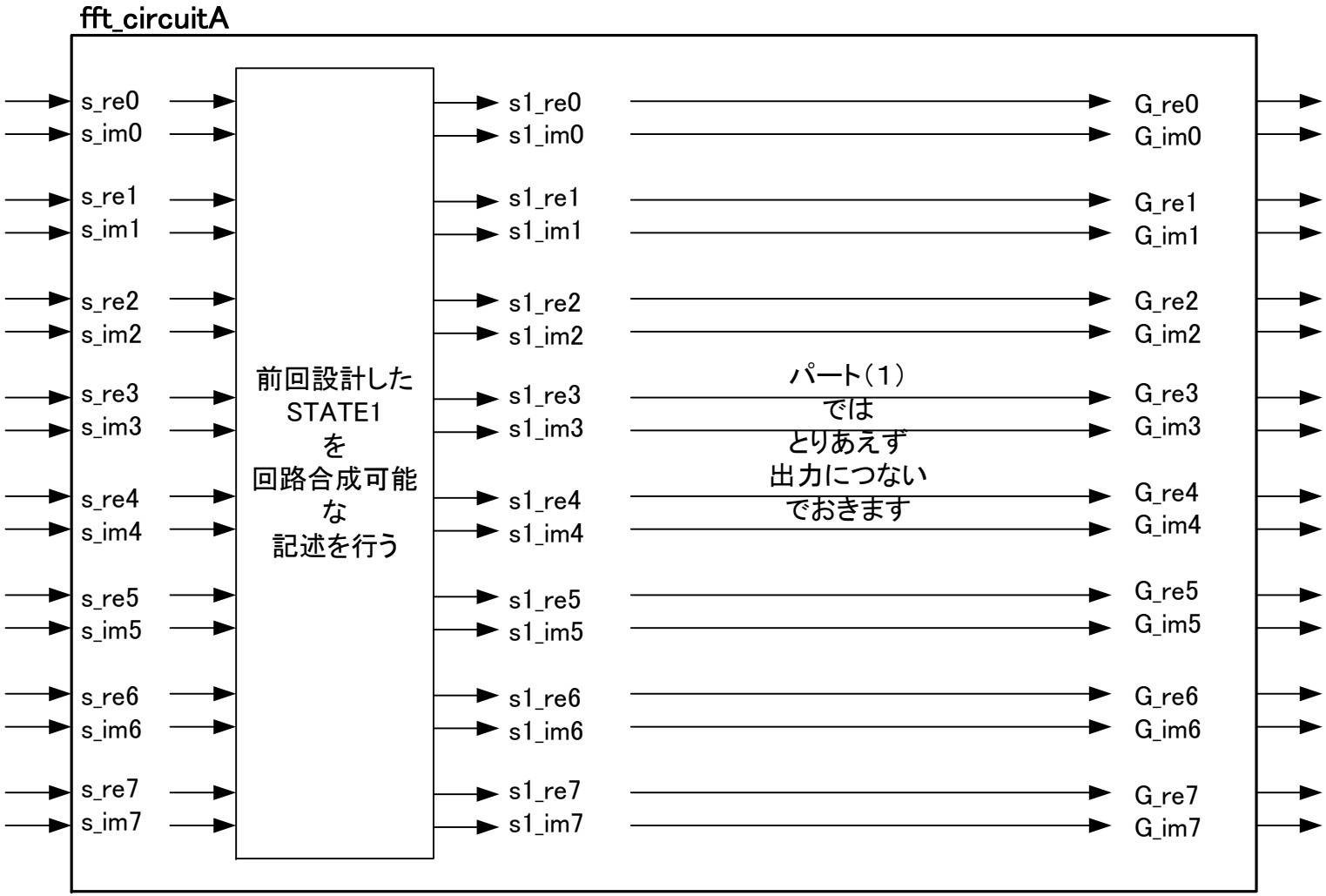
8入力FFTを　組み合わせ回路で作る

パート（1）　STAGE1までの設計

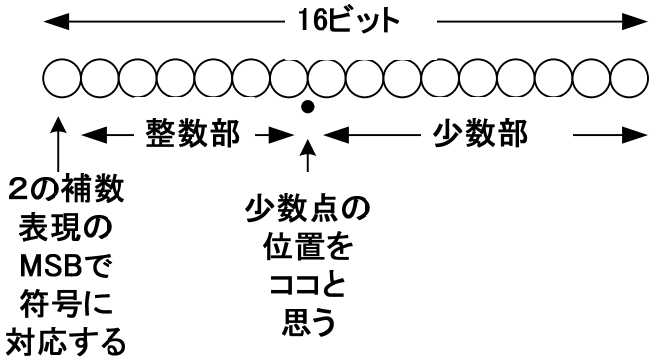


8入力FFTを　組み合わせ回路で作る

パート（2）　信号を固定小数点フォーマットで表す

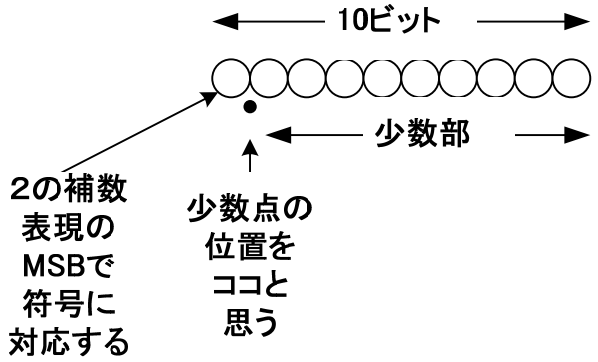
今回は、fft_circuitAで使用する信号は以下の2種類のフォーマットとする。

1）　入力信号、出力信号、中間信号s1_re1などは、
符号付き16ビット固定小数点で、小数点を以下の位置と仮定する



このフォーマットを　<16, 6, t>　と呼ぶ
16=全ビット数
6=整数ビット数
t=2の補数表現の意味　two’s compliment

2）　回転因子 w_re1などは、
符号付き10ビット固定小数点で、小数点を以下の位置と仮定する



このフォーマットを　<10, 0, t>　と呼ぶ
10=全ビット数
0=整数ビット数
t=2の補数表現の意味　two’s compliment

8入力FFTを　組み合わせ回路で作る

パート（3）　乗算の書き方

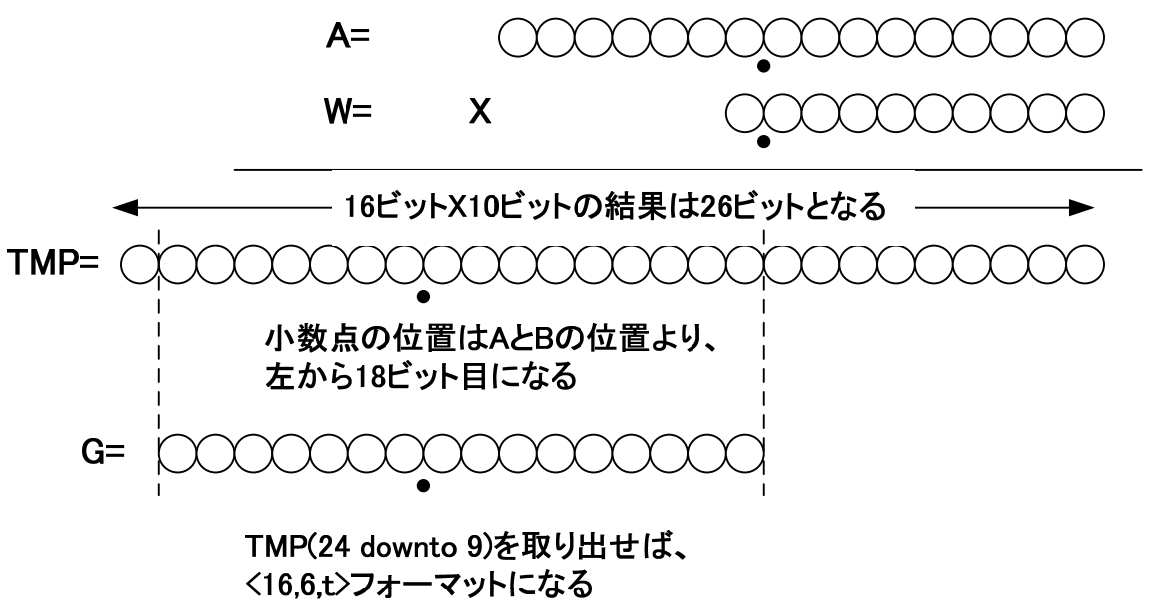
教科書P50のCOLUMN4記載のように、2の補数（符号つき数）での演算か、符号なし数での演算かで
パッケージを変えて呼び出す必要があります。
今回は、2の補数（符号つき数）での加算か乗算ですので、
std_logic_signed.all
を呼び出します。

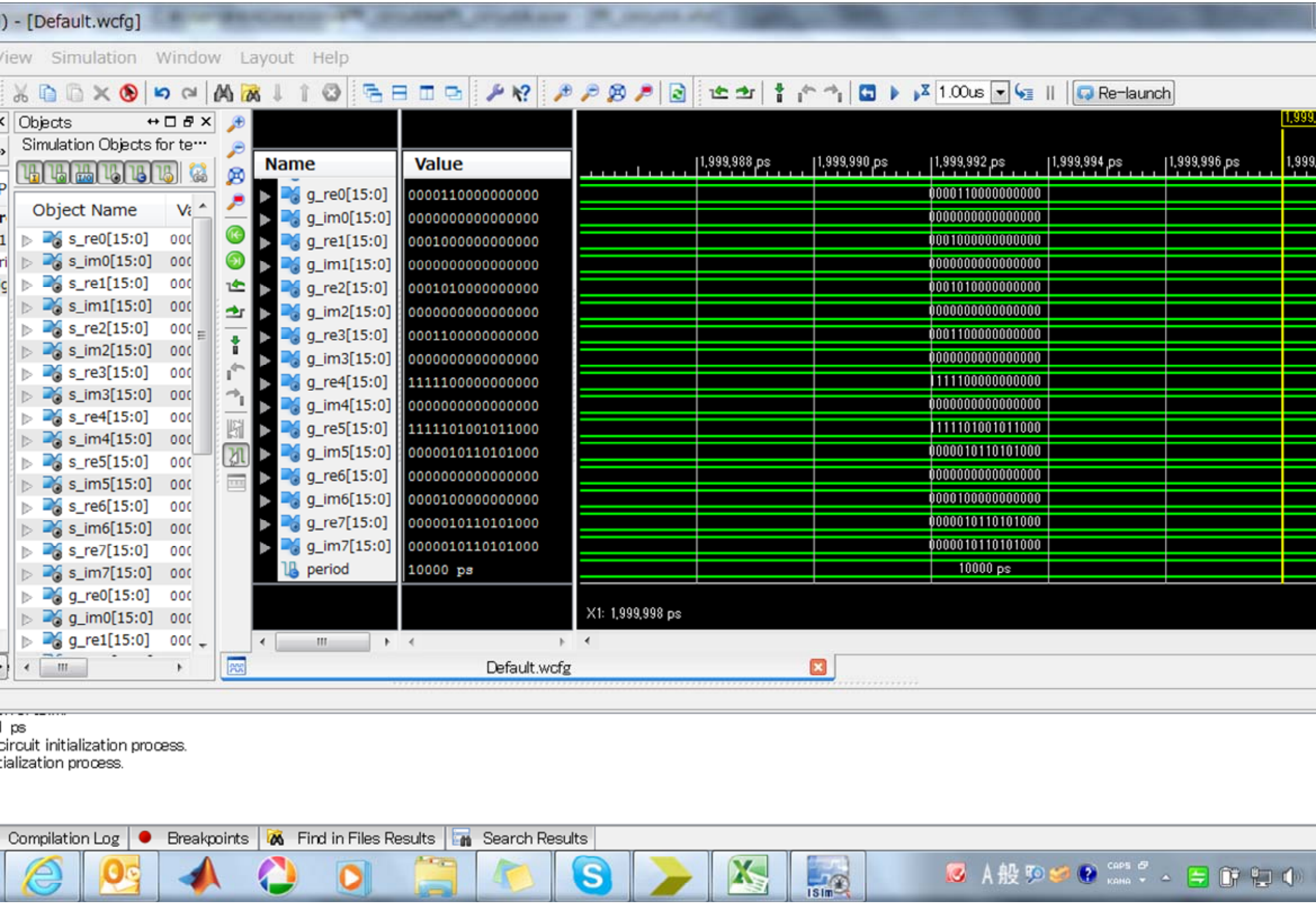
記述例

```
Library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.std_logic_signed.all;

entity FOO is
  port(
  A : in std_logic_vector(15 downto 0);  - -  <16,6,t>とする
  W : in std_logic_vector(10 downto 0); - - <10,0,t>とする
  G : out std_logic_vector(15 downto 0) );  - -  <16,6,t>とする
end FOO;
architecture RTL of FOO is
TMP: std_logic_vector(25 downto 0);
begin
TMP <= A * W;
G <= TMP(24 downto 9);
end RTL;
```

説明図




) - [Default.wcfg]
View Simulation Window Layout Help
Re-launch
Objects
Simulation Objects for te...
Object Name
s_re0[15:0]
s_im0[15:0]
s_re1[15:0]
s_im1[15:0]
s_re2[15:0]
s_im2[15:0]
s_re3[15:0]
s_im3[15:0]
s_re4[15:0]
s_im4[15:0]
s_re5[15:0]
s_im5[15:0]
s_re6[15:0]
s_im6[15:0]
s_re7[15:0]
s_im7[15:0]
g_re0[15:0]
g_im0[15:0]
g_re1[15:0]
Name
Value
g_re0[15:0] 0000110000000000
g_im0[15:0] 0000000000000000
g_re1[15:0] 0001000000000000
g_im1[15:0] 0000000000000000
g_re2[15:0] 0001010000000000
g_im2[15:0] 0000000000000000
g_re3[15:0] 0001100000000000
g_im3[15:0] 0000000000000000
g_re4[15:0] 1111100000000000
g_im4[15:0] 0000000000000000
g_re5[15:0] 1111101001011000
g_im5[15:0] 0000010110101000
g_re6[15:0] 0000000000000000
g_im6[15:0] 0000100000000000
g_re7[15:0] 0000010110101000
g_im7[15:0] 0000010110101000
period 10000 ps
1,999,988 ps
1,999,990 ps
1,999,992 ps
1,999,994 ps
1,999,996 ps
10000 ps
X1: 1,999,998 ps
Default.wcfg
ps
circuit initialization process.
ialization process.
Compilation Log
Breakpoints
Find in Files Results
Search Results

```
   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);
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    begin
97    -- STAGE1
98    -- s1(0) = s(0) + s(4)
99    --s1_re(0) <= s_re(0) + s_re(4);
100   --s1_im(0) <= s_im(0) + s_im(4);
101   s1_re0 <= s_re0 + s_re4;
102   s1_im0 <= s_im0 + s_im4;
103
104   -- s1(4) = {s(0) - s(4)}*W8**0
105   --s1_re(4) <= s_re(0) - s_re(4);
106   --s1_im(4) <= s_im(0) - s_im(4);
107   s1_re4 <= s_re0 - s_re4;
108   s1_im4 <= s_im0 - s_im4;
109
110   -- s1(1) = s(1) + s(5)
111   --s1_re(1) <= s_re(1) + s_re(5);
112   --s1_im(1) <= s_im(1) + s_im(5);
113   s1_re1 <= s_re1 + s_re5;
114   s1_im1 <= s_im1 + s_im5;
115
116   -- s1(5) = {s(1) - s(5)}*W8**1
117   --t_re     := s_re(1) - s_re(5);
118   --t_im     := s_im(1) - s_im(5);
```

```
119   --s1_re(5) <= W8_re(1) * t_re - W8_im(1) * t_im;
120   --s1_im(5) <= W8_im(1) * t_re + W8_re(1) * t_im;
121   tmp_s1_re5 <= (W8_re1 * (s_re1 - s_re5)) - (W8_im1 * (s_im1 - s_im5));
122   tmp_s1_im5 <= (W8_im1 * (s_re1 - s_re5)) + (W8_re1 * (s_im1 - s_im5));
123   s1_re5 <= tmp_s1_re5(24 downto 9);
124   s1_im5 <= tmp_s1_im5(24 downto 9);
125
126   -- s1(2) = s(2) + s(6)
127   --s1_re(2) <= s_re(2) + s_re(6);
128   --s1_im(2) <= s_im(2) + s_im(6);
129   s1_re2 <= s_re2 + s_re6;
130   s1_im2 <= s_im2 + s_im6;
131
132   -- s1(6) = {s(2) - s(6)}*W8**2
133   --W8_re(2) <=  0.0;    W8_im(2) <= -1.0;
134   --t_re     := s_re(2) - s_re(6);
135   --t_im     := s_im(2) - s_im(6);
136   --s1_re(6) <= W8_re(2) * t_re - W8_im(2) * t_im =  t_im;
137   --s1_im(6) <= W8_im(2) * t_re + W8_re(2) * t_im = -t_re;
138   s1_re6 <= s_im2 - s_im6;
139   s1_im6 <= s_re6 - s_re2;
140
141   -- s1(3) = s(3) + s(7)
142   --s1_re(3) <= s_re(3) + s_re(7);
143   --s1_im(3) <= s_im(3) + s_im(7);
144   s1_re3 <= s_re3 + s_re7;
145   s1_im3 <= s_im3 + s_im7;
146
147   -- s1(7) = {s(3) - s(7)}*W8**3
148   --W8_re(3) <= -0.7071; W8_im(3) <= -0.7071;
149   --t_re     := s_re(3) - s_re(7);
150   --t_im     := s_im(3) - s_im(7);
151   --s1_re(7) <= W8_re(3) * t_re - W8_im(3) * t_im=W8_im1 * t_re - W8_im1 * t_im;
152   --s1_im(7) <= W8_im(3) * t_re + W8_re(3) * t_im=W8_im1 * t_re + W8_im1 * t_im;
153   tmp_s1_re7 <= W8_im1 * (s_re3 - s_re7 - s_im3 + s_im7);
154   tmp_s1_im7 <= W8_im1 * (s_re3 - s_re7 + s_im3 - s_im7);
155   s1_re7 <= tmp_s1_re7(24 downto 9);
156   s1_im7 <= tmp_s1_im7(24 downto 9);
157
158   --
159   G_re0 <= s1_re0;
160   G_im0 <= s1_im0;
161   G_re1 <= s1_re1;
162   G_im1 <= s1_im1;
163   G_re2 <= s1_re2;
164   G_im2 <= s1_im2;
165   G_re3 <= s1_re3;
166   G_im3 <= s1_im3;
167   G_re4 <= s1_re4;
168   G_im4 <= s1_im4;
169   G_re5 <= s1_re5;
170   G_im5 <= s1_im5;
171   G_re6 <= s1_re6;
172   G_im6 <= s1_im6;
173   G_re7 <= s1_re7;
174   G_im7 <= s1_im7;
175
176   end Behavioral;
177
```

```
  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
 31  -- Uncomment the following library declaration if using
 32  -- arithmetic functions with Signed or Unsigned values
 33  --USE ieee.numeric_std.ALL;
 34
 35  ENTITY test_fft_circuitA IS
 36  END test_fft_circuitA;
 37
 38  ARCHITECTURE behavior OF test_fft_circuitA IS
 39
 40      -- Component Declaration for the Unit Under Test (UUT)
 41
 42      COMPONENT fft_circuitA
 43      PORT(
 44           s_re0 : IN  std_logic_vector(15 downto 0);
 45           s_im0 : IN  std_logic_vector(15 downto 0);
 46           s_re1 : IN  std_logic_vector(15 downto 0);
 47           s_im1 : IN  std_logic_vector(15 downto 0);
 48           s_re2 : IN  std_logic_vector(15 downto 0);
 49           s_im2 : IN  std_logic_vector(15 downto 0);
 50           s_re3 : IN  std_logic_vector(15 downto 0);
 51           s_im3 : IN  std_logic_vector(15 downto 0);
 52           s_re4 : IN  std_logic_vector(15 downto 0);
 53           s_im4 : IN  std_logic_vector(15 downto 0);
 54           s_re5 : IN  std_logic_vector(15 downto 0);
 55           s_im5 : IN  std_logic_vector(15 downto 0);
 56           s_re6 : IN  std_logic_vector(15 downto 0);
 57           s_im6 : IN  std_logic_vector(15 downto 0);
 58           s_re7 : IN  std_logic_vector(15 downto 0);
 59           s_im7 : IN  std_logic_vector(15 downto 0);
```

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

```
119
120   BEGIN
121
122      -- Instantiate the Unit Under Test (UUT)
123      uut: fft_circuitA PORT MAP (
124             s_re0 => s_re0,
125             s_im0 => s_im0,
126             s_re1 => s_re1,
127             s_im1 => s_im1,
128             s_re2 => s_re2,
129             s_im2 => s_im2,
130             s_re3 => s_re3,
131             s_im3 => s_im3,
132             s_re4 => s_re4,
133             s_im4 => s_im4,
134             s_re5 => s_re5,
135             s_im5 => s_im5,
136             s_re6 => s_re6,
137             s_im6 => s_im6,
138             s_re7 => s_re7,
139             s_im7 => s_im7,
140             G_re0 => G_re0,
141             G_im0 => G_im0,
142             G_re1 => G_re1,
143             G_im1 => G_im1,
144             G_re2 => G_re2,
145             G_im2 => G_im2,
146             G_re3 => G_re3,
147             G_im3 => G_im3,
148             G_re4 => G_re4,
149             G_im4 => G_im4,
150             G_re5 => G_re5,
151             G_im5 => G_im5,
152             G_re6 => G_re6,
153             G_im6 => G_im6,
154             G_re7 => G_re7,
155             G_im7 => G_im7
156           );
157
158
159      -- Stimulus process
160      stim_proc: process
161      begin
162         -- hold reset state for 100 ns.
163         wait for 100 ns;
164
165         wait for period*10;
166
167         -- insert stimulus here
168   --s_re(0) <=  1.0;  s_im(0) <= 0.0;
169   --s_re(1) <=  2.0;  s_im(1) <= 0.0;
170   --s_re(2) <=  3.0;  s_im(2) <= 0.0;
171   --s_re(3) <=  4.0;  s_im(3) <= 0.0;
172   --s_re(4) <=  5.0;  s_im(4) <= 0.0;
173   --s_re(5) <=  6.0;  s_im(5) <= 0.0;
174   --s_re(6) <=  7.0;  s_im(6) <= 0.0;
175   --s_re(7) <=  8.0;  s_im(7) <= 0.0;
176   s_re0 <=  "0000001000000000";  s_im0 <= "0000000000000000";
177   s_re1 <=  "0000010000000000";  s_im1 <= "0000000000000000";
```

```
178   s_re2 <=  "0000011000000000";  s_im2 <= "0000000000000000";
179   s_re3 <=  "0000100000000000";  s_im3 <= "0000000000000000";
180   s_re4 <=  "0000101000000000";  s_im4 <= "0000000000000000";
181   s_re5 <=  "0000110000000000";  s_im5 <= "0000000000000000";
182   s_re6 <=  "0000111000000000";  s_im6 <= "0000000000000000";
183   s_re7 <=  "0001000000000000";  s_im7 <= "0000000000000000";
184
185         wait;
186      end process;
187
188   END;
189
```