CAD　８点 FFT 関連資料

担当：和田知久　（ファイヤー和田）
所属：琉球大学　工学部　情報工学科
連絡先：wada@ie.u-ryukyu.ac.jp
Home Page: http://www.ie.u-ryukyu.ac.jp/~wada/

１）FFT 高速フーリエ変換　計算アルゴリズム

以下に、周波数間引き８点 FFT を導出する

8 点 FFT の導出

入力：s(n)
出力：G(k)

$$G(k)=\sum_{n=0}^{N-1}s(n)W_N^{nk}\cdots(2)$$

$$W_N=e^{-j\frac{2\pi}{N}}=\cos(\frac{2\pi}{N})-j\sin(\frac{2\pi}{N})\cdots(3)$$

変数変換

N=8 とし、以下の変数変換をする。

$$k=k_0+2k_1+4k_2\quad(k_0,k_1,k_2=0,1)$$
$$n=n_0+2n_1+4n_2\quad(n_0,n_1,n_2=0,1)$$

$$G(k)=\sum_{n_0=0}^{1}\sum_{n_1=0}^{1}\sum_{n_2=0}^{1}s(n_0+2n_1+4n_2)W_N^{(n_0+2n_1+4n_2)(k_0+2k_1+4k_2)}$$

となる。

（ステージ１）

n_2 に関するΣを展開する

$$G(k)=\sum_{n_0=0}^{1}\sum_{n_1=0}^{1}(s(n_0+2n_1)W_8^{(n_0+2n_1)(k_0+2k_1+4k_2)}+s(n_0+2n_1+4)W_8^{(n_0+2n_1+4)(k_0+2k_1+4k_2)})$$

$$=\sum_{n_0=0}^{1}\sum_{n_1=0}^{1}(s(n_0+2n_1)W_8^{(n_0+2n_1)k_0}W_8^{(n_0+2n_1)(2k_1+4k_2)}+s(n_0+2n_1+4)W_8^{(n_0+2n_1)k_0}W_8^{4k_0}W_8^{(n_0+2n_1)(2k_1+4k_2)}W_8^{4(2k_1+4k_2)})$$

$$=\sum_{n_0=0}^{1}\sum_{n_1=0}^{1}(s(n_0+2n_1)W_8^{(n_0+2n_1)k_0}+(-1)^{k_0}s(n_0+2n_1+4)W_8^{(n_0+2n_1)k_0})W_8^{(n_0+2n_1+4)(2k_1+4k_2)}$$

$$=\sum_{n_0=0}^{1}\sum_{n_1=0}^{1}((s(n_0+2n_1)+(-1)^{k_0}s(n_0+2n_1+4))W_8^{(n_0+2n_1)k_0})W_4^{(n_0+2n_1)(k_1+2k_2)}$$

ここで、

$$s1(n_0+2n_1+4k_0)=((s(n_0+2n_1)+(-1)^{k_0}s(n_0+2n_1+4))W_8^{(n_0+2n_1)k_0}$$

$$(k_0=0,1)$$

とすると、

$$G(k)=\sum_{n_0=0}^{1}\sum_{n_1=0}^{1}s1(n_0+2n_1+4k_0)W_4^{(n_0+2n_1)(k_1+2k_2)}$$

となり、これは、s1 を入力とする 4 点 DFT に相当します。ここで、k_0=0,1 なので２つの DFT 計算が必要となります。

（ステージ２）

同様に、n_1 に関するΣを展開する

$$G(k)=\sum_{n_0=0}^{1}(s1(n_0+0+4k_0)W_4^{(n_0)(k_1+2k_2)}+s1(n_0+2+4k_0)W_4^{(n_0+2)(k_1+2k_2)})$$

$$=\sum_{n_0=0}^{1}(s1(n_0+0+4k_0)W_4^{n_0k_1}W_4^{n_02k_2}+s1(n_0+2+4k_0)W_4^{n_0k_1}W_4^{2k_1}W_4^{n_02k_2}W_4^{4k_2})$$

$$=\sum_{n_0=0}^{1}((s1(n_0+0+4k_0)+(-1)^{k_1}s1(n_0+2+4k_0))W_4^{n_0k_1})W_2^{n_0k_2}$$

ここで

$$s2(n_0+2k_1+4k_0)=((s1(n_0+0+4k_0)+(-1)^{k_1}s1(n_0+2+4k_0))W_4^{n_0k_1}$$

$$(k_1=0,1)$$

とすると

$$G(k)=\sum_{n_0=0}^{1} s2(n_0+2k_1+4k_0)W_2^{n_0k_2}$$

となり、これは、s2 を入力とする 2 点 DFT に相当します。ここで、k_0, k_1=0,1 なので 4 つの 2 点 DFT 計算が必要となります。

（ステージ３）

同様に、n_0 に関するΣを展開する

$$G(k)=s2(0+2k_1+4k_0)+s2(1+2k_1+4k_0)W_2^{k_2}$$
$$=s2(0+2k_1+4k_0)+(-1)^{k_2}s2(1+2k_1+4k_0)$$

ここで

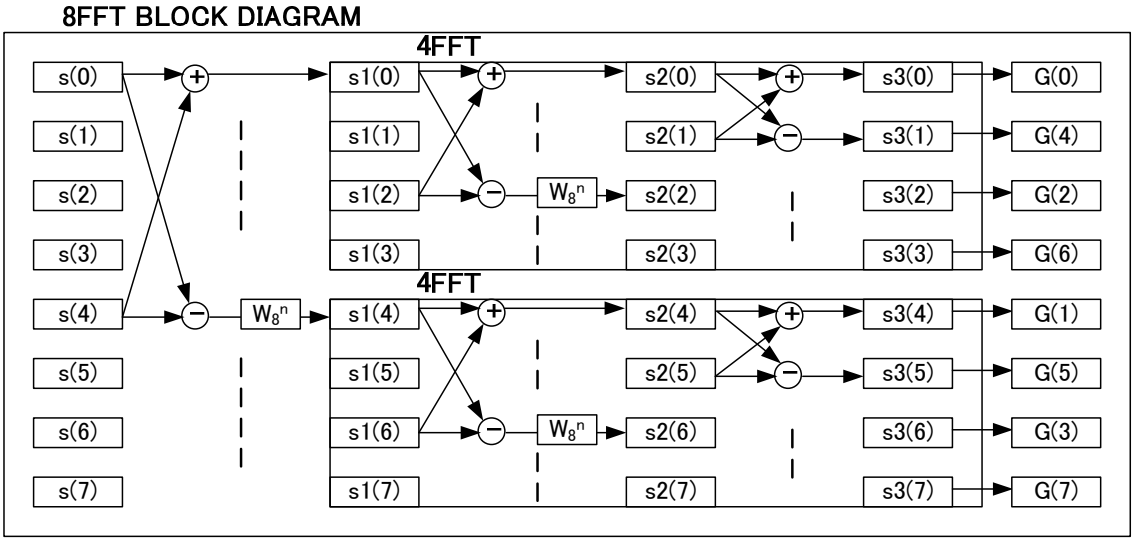
$$s3(k_2+2k_1+4k_0)=s2(0+2k_1+4k_0)+(-1)^{k_2}s2(1+2k_1+4k_0)$$

とすると、

$$G(k)=G(k_0+2k_1+4k_2)=s3(k_2+2k_1+4k_0)$$

となり、INDEX の値が変わっています。
最後に、順番を入れ替える必要があります。

３）ブロック図



CAD　８点 FFT　 2012/1/11

３）８点 FFT と８点 DFT の演算量に比較

８点 DFT

（８乗算＋７加算）＊８＝６４乗算＋５６加算

８点 FFT

バラフライ演算 1 個には、加減算 2 個、乗算 1 個とすると

ステージ１：

（1 乗算＋2 加減算）＊4＝4 乗算＋8 加減算

ステージ２：

（1 乗算＋2 加減算）＊4＝4 乗算＋8 加減算

ステージ３

（2 加減算）＊4＝8 加減算

トータル：　8 乗算＋24 加減算

以上

宿題５

講義で説明した STATE1 までの VHDL によるビヘビア計算を

STAGE2/STATE3/REORDER まで完成させ、８点 FFT の結果がでるようにせよ！

宿題４で手計算した３種の入力に対して、上記 VHDL により８点 FFT 計算を行え

宿題レポートには完成されたビヘビア VHDL コードと３つの FFT 計算の結果を示せ。

以上

```
-- 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

                wait for 100 ns; -- wait until global set/reset completes
-- STAGE3: DESCRIBE STATE3 CALCULATION

                wait for 100 ns; -- wait until global set/reset completes
-- REORDER: DECRIBE G OUTPUT

                wait; -- will wait forever

    END PROCESS tb;
  --  End Test Bench

  END;
```