FFT の計算アルゴリズム

```
-- STAGE1
        -- s1(0)=s(0)+s(4)
        -- s1(4)=(s(0)-s(4))*W8**0
        -- s1(1)=s(1)+s(5)
        -- s1(5)=(s(1)-s(5))*W8**1
        -- s1(2)=s(2)+s(6)
        -- s1(6)=(s(2)-s(6))*W8**2
        -- s1(3)=s(3)+s(7)
        -- s1(7)=(s(3)-s(7))*W8**3
-- STAGE2:
        -- s2(0)=s1(0)+s1(2)
        -- s2(2)=(s1(0)-s1(2))*W4**0
        -- s2(1)=s1(1)+s2(3)
        -- s2(3)=(s1(1)-s2(3))*W4**1
        -- s2(4)=s1(4)+s1(6)
        -- s2(6)=(s1(4)-s1(6))*W4**0
        -- s2(5)=s1(5)+s1(7)
        -- s2(7)=(s1(5)-s1(7))*W4**1
-- STAGE3:
        -- s3(0)=s2(0)+s2(1)
        -- s3(1)=s2(0)-s2(1)
        -- s3(2)=s2(2)+s2(3)
        -- s3(3)=s3(2)-s2(3)
        -- s3(4)=s3(4)+s3(5)
        -- s3(5)=s3(4)-s3(5)
        -- s3(6)=s3(6)+s3(7)
        -- s3(7)=s3(6)-s3(7)
-- REORDER:
        -- G(0) = s3(0)
        -- G(1) = s3(4)
        -- G(2) = s3(2)
        -- G(3) = s3(6)
        -- G(4) = s3(1)
        -- G(5) = s3(5)
        -- G(6) = s3(3)
        -- G(7) = s3(7)
```

```
-- 8POINT FFT behavior description

  LIBRARY ieee;
  USE ieee.std_logic_1164.ALL;
  USE ieee.numeric_std.ALL;

  ENTITY testbench IS
  END testbench;

  ARCHITECTURE behavior OF testbench IS

          type real_array8 is array (0 to 7) of real;
          SIGNAL s_re  :  real_array8;
                        SIGNAL s_im  :  real_array8;
                        SIGNAL s1_re :  real_array8;
                        SIGNAL s1_im :  real_array8;
                        SIGNAL s2_re :  real_array8;
                        SIGNAL s2_im :  real_array8;
                        SIGNAL s3_re :  real_array8;
                        SIGNAL s3_im :  real_array8;
                        SIGNAL G_re  :  real_array8;
                        SIGNAL G_im  :  real_array8;

                        SIGNAL W8_re :  real_array8;
                        SIGNAL W8_im :  real_array8;

  BEGIN
  -- VALUE DEFINITION
  -- TWIDDLE FACTOR
  W8_re(0) <=  1.0;    W8_im(0) <=  0.0;
  W8_re(1) <=  0.7071; W8_im(1) <= -0.7071;
  W8_re(2) <=  0.0;    W8_im(2) <= -1.0;
  W8_re(3) <= -0.7071; W8_im(3) <= -0.7071;
  W8_re(4) <= -1.0;    W8_im(4) <=  0.0;
  W8_re(5) <= -0.7071; W8_im(5) <=  0.7071;
  W8_re(6) <=  0.0;    W8_im(6) <=  1.0;
  W8_re(7) <=  0.7071; W8_im(7) <=  0.7071;
```

```
  -- INPUT
  s_re(0) <=  1.0;    s_im(0) <=  0.0;
  s_re(1) <=  2.0;    s_im(1) <=  0.0;
  s_re(2) <=  3.0;    s_im(2) <=  0.0;
  s_re(3) <=  4.0;    s_im(3) <=  0.0;
  s_re(4) <=  5.0;    s_im(4) <=  0.0;
  s_re(5) <=  6.0;    s_im(5) <=  0.0;
  s_re(6) <=  7.0;    s_im(6) <=  0.0;
  s_re(7) <=  8.0;    s_im(7) <=  0.0;


  --  Test Bench Statements
     tb : PROCESS
          variable t_re : real;
          variable t_im : real;
     BEGIN

                wait for 100 ns; -- wait until global set/reset completes
-- STAGE1
                -- s1(0)=s(0)+s(4)
                s1_re(0) <= s_re(0) + s_re(4);
                s1_im(0) <= s_im(0) + s_im(4);
                -- s1(4)=(s(0)-s(4))*W8**0
                s1_re(4) <= s_re(0) - s_re(4);
                s1_im(4) <= s_im(0) - s_im(4);

                -- s1(1)=s(1)+s(5)
                s1_re(1) <= s_re(1) + s_re(5);
                s1_im(1) <= s_im(1) + s_im(5);
                -- s1(5)=(s(1)-s(5))*W8**1
                t_re     := s_re(1) - s_re(5);
                t_im     := s_im(1) - s_im(5);
                s1_re(5) <= W8_re(1) * t_re - W8_im(1) * t_im;
                s1_im(5) <= W8_im(1) * t_re + W8_re(1) * t_im;

                -- s1(2)=s(2)+s(6)
                s1_re(2) <= s_re(2) + s_re(6);
                s1_im(2) <= s_im(2) + s_im(6);
```

```
		-- s1(6)=(s(2)-s(6))*W8**2
		t_re     := s_re(2) - s_re(6);
		t_im     := s_im(2) - s_im(6);
		s1_re(6) <= W8_re(2) * t_re - W8_im(2) * t_im;
		s1_im(6) <= W8_im(2) * t_re + W8_re(2) * t_im;

		-- s1(3)=s(3)+s(7)
		s1_re(3) <= s_re(3) + s_re(7);
		s1_im(3) <= s_im(3) + s_im(7);
		-- s1(7)=(s(3)-s(7))*W8**3
		t_re     := s_re(3) - s_re(7);
		t_im     := s_im(3) - s_im(7);
		s1_re(7) <= W8_re(3) * t_re - W8_im(3) * t_im;
		s1_im(7) <= W8_im(3) * t_re + W8_re(3) * t_im;


		wait for 100 ns; -- wait until global set/reset completes
-- STAGE2: DESCRIBE STATE2 CALCULATION
		-- s2(0)=s1(0)+s1(2)
		s2_re(0) <= s1_re(0) + s1_re(2);
		s2_im(0) <= s1_im(0) + s1_im(2);
		-- s2(2)=(s1(0)-s1(2))*W4**0
		s2_re(2) <= s1_re(0) - s1_re(2);
		s2_im(2) <= s1_im(0) - s1_im(2);

		-- s2(1)=s1(1)+s2(3)
		s2_re(1) <= s1_re(1) + s1_re(3);
		s2_im(1) <= s1_im(1) + s1_im(3);
		-- s2(3)=(s1(1)-s2(3))*W4**1
		t_re     := s1_re(1) - s1_re(3);
		t_im     := s1_im(1) - s1_im(3);
		s2_re(3) <= W8_re(2) * t_re - W8_im(2) * t_im;
		s2_im(3) <= W8_im(2) * t_re + W8_re(2) * t_im;

		-- s2(4)=s1(4)+s1(6)
		s2_re(4) <= s1_re(4) + s1_re(6);
		s2_im(4) <= s1_im(4) + s1_im(6);
		-- s2(6)=(s1(4)-s1(6))*W4**0
```

```
                s2_re(6) <= s1_re(4) - s1_re(6);
                s2_im(6) <= s1_im(4) - s1_im(6);

                -- s2(5)=s1(5)+s1(7)
                s2_re(5) <= s1_re(5) + s1_re(7);
                s2_im(5) <= s1_im(5) + s1_im(7);
                -- s2(7)=(s1(5)-s1(7))*W4**1
                t_re     := s1_re(5) - s1_re(7);
                t_im     := s1_im(5) - s1_im(7);
                s2_re(7) <= W8_re(2) * t_re - W8_im(2) * t_im;
                s2_im(7) <= W8_im(2) * t_re + W8_re(2) * t_im;

                wait for 100 ns; -- wait until global set/reset completes
-- STAGE3: DESCRIBE STATE3 CALCULATION
                -- s3(0)=s2(0)+s2(1)
                s3_re(0) <= s2_re(0) + s2_re(1);
                s3_im(0) <= s2_im(0) + s2_im(1);
                -- s3(1)=s2(0)-s2(1)
                s3_re(1) <= s2_re(0) - s2_re(1);
                s3_im(1) <= s2_im(0) - s2_im(1);
                -- s3(2)=s2(2)+s2(3)
                s3_re(2) <= s2_re(2) + s2_re(3);
                s3_im(2) <= s2_im(2) + s2_im(3);
                -- s3(3)=s3(2)-s2(3)
                s3_re(3) <= s2_re(2) - s2_re(3);
                s3_im(3) <= s2_im(2) - s2_im(3);
                -- s3(4)=s3(4)+s3(5)
                s3_re(4) <= s2_re(4) + s2_re(5);
                s3_im(4) <= s2_im(4) + s2_im(5);
                -- s3(5)=s3(4)-s3(5)
                s3_re(5) <= s2_re(4) - s2_re(5);
                s3_im(5) <= s2_im(4) - s2_im(5);
                -- s3(6)=s3(6)+s3(7)
                s3_re(6) <= s2_re(6) + s2_re(7);
                s3_im(6) <= s2_im(6) + s2_im(7);
                -- s3(7)=s3(6)-s3(7)
                s3_re(7) <= s2_re(6) - s2_re(7);
                s3_im(7) <= s2_im(6) - s2_im(7);
```

```
                wait for 100 ns; -- wait until global set/reset completes
-- REORDER: DECRIBE G OUTPUT
                -- G(0) = s3(0)
                G_re(0) <= s3_re(0);
                G_im(0) <= s3_im(0);
                -- G(1) = s3(4)
                G_re(1) <= s3_re(4);
                G_im(1) <= s3_im(4);
                -- G(2) = s3(2)
                G_re(2) <= s3_re(2);
                G_im(2) <= s3_im(2);
                -- G(3) = s3(6)
                G_re(3) <= s3_re(6);
                G_im(3) <= s3_im(6);
                -- G(4) = s3(1)
                G_re(4) <= s3_re(1);
                G_im(4) <= s3_im(1);
                -- G(5) = s3(5)
                G_re(5) <= s3_re(5);
                G_im(5) <= s3_im(5);
                -- G(6) = s3(3)
                G_re(6) <= s3_re(3);
                G_im(6) <= s3_im(3);
                -- G(7) = s3(7)
                G_re(7) <= s3_re(7);
                G_im(7) <= s3_im(7);

                wait; -- will wait forever

    END PROCESS tb;
  --  End Test Bench

  END;
```