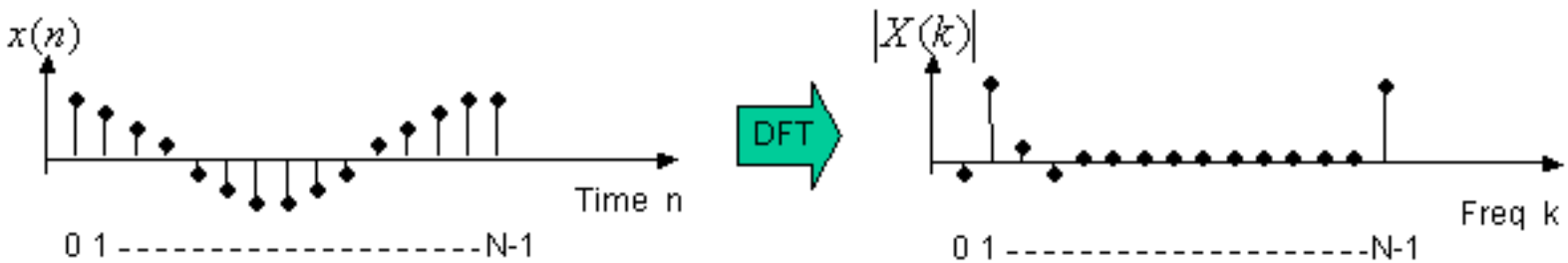
FFT設計（1）

ファイヤー和田　知久
wada@ie.u-ryukyu.ac.jp
琉球大学・工学部・情報工学科　教授
http://www.ie.u-ryukyu.ac.jp/~wada

離散フーリエ変換 (DFT)

- 離散フーリエ変換は通常スペクトル解析に使用されます。
- x(n)は解析するN点の区間で、ほぼ1周期の正弦波となっています。
- これを離散フーリエ変換 (DFT) を行うと、右の図のようにk=1に大きな成分が現れ、元信号x(n)にはk=1の周波数成分が多く含まれることがわかります。
- このような解析をスペクトラム解析と呼びます。
- よく見るとk=N-1の部分にも大きな成分があります。DFTでは、N点の点が周期的に繋がっている関係にあり、k=N-1は実はk=-1の周波数を示すことになります。
- 負の周波数とは理解が困難かも知れませんが、x(n)の1周期の波形は、K=-1の周波数とK=1の周波数成分からできていると考えることができます。



DFTとIDFT (逆変換) の式

- N点のx(n)信号をN点のX(k)信号に変換するのが、DFTであり、その逆がIDFTです。DFTを高速に演算する方式がFFTであり、IDFTを高速に演算する方法がIFFTですので、以下、DFTをすべてFFT、IDFTのことをすべてIFFTと呼ぶことにします。ただし、FFTではNは2の指数乗である制約があります。

$$DFT(FFT):$$

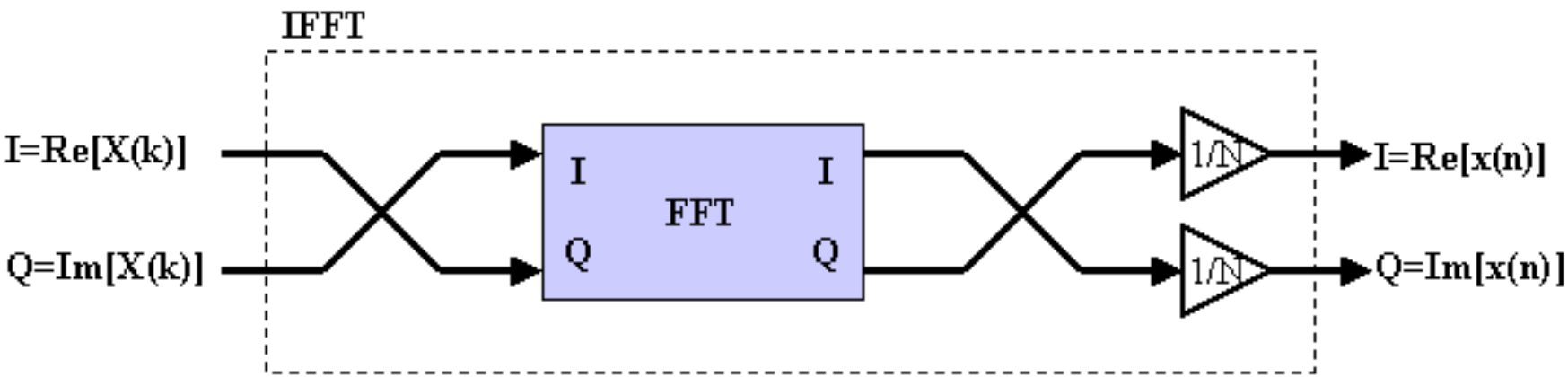
$$X(k)=\sum_{n=0}^{N-1}x(n)\cdot e^{-j\left(\frac{2\pi}{N}\right)nk}\ (k=0,1,\ldots,N-1)$$

$$IDFT(IFFT):$$

$$x(n)=\frac{1}{N}\sum_{k=0}^{N-1}X(k)\cdot e^{j\left(\frac{2\pi}{N}\right)nk}\ (n=0,1,\ldots,N-1)$$

DFTとIDFT（逆変換）の式（2）

- 2つの式をよく見ると以下の2つの違いがあります。
- 1） 最初に1/Nがあるかないか
- 2） 指数のjの前の符号
- したがって、2つの変換は非常に良く似たものであり、以下の図に示すように、FFT回路があれば、IFFTを簡単に実現することができます。



FFT演算の中身

- ここでは式をTwiddle Factor W_Nを用いて変形しています。

$$X(k) = \sum_{n=0}^{N-1} x(n) \cdot e^{-j\left(\frac{2\pi}{N}\right)nk} \quad (k = 0,1,\ldots,N-1)$$

$$W_N = e^{-j\left(\frac{2\pi}{N}\right)}$$

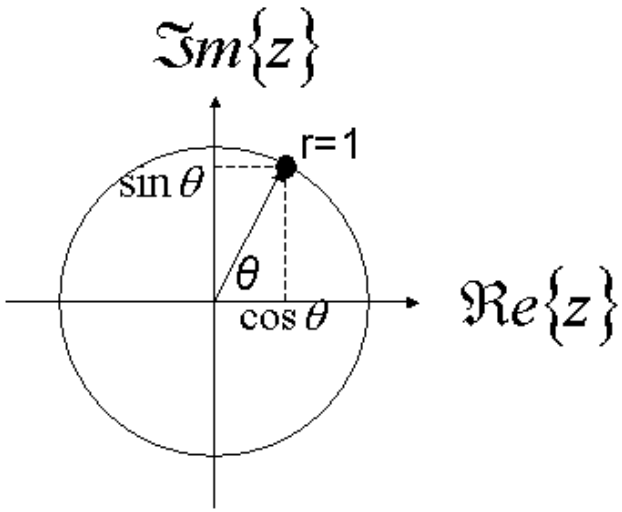
twiddle Factor!

$$X(k) = \sum_{n=0}^{N-1} x(n) \cdot W_N^{nk} \quad (k = 0,1,\ldots,N-1)$$

複素平面とオイラーの公式

- Eulerの公式より以下が成立します。これをX軸を実数(Real)軸、Y軸を虚数(Imaginary)軸とする2次元平面に示すと以下のようになります。
- すなわち、$e^{j\theta}$という関数はθを変化させることによって、半径1の円周を回転する点を示すことになります。フーリエ変換ではこの複素平面での回転を使って、波を解析します。したがって、半時計回りの回転は正の周波数に対応し、時計回りの回転は負の周波数に対応することになります。

$$e^{j\theta} = \cos\theta + j\sin\theta$$

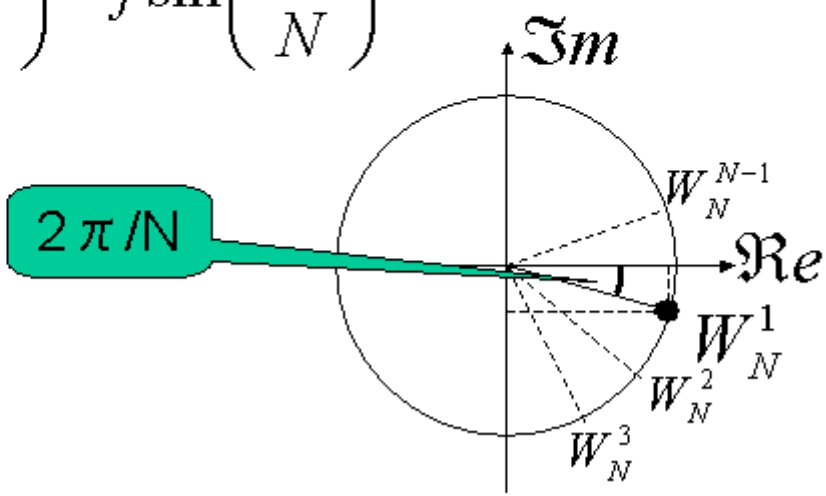


Twiddle Factor W_N^{nk}

- Twiddleとは指でぐるぐる回すという意味ですので、回転を示すものです。ここで2次元平面を用いて、このTwiddle Factor W_Nを説明します。
- したがって、Twiddle Factor W_Nを複素平面に示すと以下のようになります。実際FFTの式ではこのTwiddle Factor W_Nは指数乗されますので、指数が増加すると、図に示すように、複素平面で時計方向の回転をすることになります。

$$W_N = e^{-j\left(\frac{2\pi}{N}\right)}$$

$$= \cos\left(\frac{2\pi}{N}\right) - j\sin\left(\frac{2\pi}{N}\right)$$



固定小数点フォーマット

- 今回の課題では0.0625のような小数点以下の数をデジタル回路で取り扱う必要があります。たとえば4ビットの2進数"0111"ですが、小数点をどこの位置に置くかで表す値が変わってきます。たとえば、"01.11"ならば10進数表現では＋1.75、"0111."ならば、10進表現では＋7です。
- また、4ビットの2進数ですが、それが符号無し数すなわち正または0の数を表しているか、2の補数表現で正または負の数を表しているかを区別する必要があります。"11.10"は符号無し数であれば、10進表現で+3.50となり、2の補数表現であれば、10進表現で-0.50となります。
- すなわち、同じ4ビットの2進数でも、「小数点の位置」と「符号無し表現か2の補数表現か」を明確にしないとまったく異なった数に対応してしまいます。
- ここではSignal Processing Workbenchで用いられている固定小数点の属性表記方法を解説し、用います。
- <8,2,t>と表記すると、その信号は
- 8 ： 全体のビット数が8ビット
  2 ： 整数ビットが2ビット
  t ： two's complement ということで2の補数表現、すなわち最上位ビットは符号ビットとなる
- という意味です。したがって、"01101111"なる8ビットの数の属性が<8,2,t>とすると、

0	1	1	0	1	1	1	1
符号ビット	整数部		小数部				

- 01101111符号ビット整数部小数部ということになり、小数部は5ビットとなり、10進表現では＋3.46875に対応します。

- <8,2,u>と表記すると、その信号は
- 8 ： 全体のビット数が8ビット
  2 ： 整数ビットが2ビット
  u ： unsignedということで、符号無し数すなわち負の数を表さない
- という意味です。したがって、同じ8ビットの数"01101111"の属性が<8,2,u>とすると、

0	1	1	0	1	1	1	1
整数部		小数部					

- 01101111整数部小数部ということになり、小数部は6ビットとなり、10進表現では＋1.734375に対応します。
- 整数部のビット数が多いほど表現できる最大数すなわちレンジが大きくなり、小数部のビット数が多いほど表現できる最小数が小さくなり、解像度が向上します。

宿題5　TWIDDLE 生成回路

- **以下のENTITY仕様で64FFT対応のTWIDDLEファクター生成の組み合わせ回路を作成せよ！**（FF**は使用禁止！**）
- W_I= cos (2*π* ADDR /64)
- W_Q = -sin (2*π* ADDR/64)
- **以下にリンク**
  - http://www.ie.u-ryukyu.ac.jp/~wada/cad06/twiddle/cos.txt
  - http://www.ie.u-ryukyu.ac.jp/~wada/cad06/twiddle/msin.txt
  - http://www.ie.u-ryukyu.ac.jp/~wada/cad06/twiddle/twiddle.vhd
  - http://www.ie.u-ryukyu.ac.jp/~wada/cad06/twiddle/test_twiddle.vhd

```
library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.std_logic_arith.all;

entity TWIDDLE is
  port (
    ADDR : in  std_logic_vector(5 downto 0); -- <6,6,u>
    W_I  : out std_logic_vector(9 downto 0); -- <10,0,t>
    W_Q  : out std_logic_vector(9 downto 0)); -- <10,0,t>
end;

architecture RTL of TWIDDLE is
begin

-- WRITE YOUR CODE

end;
```

TWIDDLE動作波形例