEGER(G_re1) ) / 512.0;
181       F_im1 <= real ( CONV_INTEGER(G_im1) ) / 512.0;
182       F_re2 <= real ( CONV_INTEGER(G_re2) ) / 512.0;
183       F_im2 <= real ( CONV_INTEGER(G_im2) ) / 512.0;
184       F_re3 <= real ( CONV_INTEGER(G_re3) ) / 512.0;
185       F_im3 <= real ( CONV_INTEGER(G_im3) ) / 512.0;
186       F_re4 <= real ( CONV_INTEGER(G_re4) ) / 512.0;
```

```
187       F_im4 <= real ( CONV_INTEGER(G_im4) ) / 512.0;
188       F_re5 <= real ( CONV_INTEGER(G_re5) ) / 512.0;
189       F_im5 <= real ( CONV_INTEGER(G_im5) ) / 512.0;
190       F_re6 <= real ( CONV_INTEGER(G_re6) ) / 512.0;
191       F_im6 <= real ( CONV_INTEGER(G_im6) ) / 512.0;
192       F_re7 <= real ( CONV_INTEGER(G_re7) ) / 512.0;
193       F_im7 <= real ( CONV_INTEGER(G_im7) ) / 512.0;
194
195       -- Stimulus process
196       stim_proc: process
197       begin
198          -- hold reset state for 100 ns.
199          wait for 100 ns;
200
201          wait for period*10;
202
203          -- insert stimulus here
204    --s_re(0) <=  1.0;  s_im(0) <= 0.0;
205    --s_re(1) <=  2.0;  s_im(1) <= 0.0;
206    --s_re(2) <=  3.0;  s_im(2) <= 0.0;
207    --s_re(3) <=  4.0;  s_im(3) <= 0.0;
208    --s_re(4) <=  5.0;  s_im(4) <= 0.0;
209    --s_re(5) <=  6.0;  s_im(5) <= 0.0;
210    --s_re(6) <=  7.0;  s_im(6) <= 0.0;
211    --s_re(7) <=  8.0;  s_im(7) <= 0.0;
212    s_re0 <=  "0000001000000000";  s_im0 <= "0000000000000000";
213    s_re1 <=  "0000010000000000";  s_im1 <= "0000000000000000";
214    s_re2 <=  "0000011000000000";  s_im2 <= "0000000000000000";
215    s_re3 <=  "0000100000000000";  s_im3 <= "0000000000000000";
216    s_re4 <=  "0000101000000000";  s_im4 <= "0000000000000000";
217    s_re5 <=  "0000110000000000";  s_im5 <= "0000000000000000";
218    s_re6 <=  "0000111000000000";  s_im6 <= "0000000000000000";
219    s_re7 <=  "0001000000000000";  s_im7 <= "0000000000000000";
220
221          wait;
222       end process;
223
224    END;
225
```

シミュレーション結果



↑ real 信号を入力してみやすくした。

デバイスを XC4vfx40 にし

==

* Final Report *

Synthesisレポートの Final レポート部

==

Final Results

RTL Top Level Output File Name : fft_circuitA.ngr

Top Level Output File Name : fft_circuitA

Output Format : NGC

Optimization Goal : Speed

Keep Hierarchy : No

入力 8×2×16
出力 8×2×16

Design Statistics

\# IOs : 512

Cell Usage :

\# BELS : 2529

\# GND : 1

\# LUT2 : 866

\# MUXCY : 813

\# VCC : 1

\# XORCY : 848

\# IO Buffers : 512

\# IBUF : 256

\# OBUF : 256

\# DSPs : 6

\# DSP48 : 6 ← 乗算など

==

Device utilization summary:

Selected Device : 4vfx40ff672-10

Number of Slices:	434	out of	18624	2%
Number of 4 input LUTs:	866	out of	37248	2%
Number of IOs:	512			
Number of bonded IOBs:	512	out of	352	145% (*)
Number of DSP48s:	6	out of	48	12%

WARNING:Xst:1336 - (*) More than 100% of Device resources are used

---

Partition Resource Summary:

No Partitions were found in this design.
