(1) Japanese Patent Application 2014-135497

(2) Paper2014 related with JPA 2014-135497

ECT - 14 - 060

離散フーリエ変換(DFT)処理回路の設計と性能予想

梁 維 2 維 2 維 2 未 梁 準 2 未 紀 志 3 末 永 ٨ ٨ <

Design and Performance Estimation of DFT Processing Circuits Weikun Liang, Yuji Yoshida, Kishi Fukakusa, Yoshiaki Hagiwara (Sojo University) DFT is essential for voice and picture recognition. Normally DFT is processed by software, and the processing time is not negligible. This paper reports a challenge to design a DFT hardware engine circuit and estimate its performance by use of a recursive design procedure. キーワード:離散フーリエ変換、再帰的手続き、デジタル回路設計、回路性能予想、 (DFT, Recursive Procedure, Digital Circuits Design, Circuit Simulation)

(3) Paper2015 related with JPA 2014-135497

一般社団法人 電子情報通信学会 THE INSTITUTE OF ELECTRONICS, INFORMATION AND COMMUNICATION ENGINEERS

信学技報 IEICE Technical Report

非均等な時間間隔サンプリングされたデータの周波数成分 ベクトルを求める演算回路

田中 優* 梁 維焜 萩原良昭(崇城大学)

E-mail: hagiwara@cis.sojo-u.ac.jp

Digital Frequency Transformation Circuit for Time-wise Unequally Sampled Data

Masaru TANAKA, Weikun LIANG, and Yoshiaki HAGIWARA

DFT is essential for voice and picture recognition. Normally DFT is processed for Time-wise Equally Sampled Data. This paper reports a challenge to design a DFT circuit for Time-wise Unequally Sampled Data.

キーワード:離散フーリエ変換、再帰的手続き、デジタル回路設計、回路性能予想

Keywords DFT , Recursive Procedure, Digital Circuits Design, , Circuit Simulation

整理番号:H1914-01	特願2014-135497	(Proof)	提出日:平成26年 7	月 1日	<u>1/E</u>
【書類名】	特許願				
【整理番号】	H1914-01				
【あて先】	特許庁長官殿				
【国際特許分類】	G06F 17/14				
【発明者】					
【住所又は居所】	神奈川県厚木市上	荻野43	$1 \ 3 - 1$		
【氏名】	萩原 良昭				
【特許出願人】					
【住所又は居所】	神奈川県厚木市上	荻野43	$1 \ 3 - 1$		
【氏名又は名称】	萩原 良昭				
【代理人】					
【識別番号】	100154210				
(弁理士)					
【氏名又は名称】	金子宏				
【手数料の表示】					
【予納台帳番号】	606248				
【納付金額】	15,000円				
【提出物件の目録】					
【物件名】	明細書 1				
【物件名】	特許請求の範囲	1			
【物件名】	要約書 1				
【物件名】	図面 1				

【書類名】明細書

【発明の名称】時間領域データを周波数領域データに変換する演算回路

【技術分野】

[0001]

本発明は、時間領域データを周波数領域データに変換する演算回路、詳しくはデジタル フーリエ変換を行う演算回路に関する。

【背景技術】

[0002]

サンプリングされた信号の時間領域のデジタル値から、その信号の周波数特性を求める ために時間領域の値を周波数領域の値に変換するデジタルフーリエ変換(DFT)が知ら れている。

 $\begin{bmatrix} 0 & 0 & 0 & 3 \end{bmatrix}$

従来のDFTにおいては、時間領域のデータが等時間間隔でサンプリングされていた。 これを前提として、周波数領域の値(等周波数間隔のベクトル値)を時間領域の値(等時 間間隔のベクトル値)に変換する変換行列が既知であり、変換行列の逆行列を時間領域の 値を表すベクトルに作用させることで時間領域の値を周波数領域の値に変換していた。こ の際、逆行列の成分の値を時間領域の値に乗算するが、この乗算に演算時間を要していた

 $\begin{bmatrix} 0 & 0 & 0 & 4 \end{bmatrix}$

時間領域のデータが等時間間隔でサンプリングされた場合のDFTについて、高速化が 試みられていた(例えば特許文献1及び2)。しかし、(特に高周波数の情報を得るため) には)サンプル点の数が減少するものでなく、通常の演算に対して100倍以上のオーダ ーでの高速化は実現されていなかった。

【先行技術文献】

【特許文献】

[0005]

【特許文献1】特開2007-148623号公報

【特許文献2】特開2011-060177号公報

【発明の概要】

【発明が解決しようとする課題】

[0006]

本発明は、演算を高速に行うことができ、短時間でDFTを実行する演算回路を提供す ることを課題とする。

【課題を解決するための手段】

[0007]

時間領域のデータのサンプリングを不等時間間隔で行う。サンプリングの時間間隔の定 め方により、少ないサンプル点で高周波数の情報を得ることができる。

[0008]

サンプリングを不等時間間隔で行う場合における変換行列は既知のものでなく、サンプ リングに合わせて決定される。この変換行列の逆行列、又はその逆行列に2のべき乗の自 然数を乗算した行列、の成分の値(の絶対値)が、0、1又は2のべき乗となるように、 サンプリングの時間間隔を定める。ビットシフトによって乗算処理が行えるので、演算を 高速に行うことができる。

 $\begin{bmatrix} 0 & 0 & 0 & 9 \end{bmatrix}$

本発明の時間領域データを周波数領域データに変換する演算回路は、

時間領域データを周波数領域データに変換する演算回路であって、

前記時間領域データは、周期Tに亘ってN個サンプリングされた時間領域N次元ベクト ルであり、

前記N個のサンプリングは、前記周期において0からTに増加する時間 t が t = (k (n) /n) · T (n $\ln N_{max}$ 以下の自然数、k (n) $\ln b$ 互いに素である n 未満の自

前記周波数領域データは、Tを周期とする周波数ω(ω=1/T)について、ωのm倍 (mは0又はMmax以下の自然数)の周波数を表す周波数領域M次元ベクトルであり、 前記周波数領域M次元ベクトルを前記時間領域N次元ベクトルに変換する行列Hの逆行 列である行列H⁻¹を前記時間領域N次元ベクトルに作用させることで前記周波数領域M 次元ベクトルを求めることを特徴とする。

 $\begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix}$

行列H-1は、後述する自然数Qを乗算することで、その成分の値(の絶対値)が、0、1又は2のべき乗となる。演算が容易である。

ここで、N、N $_{max}$ 、M、M $_{max}$ の値は、設計によって定めればよい。すなわち、 N個のnの値、M個のmの値を任意に設計することができる。これらのn、mの最大値が それぞれN $_{max}$ 、M $_{max}$ となる。

[0011]

本発明の時間領域データを周波数領域データに変換する演算回路は、

前記k(n)の値が、nによらず全て1であることを特徴とする。

[0012]

t = 0 に近い時間に全てのサンプリングを行うものである。短時間でサンプリングを行 うことができる。

[0013]

本発明の時間領域データを周波数領域データに変換する演算回路は、

前記行列H-1を前記時間領域N次元ベクトルに作用させるに当たり、

前記行列H⁻¹の成分をQ倍(Qは、2^P(pは自然数)となるような、Nmax、Mmaxのうち大きいほうに等しい又はそれ未満の最大の自然数)した行列H'を作用させて得られるN次元ベクトルを求め、

前記行列H'の成分のうち2のべき乗の値である成分について、該成分と時間領域N次 元ベクトルの要素との乗算を、該要素の値をビットシフトすることによって行うことを特 徴とする。

[0014]

2のべき乗の値の乗算をビットシフトによって高速化する。

[0015]

本発明の時間領域データを周波数領域データに変換する演算回路は、

前記行列H'を作用させて得られるN次元ベクトルを前記Qで除算する処理を含み、 前記除算を除算前の値をビットシフトすることによって行うことを特徴とする。

[0016]

2のべき乗の値の除算もビットシフトによって高速化することができる。単純にビット シフトを行うと端数を切り捨てる演算となる。切り捨て以外の端数処理は、別途に行うこ とができる。

[0017]

本発明の時間領域データを周波数領域データに変換する演算回路は、

前記時間 t が t = (k (n) / n) • T (n $\ln \ln x$ $\ln x$ \ln

[0018]

演算対象の時間領域データを、サンプリングによって取得する。

[0019]

本発明の時間領域データを周波数領域データに変換する演算回路は、

前記サンプリング回路は、

T/Rを(Rは前記サンプリングに用いられる全てのnの最小公倍数、又はその倍数) 周期とするクロック信号発生器と、

前記時間 t が t = $(k (n) / n) \cdot T (n \downarrow N_{max} \downarrow V \neg D \land K)$ (n) $\downarrow n$)

整理番号:H1914-01 特願2014-135497 (Proof) 提出日:平成26年 7月 1日

互いに素であるn未満の自然数)の時にトリガー信号を発生するトリガー発生器と、 前記トリガー信号を受信してサンプリングを行うサンプリング器とを備えるものである ことを特徴とする。

[0020]

サンプリングを行うべきタイミングの全てに対応するクロック信号により、トリガー信 号を発生してサンプリングを行うことができる。

[0021]

本発明の時間領域データを周波数領域データに変換する演算回路は、

前記サンプリング回路は、

t=0の時にトリガー信号を発生する一次トリガー発生器と、

抵抗とコンデンサとを含み前記トリガー信号を時間遅れを持って伝達するN個のCR回路と、

前記CR回路を経た前記トリガー信号(二次トリガー)を受信してサンプリングを行う サンプリング器とを備えるものであることを特徴とする。

[0022]

サンプリングのタイミングは、クロック信号でなくCR回路によっても調整可能である

ここで「CR回路」とは、コンデンサとレジスタ(抵抗)を組み合わせて信号を遅延さ せる回路を言う。コンデンサの容量及びレジスタの抵抗値によって遅延時間を設定するこ とができる。

【発明の効果】

[0023]

本発明の時間領域データを周波数領域データに変換する演算回路は、演算を高速に行う ことができ、短時間でDFTを実行する。

【図面の簡単な説明】

[0024]

【図1】図1は、サンプリングのタイミングを示す図である。

【図2】図2は、関数hm(t)を示す図である。

【図3】図3は、逆行列H-1の例を示す図である。

【図4】図4は、時間領域データを周波数領域データに変換する演算回路の構成を示す図である。

【図5】図5は、サンプリング回路を示す図である。

【発明を実施するための形態】

[0025]

(理論的背景)

DFTにおける変換行列Hは、各行に周波数が対応し、各列にサンプリングされた点(時間)が対応する。従来の、時間均等にサンプリングを行うDFTにおいては、複素平面の単位円を均等に分割した点に相当する値によって、変換行列Hが容易に定まった。また

、変換行列Hは、複素共役であり、その逆行列H-1も容易に求めることができた。

[0026]

サンプリングを不等時間間隔で行う場合における変換行列は、以下のように求められる。いま、変換行列Hのn行m列の値をH(n,m)と書くと、H(n、m)は、m番目の周波数に対応する周期関数hm(t)に基づいて、hm(tn)で与えられる。ここで、tnは、n番目のサンプル点がサンプリングされた時間である。

[0027]

以下、時間不均等なサンプリング、及び変換行列Hを求める手順の例を、nもmも8つ の値をとる(1~8の間である)ものとして説明する。時間領域データは8次元のベクト ル(次元数N=8)、周波数領域データも8次元のベクトル(次元数M=8)である。

[0028]

図1は、サンプリングのタイミングを示す図である。基本周期Tが定められており、時

<u>整理番号:H1914-01</u>特願2014-135497 (Proof) 提出日:平成26年 7月 1日 4 間 t が 0 から T ま で の 間 の うち、 t n = T / n (n = 1、2、... 8)において サンプ リングを 行う。

 $\begin{bmatrix} 0 & 0 & 2 & 9 \end{bmatrix}$

ここで、例えば、t₇-t₈=T/56となる。DFTにおいてはサンプル点の間隔の 2倍の周期に対応する周波数成分までが求められることが通常であり、わずか8点のサン プリングによって、Tの逆数である周波数 ω に対して、その28倍の28 ω までの周波数 成分までを求めることができる。

[0030]

図2は、関数hm(t)を示す図である。hm(t)は、T/mを周期として繰り返す 関数であり、0 \leq t \leq (T/m)においては、t=0及びt=T/mの近傍において1、 t=T/2mの近傍において-1の値となり、他のtの値についてはhm(t)=0であ る。

[0031]

図においては、 $h_m(t) = 1$ または $h_m(t) = -1$ となる箇所に d だけの幅(t 軸 方向の幅)を持たせているが、幅がないものと考えることが正しい。 d $\rightarrow 0$ の極限が h_m (t)を正確に表す。

[0032]

以上に基づき、H(n, m)は以下のとおりとなる。

(1) mがnの倍数であれば、H (n, m) = h_m (t_n) = 1

(2) 2 mが n の奇数倍であれば、H (n, m) = h_m (t_n) = -1

(3) 上記以外では、H (n, m) = h_m (t_n) = 0

変換行列Hは、以下のとおりである。

【数1】

	(1	1	1	1	1	1	1	1)
	-1	1	-1	1	-1	1	-1	1
	0	0	1	0	0	1	0	0
11	0	-1	0	1	0	-1	0	1
$\Pi_8 =$	0	0	0	0	1	0	0	0
	0	0	-1	0	0	1	0	0
	0	0	0	0	0	0	1	0
	0	0	0	-1	0	0	0	1)

なお、Hの添字8は、N=M=8であることを示す。 【0033】 上記変換行列Hの逆行列H⁻¹は、以下のとおりである。

	-								
	(4	-4	-4	0	-8	4	-8	0)	
	2	2	-4	-4	0	-4	0	0	
	0	0	4	0	0	-4	0	0	
$u^{-1} - 1$	1	1	0	2	0	0	0	-4	
$H_8 = \frac{1}{8}$	0	0	0	0	8	0	0	0	
	0	0	4	0	0	4	0	0	
	0	0	0	0	0	0	8	0	
	(1	1	0	2	0	0	0	4)	

 \dot{U} 行列H⁻¹は、それを8倍(2³倍)した行列H[']の成分の絶対値が、0,1,2, 4,8のいずれかであり、全ての成分の絶対値が2のべき乗である。

[0034]

【数2】

以上、N=M=8の場合を例に説明した。以下、他の場合について簡単に説明する。図 3は、逆行列H⁻¹の例を示す図である。N=M=2、N=M=3、N=M=4の場合の ものである。逆行列H-1は、それを2倍(21倍)又は4倍(22倍)した行列H'の 成分の絶対値が、0, 1, 2, 4のいずれかである。逆行列H-1に2のべき乗の自然数 Qを乗算した行列H'の全ての成分(以下行列H'の成分を「逆行列整数成分」と言う。)の絶対値が2のべき乗であることは、N=M=8の場合と同様である。出願人は、次元 数(N及びMの値:N=Mとする)が1024までの逆行列H-1について、これを確認 した。

[0035]

自然数Qは、次元数が2のべき乗であれば次元数に等しく、次元数が2のべき乗でない 場合には2のべき乗となる次元数未満の自然数のうち最大のものである。図3において、 逆行列の先頭に示した分数の分母(以下「逆行列分母」と言う。)がQであるが、次元数 が2及び4のものはQが次元数に等しく、次元数が3のものは、2のべき乗となる3未満 の自然数のうち最大のものである2がQの値である。

[0036]

以上、n及びmが共に1~8の値を取るものとして説明したが、サンプリングを不等時 間間隔で行う場合には、より高周波数の(大きなmの)周波数領域データを求めることが 可能である。例えば、n=1、2、... 8としつつ、m=11, 12、... 18とす るなどの設計も可能である。

[0037]

また、t_n=T/nであるとして説明したが、t_nとして他の値を使用することもでき る。tnがT/nの倍数(k(n)倍とする)、k(n)がnと互いに素であるn未満の 自然数であれば、上記のH(n,m)は同一である。かかるk(n)を用いて、tn=T ・ k (n) / n としてもよい。特に、 k (n) = n - 1 とする (t_n = T (n - 1) / n とする)と、時間軸に関し、tn=T/nとする場合と対称のサンプリングとなるので、 同等のDFTが行えることが明白である。

【実施例1】

[0038]

図5は、時間領域データを周波数領域データに変換する演算回路の構成を示す図である 時間領域データを周波数領域データに変換する演算回路1は、クロック信号発生器 21、トリガー発生器22、サンプリング器23、時間領域データ保存レジスタ41、行 列演算中間データ保存レジスタ42、周波数領域データ保存レジスタ、乗算器61、加算

整理番号:H1914-01 特願2014-135497 (Proof) 提出日:平成26年 7月 1日

器62及び除算器63を備え、逆行列(整数成分)のデータ51及び逆行列(分母)のデ ータ52を保持している。

6

 $\begin{bmatrix} 0 & 0 & 3 & 9 \end{bmatrix}$

(サンプリング)

まず、サンプリングの仕組みを説明する。クロック信号発生器21、トリガー発生器2 2及びサンプリング器23が全体としてサンプリング回路を構成する。

[0040]

クロック信号発生器21は、T/Rを周期とするクロック信号を発生してトリガー発生器22に送る。Rは、サンプリングに用いられる全てのnの最小公倍数である。本実施例では、n=1,2,...8のサンプリングを行うものとする。R=840である。

[0041]

トリガー発生器22は、適宜に定める開始時刻から、105クロック、120クロック、140クロック、168クロック、210クロック、280クロック、420クロック、及び840クロック後に、トリガーを発生してサンプリング器23に送る。開始時刻においてt=0とし、 $t_n=T$ /n (n=8、7、... 1)にトリガーが送られる。

[0042]

サンプリング器23は、トリガーを受けると、アナログ信号3(例えば電圧)を測定し、A/D変換を行ったデジタル値(時間領域データ)を、時間領域データ保存レジスタ41には、nの値に対応した8つの領域があり、それぞれに $a_1 \sim a_N$ (a_8)までの時間領域データを保存する。サンプリング器23は、 a_8 、 a_7 、... a_1 の順に時間領域データを格納する。その後、再び a_8 からの格納を繰り返す。

[0043]

(演算)

次に、時間領域データ保存レジスタ41の値に対する演算を説明する。制御器(非図示)は、サンプリング器23が時間領域データ保存レジスタ41の全ての領域のデータを保存した時に、演算を開始する。

[0044]

m=1とし、逆行列整数成分51のうち1行目の値(v11、v12、・・・v18) を行列演算中間データ保存レジスタに読み込む。その後、乗算器61によって、a1とv 11、a2とv12、・・・a8とv18の乗算を行う。乗算器61を8つ設け、8つの 乗算を同時に行うことができる。

[0045]

乗算器61は、v(v11等の、行列演算中間データ保存レジスタ42に保存された値の1つをvで表す。)をa(a1等の、時間領域データ保存レジスタ41に保存された値の1つをaで表す。)に乗算する際、vの値が0であれば0を乗算結果とする。vの値が0でない場合には、vの値が負であればaの値の正負を反転させた後にvの値の絶対値(2のべき乗である)に対応してaをビットシフトさせることで乗算を実行する。例えばaの値を16ビットの整数として保存している場合、正負の反転はビットごとの0/1反転であり、ビットシフトは上位桁に向けて行い下位桁に0を補えばよい。高速の乗算が可能である。

[0046]

全ての(8つの)乗算が終了すると、加算器62が、全ての乗算結果を加算する。乗算器61と加算器62とを合わせて積和計算回路となる。

[0047]

除算器63は、加算器の出力する積和値を逆行列分母52で除算する。逆行列分母Qは、全ての計算において共通である。除算器63がデータを読み込むことでも、除算器63 に予め設定されていることでもよい。逆行列分母Qは2のべき乗であり、ビットシフトによる高速演算が可能である。なお、ビットシフトによる除算では剰余が切り捨てられるが、結果値が最大でも1だけ相違するものであり大きな問題とはならない。 [0048]

また、周波数領域データを求める目的は、周波数特性を求めることである場合が多い。 かかる場合には、すべての周波数において強度がQ倍されていても問題なく、除算を行わ ないことも考えられる。

[0049]

除算器63は、除算結果を周波数領域データ保存レジスタ43に書き込む。周波数領域 データ保存レジスタ43には、nの値に対応した8つの領域があり、それぞれに $f_1 \sim f_N$ (f_8)までの周波数領域データを保存する。除算器63は、m=1の時は f_1 に除算 結果を書き込む。以下、mの値に対応して f_m に除算結果を書き込む。

[0050]

 $f_1 \sim f_N$ (f_8)までの周波数領域データが書き込まれることで、DFTの結果が周波数領域データ保存レジスタ43に格納される。

[0051]

以上詳細に説明したように、本実施例の時間領域データを周波数領域データに変換する 演算回路は、少ない数のサンプリングで高い周波数の周波数領域データを得ることができ る。また、以下の2点において、高速化されている。(1)乗算及び除算が2のべき乗の 値で行われ、ビットシフト演算が可能である。(2)サンプリング数が少ない。行列計算 のため、処理時間はサンプリング数の2乗におおむね比例する。本実施例の時間領域デー タを周波数領域データに変換する演算回路は、tNにおけるサンプリングとtN-1にお けるサンプリングの時間間隔がおおむねN²に反比例するので、時間等間隔のサンプリン グの場合のサンプリング数に対して、おおむねその平方根の数のサンプリングとすること ができる。以上の効果により、NとMの値を20程度とする場合において、時間等間隔の サンプリングの場合の100分の1以下の時間でのDFTが可能となる。

[0052]

本実施例では、k(n)=1としたが、k(n)について他の設定も可能である。本発 明は、k(n)がnと互いに素である限り、適用できる。

【0053】

本実施例の回路は、あくまで例示である。 $t_n = (k(n) / n) \cdot T$ においてサンプ リングを行うものであれば、他の回路によっても逆行列 H^{-1} を作用させることができる

【実施例2】

[0054]

本実施例は、サンプリング回路の別形態を示すものである。演算については実施例1と 同様であり、詳細な説明を省略する。

[0055]

図5は、サンプリング回路を示す図である。実施例1と比較すると、クロック信号発生器21が設けられておらず、CR回路24が設けられている点が相違する。

[0056]

トリガー発生器22は、実施例1と相違し、t=0の時点のみでトリガー信号を発生す る。トリガー信号は、N個のCR回路24に送られる。CR回路24は、抵抗(レジスタ)とコンデンサを含み、トリガー信号を時間遅れを持ってサンプリング器23に伝達する 。コンデンサは、一端がトリガー発生器22とサンプリング器23を結ぶ回線に接続され 、他端がアースされている。遅れ時間は、コンデンサの容量及びレジスタの抵抗値によっ て設定することができる。

[0057]

N個のCR回路24のそれぞれの遅れ時間を、 $t_n = T/n$ (n = 8、7、... 1) とすることで、実施例1と同様に、サンプリング器を $t_n = T/n$ (n = 8、7、... 1) に動作させることができる。

【産業上の利用可能性】

[0058]

演算を高速に行うことができ、短時間でDFTを実行する演算回路である。DFTを必要とする多くのアプリケーションにおける利用が期待される。

- 【符号の説明】 【0059】
 - 1 時間領域データを周波数領域データに変換する演算回路
 - 21 クロック信号発生器
 - 22 トリガー発生器
 - 23 サンプリング器
 - 24 CR回路
 - 3 アナログ信号
 - 41 時間領域データ保存レジスタ
 - 42 行列演算中間データ保存レジスタ
 - 43 周波数領域データ保存レジスタ
 - 51 逆行列(整数成分)
 - 52 逆行列(分母)
 - 61 乗算器
 - 62 加算器
 - 63 除算器

【書類名】特許請求の範囲

【請求項1】

時間領域データを周波数領域データに変換する演算回路であって、

前記時間領域データは、周期Tに亘ってN個サンプリングされた時間領域N次元ベクトルであり、

前記N個のサンプリングは、前記周期において0からTに増加する時間 t が t = (k (n) / n) ・T (n k_{max} 以下の自然数、k (n) k_{n} と互いに素であるn未満の自然数)の時に行われ、

前記周波数領域データは、Tを周期とする周波数 ω ($\omega = 1 / T$)について、 ω のm倍(mは0又は M_{max} 以下の自然数)の周波数を表す周波数領域M次元ベクトルであり、

前記周波数領域M次元ベクトルを前記時間領域N次元ベクトルに変換する行列Hの逆行 列である行列H-1を前記時間領域N次元ベクトルに作用させることで前記周波数領域M 次元ベクトルを求めることを特徴とする、時間領域データを周波数領域データに変換する 演算回路。

【請求項2】

前記k(n)の値が、nによらず全て1であることを特徴とする、請求項1に記載の時 間領域データを周波数領域データに変換する演算回路。

【請求項3】

前記行列H-1を前記時間領域N次元ベクトルに作用させるに当たり、

前記行列H⁻¹の成分をQ倍(Qは、2p(pは自然数)となるような、N_{max}、M_{max}のうち大きいほうに等しい又はそれ未満の最大の自然数)した行列H'を作用させて得られるN次元ベクトルを求め、

前記行列H'の成分のうち2のべき乗の値である成分について、該成分と時間領域N次 元ベクトルの要素との乗算を、該要素の値をビットシフトすることによって行うことを特 徴とする、請求項1又は2に記載の時間領域データを周波数領域データに変換する演算回 路。

【請求項4】

前記行列H'を作用させて得られるN次元ベクトルを前記Qで除算する処理を含み、

前記除算を除算前の値をビットシフトすることによって行うことを特徴とする、請求項 3に記載の時間領域データを周波数領域データに変換する演算回路。

【請求項5】

前記時間 t が t = (k (n) / n) ・T (n $\ln \ln x$ 以下の自然数、k (n) $\ln b$ 互いに素である n 未満の自然数)の時に時間領域信号をサンプリングするサンプリング回路を備えることを特徴とする、請求項1~3のいずれか1項に記載の時間領域データを周波数領域データに変換する演算回路。

【請求項6】

前記サンプリング回路は、

T/Rを(Rは前記サンプリングに用いられる全てのnの最小公倍数、又はその倍数) 周期とするクロック信号発生器と、

前記時間 t が t = (k (n) / n) ・T (n $l N_{max}$ 以下の自然数、k (n) l nと 互いに素である n 未満の自然数)の時にトリガー信号を発生するトリガー発生器と、

前記トリガー信号を受信してサンプリングを行うサンプリング器とを備えるものである ことを特徴とする、請求項5に記載の時間領域データを周波数領域データに変換する演算 回路。

【請求項7】

前記サンプリング回路は、

t=0の時にトリガー信号を発生する一次トリガー発生器と、

抵抗とコンデンサを含み前記トリガー信号を時間遅れを持って伝達するN個のCR回路 と、

前記CR回路を経た前記トリガー信号(二次トリガー)を受信してサンプリングを行う

1

<u>整理番号:H1914-01</u> 特願2014-135497 (Proof) 提出日:平成26年 7月 1日 <u>2/E</u>

サンプリング器とを備えるものであることを特徴とする、請求項5に記載の時間領域デー タを周波数領域データに変換する演算回路。 【書類名】要約書

【要約】

【課題】演算を高速に行うことができ、短時間でDFTを実行する演算回路を提供するこ と。

【解決手段】時間領域のデータのサンプリングを不等時間間隔で行う。変換行列の逆行列 、又はその逆行列に2のべき乗の自然数を乗算した行列、の成分の値(の絶対値)が、0 、1又は2のべき乗となるように、サンプリングの時間間隔を定める。ビットシフトによ って乗算処理が行えるので、演算を高速に行うことができる。例えば、周期Tにわたって 、t=T/nのタイミングでサンプリングを行うことで、変換行列の逆行列がこの要件を 満たす。

【選択図】図4





【図5】



2014年、大学院生の修士論文指導し、研究発表

題目「離散フーリエ変換回路の設計」

と題して、2014年7月4日@島根県出雲市で、 電気学会主催の集積回路研究会にて研究発表した。

離散フーリエ変換(DFT)処理回路の設計と性能予想

梁 維焜* 吉田 侑司

深草 紀志 萩原 良昭(崇城大学)

Design and Performance Estimation of DFT Processing Circuits

Weikun Liang, Yuji Yoshida, Kishi Fukakusa ,

Yoshiaki Hagiwara (Sojo University)

DFT is essential for voice and picture recognition. Normally DFT is processed by software,

and the processing time is not negligible. This paper reports a challenge to design a DFT hardware

engine circuit and estimate its performance by use of a recursive design procedure.

キーワード:離散フーリエ変換、再帰的手続き、デジタル回路設計、回路性能予想、

(DFT, Recursive Procedure, Digital Circuits Design,, Circuit Simulation)

1. はじめに

離散フーリエ変換(DFT)処理は音声認識や画像処 理に不可欠である。通常ソフトウエアで実行され処理 時間に限界がある。自動走行車やいろいろな家庭内ア プリでは Real Time 性が不可欠で、離散フーリエ変 換処理機能を Hardware Engine として高速デジタ ル回路できないか挑戦。DFT 並列処理回路の内部の 下部階層 BLOCK 回路を再帰的手続き法を使って定 義し設計。その性能シミュレーションした。

2. 背景

現在のパソコンは 64 ビットが主流である。デジタ ル信号処理にかならず登場する高速フーリエ変換回 路でも64点の離散フーリエ変換 (DFT) 技術が実用 化され、既に IEEE802.11a/g/n 等の Wireless LAN に も応用されている。

年代	ビット幅	ビン数
1970	4 bit	40 pin
1980	8 bit	100 pin
1990	16 bit	168 pin
2000	32 bit	400 pin
2010	64 bit	1200 pin
2020	128 bit(予測)	2000 pin(予測)

表1 プロセッサーのビット幅とピン数

1970年はじめにに登場した4ビットプロセッ サーから45年近くたったが、2020年にはプロセ ッサーのビット幅とピン数がどうなるかを予想して みた(表1参照)。 2020年を目標に、人工知能機能搭載の自動走行 車や消エネを追及した電子機器や総合家庭知能シス テム(ロボットハウス)などの実用化に今後期待される。 その為には処理プロセッサーの中に基本演算ALU 回 路だけでなく、高速処理演算回路(Hardware Engine)の装備拡充が重要な課題になる。

特に、人間とのインタフェースが最重要課題となる。 その為、音声認識や画像認識技術のさらなる進歩実用 化が期待される。特に認識技術に離散フーリエ変換技 術は不可欠である。実時間情報を離散フーリエ変換に して、周波数成分ベクトル情報を高速に取得し、さら にそれを高速比較判別処理し、必要な情報を即時 (Real Time)に得ることが重要になる。



図1 プロセッサーの主要部品

本研究では、プロセッサーChip内で、離散フーリエ 変換を高速に実行する特殊用途ベクトル演算回路 (Hardware Engine)の実現を目的としている。 離散フーリエ変換(DFT)演算とは実時間情報ベクト ル f[]から周波数成分ベクトル A[]を、回転因子 行列 W[][]を使って、行列演算 A[] = W[][]f[] を実行する事である。まず2点離散フーリエ変換の場 合、P=2として、演算回路 DFT2()を次式で定義する。

 $A[0] = (1/2) \{ f[0] + f[1] \} \dots (1)$

 $A[1] = (1/2) \{ f[0] - f[1] \} \dots (2)$

この単純な DFT2()回路は(1)加算回路 add()と(2)減 算回路 sub()と(3)2つの1 bit Shift 回路 half()で構 成される。図2を参照。



図2 2 点入力 DFT 演算回路 DFT2()の定義

128bit の加算回路 add() とそれを変形した減算回路 sub()、および実行的に 1/2 の割り算を実行する 1-bit Shift 回路 half() は既に事前に定義設計が完 了し、Library Data Base に格納されているものとし ている。この独自の回路 Net List 記述方式は C 言語 の Coding 形式に類似する。演算回路の階層化がその まま C 言語の Subroutine Function の定義に対応す る。その上位回路であるこの DFT2()の定義では、 Add()などの基本回路 Module を呼び出すだけでよい 事になる。4 点離散フーリエ変換回路 DFT4()回路の 場合は図3に示す様な4 x 4 行列式の演算回路と定義 できる。DFT4()回路は、DFT2()回路を4 つコピー した DFT2(1)回路、DFT2(2)回路、DFT2(3)回路、DFT2(4) 回路を基本回路 BLOCK として構成できる。DFT4()回 路はこの行列演算を実行する複素数演算回路である。



図3 4x4の回転因子行列を使った 4 点入力の DFT4() 演算回路の定義

3. 手法(1) RADIX-4 64 点 DFT 演算の説明

64 点の離散フーリエ変換(DFT)の場合は、すでに 実用化され、既に IEEE802.11a/g/n 等の Wireless LAN にも応用されている。図4に RADIX-4 64 点 DFT 演算回路のアーキテクチャの Block 図を示す。



図4 RADIX-464 点 DFT 演算回路の Block 図

通常ASICを起こすには時間と費用がかかりすぎる ので、FPGA などで実装する場合が多い。その時によ く Shift Register を用いて、Serial 処理で実行され る。64 点 DFT の処理は図4に示すように3つの Stage による処理と信号線の配線を並びかえる Reorder 回路の合計4つの処理 Block で構成される。 この手法(1)では、最初の3つのStageを構成する 回路はまったく同じ回路を3つ使用している。

まず、 ^{№4}=1 の根を考える。この 64 個の根は 複 素数平面上の半径1の円の周辺の点である。角度を 64 等分した点 64 個で、第1根は1である。第2根をW₆₄ と書くことにする。

W⁶⁴=1;W³²=-1;W¹⁶=-i;W⁴⁸=i;....(1) である。

この RADIX-4 64 点 DFT 演算回路の場合、

 $A[M] = \sum_{N=0}^{63} W_{64}^{MN} f[N]$ for M=0 to 63;.....(4) と書くことができる。

 $\Box \Box \overline{C}, M = \mathbf{m_0} + 4 \mathbf{m_1} + 16 \mathbf{m_2}; \dots (5)$ N = $\mathbf{n_0} + 4 \mathbf{n_1} + 16 \mathbf{n_2}; \dots (6)$

とする。M と N の値の範囲はそれぞれ 0 から 63 で あるが、 (\mathbf{m}_0 , \mathbf{m}_1 , \mathbf{m}_2) と (\mathbf{n}_0 , \mathbf{n}_1 , \mathbf{n}_2) は、 (0, 1, 2, 3) の値を取るものとし、M と N の値の 範囲、0 から 63 に対応させることにする。 まず Stage One ではもとの音声情報など f(t) を実時間 (t[N], N = 0 to P) で P個 (f[N], N = 0 to P) サンプリングしたベクトル f[]を入力としている。

ただし 64 の要素(0 to 63)をもつベクトル f[]の 64 の点のうち 16 点ずつ離れた 4 点ずつを選んでいく。

R1[$n_0 + 4 n_1 + 16m_0$]

 $= W_{64}^{m_0 (n_0 + 4n_1)} \{ f[n_0 + 4n_1] \\ + (-j)^{m_0} f[n_0 + 4n_1 + 16] \\ + (-1)^{m_0} f[n_0 + 4n_1 + 32] \\ + (j)^{m_0} f[n_0 + 4n_1 + 43] \} \dots (7)$

最初に選んだ4点の組み合わせ(0,16,32,48)は R1(0)回路に入力される4点となる。 この4点は

 $(n_0, n_1) = (0, 0)$; $m_0 = 0 \sim 3$;(8) の場合に対応する。

次の4点(1,17,33,49)はR1(2)回路に入力される。最後の4点(15,31,47,63)は

(n₀, n₁)=(3,3) ; m_n =0 ~ 3 ;.....(9) に対応し、R1(16)回路に入力している。

R1(0)から R1(15)の演算回路はすべて同じ原型回路 を 16 個コピーしたものである。その原型回路は 4 x 4 の回転行列式の演算を実行する 4 点 DFT 演算回路 DFT4 0 で構成されるが、さらにその出力 4 点にそれ

ぞれ W^{m₀(n₀+4n₁)}の回転因子をかける複素数乗算

器を必要としている。こうして、最終的に中間ベクト ル値 R1[]={R1[N]; N=0 to 15}が計算される。

ただし、ここでも並び換え作業があり、R1(0)から R1(15)の各4つの出力点は最終出力ベクトルR1[]の 16 点ずつ離れた4点の値になる。

すなわち、(4)式で、最初の4点、すなわち

 $(\mathbf{n}_0, \mathbf{n}_1) = (0, 0)$; $\mathbf{m}_0 = 0 \sim 3$;(10)

に対応する出力値は R1[0],R1[16],R[32],R[48]の値 となる。最後の4点、すなわち

 $(\mathbf{n}_0, \mathbf{n}_1) = (3, 3)$; $\mathbf{m}_0 = 0 \sim 3$; (11)

に対応する出力値は R1[15], R1[31], R[47], R[63]の 値となる。

Stage Two では中間ベクトル値 R1[]を入力とし ている。ただし、64の要素(0 to 63) をもつベクトル R1[]の 64の点のうち、4 点ずつ離れた 4 点ずつを選 んでいく。

 $R2[n_{0} + 4 m_{1} + 16m_{0}]$ $= W_{16}^{m_{2} n_{0}} \{ R1[n_{0} + 16m_{0}] + (j)^{m_{2}} R1[n_{0} + 16m_{0} + 4] + (-1)^{m_{2}} R1[n_{0} + 16m_{0} + 8] + (j)^{m_{1}} R1[n_{0} + 16m_{0} + 12] \} \dots (12)$

最初に選んだ4点の組み合わせ(0,4,8,12)はR2(0) 回路の入力している。次の4点(1,5,9,13)はR2(2)回 路に入力し、随時、最後の4点(51,55,59,63)をR2(16) 回路に入力している。 R2(0)からR2(15)も4x4の DFT4 演算回路と回転因子の乗算回路で構成される。 全て16個とも全く同じ回路で構成される。こうして 中間ベクトル値 R2[]={R2[N]; N=0 to 15}が計 算される。

ただし、ここでも並び換え作業があり、R2(0)から R2(15)の各4つの出力点は最終出力ベクトルR2[]の 4点ずつ離れた4点の値になる。

Stage Three では中間ベクトル値 R2[]を入力としている。

ここでは R2[]の隣接する 4 点から出力ベクトル R3[]の隣接する 4 点の値を R3(0)から R3(15) の 16 個の回路で計算している。R3()演算回路は DFT4() 演算回路そのもので、この Stage Three では回転因子 の乗算回路は必要ない。 最終段の REORDER() 回路では、

A [$m_0 + 4 m_1 + 16 m_2$]

= R3 [$m_2 + 4 m_1 + 16m_0$](14)

となるように配線の並び換えを実行している。これは配線 並び変えだけでゲート回路を使用しない。集積回路に実装 する場合は信号配線だけですむ。論理回路規模はゼロで遅 延時間も配線遅延のみで無視できる。

これらの並び換え処理は通常バタフライ演算と言われ、FPGA などの実装では Shift Register を使って 実行処理され、たいへんめんどうな設計努力を必要と する。配線の並びかえをするために多くの Shift Register の Shift&入出力操作とその操作時間を必要 としている。しかし、半導体 Chip として集積回路化 する場合、処理プロセサの Hardware Engine として Silicon Chip に組み込むわけで、この場合ゲート回路 や論理回路は使用せず、遅延時間も無視できる。

Shift&入出力操作に必要な時間もなくなり、配線の 並びかえに必要な時間は集積回路内の信号配線の遅 延時間だけとなりほぼ無視できる時間となる。しかし 集積化する場合にはこのR1()回路を16個、R2()回路 も16個、R3()回路も16個の、合計48個のDFT2 回路相当を並列処理回路として集積した構成となり その集積回路の規模は膨大になると予想される。

4. 手法(2) 偶数奇数 2 分割 DFT 演算の説明

今迄回路規模をできるだけ実用範囲に抑えて、DFT 演算処理時間を犠牲していた。今後、半導体 Chip の 集積化が進み、今後実現可能な回路規模はさらに大き くなるものと予想される。そんな場合でも回路規模を 低減する努力はいつまでも必要となる。その1つの工 夫に、P 点の実時間の入力信号ベクトル f[] を偶数ベ クトル fe[] と奇数ベクトル fo[] に2分割、Q= P/2 bit の DFTQ() 回路で演算処理し、それぞれ Q bit の周波数成分ベクトル Ae[] と Ao[] をもとめ、さ らにそれから P bit の最終周波数成分ベクトル A[] をもとめる手法がある。その回路構成アーキテクチャ を図5に示す。



図5 偶数奇数2分割 DFT 演算回路アーキテクチャ

ここでかなりの代数演算となるが、偶数と奇数の2 分割 DFT 演算の導入式についてその詳細を説明する。

まず、0≤t<T の時間区間で P 個の Vector Data f[N], N=0,1,,,(P-1)を Sampling する。

 $\mathbf{f}[\] = \{\ \mathbf{f}[0],\ \mathbf{f}[1],\ ,\ ,\ \mathbf{f}[\mathbf{N-1}]\ \}.....(1\ 5)$

次の関係式を使って P 個の周波数成分 Vector A[M], M=0,1,,,(P-1) を計算する。

$$A[M] = (\frac{1}{n}) \sum_{N=0}^{p-1} \exp(\frac{-2\pi \eta NM}{p}) f[N] \dots (1 6)$$

ここで、Q=P/2 として、P 個の総和項を偶数項(2N) と奇数項(2N+1) にわける。

A[M]=(
$$\frac{1}{p}$$
) $\sum_{N=0}^{Q-1}$ exp($\frac{-2\pi j(2N)M}{P}$) f[2N] ←偶数項

+ $(\frac{1}{n})$ Σ_{N=0}^{Q-1} exp($\frac{-2\pi j(2N+1)M}{P}$) f[2N+1] ←奇数項

この式はつぎのように変形できる。

$$A[M] = \left(\frac{1}{2}\right) \left\{ \left(\frac{1}{Q}\right) \sum_{N=0}^{Q-1} \exp\left(\frac{-2\pi j N M}{Q}\right) f[2N] \right\}$$

+
$$\exp(\frac{-2\pi i M}{P}) \begin{pmatrix} 1 \\ Q \end{pmatrix} \sum_{N=0}^{Q-1} \exp(\frac{-2\pi i N M}{Q}) f[2N+1] \}$$

.....(18)

次の関係式を得る。

$$A[M] = \left(\frac{1}{2}\right) \left\{ Afe[M] + \exp\left(\frac{-2\pi jM}{P}\right) Afc[M] \right\}$$

......(19) ここで、Afe[M]と Afo[M]を次式で定義している。

Afe[M] =
$$\left(\frac{1}{Q}\right) \sum_{N=0}^{Q-1} \exp\left(\frac{-2\pi \eta NM}{Q}\right) f[2N] \dots (2 0)$$

Afo[M] =
$$\left(\frac{1}{Q}\right) \sum_{N=0}^{Q-1} \exp\left(\frac{-2\pi \eta NM}{Q}\right) f[2N+1]...(21)$$

さて、M=Q,(Q+1),,,,(P-1) の場合に A[M] を次式 (22)で計算しても、

 $A[M] = \left(\frac{1}{p}\right) \sum_{N=0}^{p-1} \exp\left(\frac{-2\pi \eta NM}{P}\right) f[N] \dots (2 \ 2)$

M の値を Q だけ Shift して M=0,1,,,,(Q-1) で次式 (23)でA[M] を計算しても、結果は同じである。

$$A[M+Q] = \left(\frac{1}{n}\right) \sum_{N=0}^{p-1} \exp\left(\frac{-2\pi N(M-Q)}{P}\right) f[N] \dots (2 3)$$

前回と同様に、この式を P 個の総和項を偶数項(2N) と奇数項(2N+1) にわける。

$$A[M+Q] = \left(\frac{1}{2}\right) \left(\frac{1}{2}\right) \sum_{N=0}^{Q-1} \exp\left(\frac{-2\pi i \left(2N\right) \left(Q+M\right)}{P}\right) f[2N]$$

+
$$\left(\frac{1}{2}\right)\left(\frac{1}{Q}\right)\sum_{N=0}^{Q-1} \exp\left(\frac{-2\pi j(2N+1)(Q+M)}{P}\right) f[2N+1]$$

... (2.4)

次の関係が成り立つことに注目し、

$$\exp\left(\frac{-2\pi j(2N)Q}{F}\right) = \exp\left(\frac{-2\pi jNQ}{Q}\right) = \exp(-2\pi Nj) = 1$$

... (25)

$$\exp(\frac{-2\pi i Q}{P}) = \exp(-\pi j) = -1$$
 ... (2.6)

 $\exp\left(\frac{-2\pi j(1NM + M + Q)}{P}\right) = -\exp\left(\frac{-2\pi jM}{P}\right) \exp\left(\frac{-2\pi jM}{Q}\right)$ $\dots (27)$

さらに変形する。

$$\begin{split} \mathbf{A}[\mathbf{M}+\mathbf{Q}] = & \left(\frac{1}{2}\right) \left\{ \left(\frac{1}{\mathbf{Q}}\right) \sum_{\mathbf{N}=0}^{\mathbf{Q}-1} \exp\left(\frac{-2\pi \mathbf{g}\mathbf{N}\mathbf{M}}{\mathbf{Q}}\right) \mathbf{f}[2\mathbf{N}] \\ & - \exp\left(\frac{-2\pi \mathbf{g}\mathbf{M}}{\mathbf{P}}\right) \left(\frac{1}{\mathbf{Q}}\right) \sum_{\mathbf{N}=0}^{\mathbf{Q}-1} \exp\left(\frac{-2\pi \mathbf{g}\mathbf{N}\mathbf{M}}{\mathbf{Q}}\right) \mathbf{f}[2\mathbf{N}+1] \right\} \\ & \dots (2\ 8) \end{split}$$

従ってまず、f[]を偶数項 fe]]と奇数項 fo]]にわけて Afe[M]と Afo[M]を次式で計算すると、

$$Afe[M] = \left(\frac{1}{Q}\right) \sum_{N=0}^{Q-1} \exp\left(\frac{-2\pi j N M}{Q}\right) f[2N]$$
$$Afo[M] = \left(\frac{1}{Q}\right) \sum_{N=0}^{Q-1} \exp\left(\frac{-2\pi j N M}{Q}\right) f[2N+1]$$
$$\dots (2 9)$$

M=0,1,,,(Q-1) で、A[M] と A[M+Q] を次式(30)と (31)式から計算すればいいことになる。

$$A[M] = \left(\frac{1}{2}\right) \{Afe[M] + \exp\left(\frac{-2\pi jM}{p}\right) Afe[M]\} \dots (3 0)$$

$$A[Q+M] = \binom{1}{\pi} \{Afe[M] - \exp\left(-\frac{2\pi M}{P}\right) Afe[M]\}$$

...(31)

この手法(2) で P 点の DFT 演算処理を行う場合 P を 2 の累乗個にすると有利であることが知られてい る。P 点を Q=P/2 個の偶数と奇数に 2 分割し、さらに Q 点を Q/2 の奇数と偶数項に 2 分割として、順次階 層深く押し進む。最終的に図 2 に示した、2 点の DFT 演算回路 DFT2()に行きつく。結果として演算努力は P の 2 乗回から(P/2)log2(P)回に低減できる。

5. 手法(3)本研究の演算手法の提案

呼び出された各回路 Module には add(1) のよう に()の中に番号がつく。こうすることにより上位 回路の定義でいくつもの Module 回路を呼び出し、区 別して使えることになる。この手法で、4 点 DFT 演 算回路 DFT4()を定義すると図6のようになる。



16=rom(-i); 15=12; 1= x(1)(14,16); (6,8)=DFT2(3)(11,13); (7,9)=DFT2(4)(15,1);

define DFT4() { input(2,3,4,5);output(6,7,8,9); (11,12)=DFT2(1)(2,4);(13,14)=DFT2(2)(3,5); (6,7,8,9)=DFT4A(1)(11,13,12,14);}

図6 4点 DFT 演算回路 DFT4()の定義

DFT4()回路は図3で示した行列演算を実行する複素 数演算回路である。DFT 演算では複素数を取り扱う 必要がある。基本回転因子の値はあらかじめ用意さ れ、rom()回路から求める。またかけ算器 x()も用意 されている。この独自の Coding 定義方式では内部の 配線名で 15=12 と Coding しているが。これは配線 (15)と配線(12)が同一配線である事、すなわち、2つ の呼び名をもつ事を意味する。過去の膨大な Library の設計資産を活用する場合に便利である。 さて、図6に示す4点 DFT 演算回路 DFT4()の右半分 に注目する。この右半分では、DFT2()回路を2つ呼 び出し DFT2(3)と DFT2(4)回路として呼び出し埋め込 んでいる。その部分だけをわざわざ DFT4A()回路と して定義している。DFT4()回路の全体回路に、この DFT4A()回路を1つコピーし DFT4A(1)回路とし て DFT4()回路の一部として組み込んでいる。

この手法(3)では DFT4()回路の後段部分だけを DFT4A()回路として登録する。DFT8()回路を定義 する時に DFT4()回路と DFT4A()回路の2つを構成 部品としている。手法(1)のIX-4 64点 DFT 演算 回路の場合は R1(0)から R3(15)までの、合計 48 個の 4点演算回路で構成されるが、その4点演算回路は4 点 DFT4()回路を主要部品としている。R3()回路は回 転因子の乗算器を必要としないので単純に R3()=DFT4()と Coding 内で宣言すれば R3(0)から R3(15)までの合計 16 個の4点演算回路 R3()が定義 される。演算回路 R1()と R2()に関しては

R1() = DFT4() + W1();(31)

R2() = DFT4() + W2();(3.2)

とも この Coding 定義方式では Coding 可能である。 R1(0)から R1(15) と R2(0)から R2(15)までの、合計 24 個の4点 DFT 演算回路が瞬時に定義される。



(20-27)=W8B(10-17);(40-47)=DFT8A(20-27);}

図7 8点 DFT 演算回路 DFT8()の定義

ただしこの2つの式が意味を持つ為には W1(0)~W1(15) および W2(0)~W2(15)の回転因子の乗算回路を式(7)と式(12)に従ってあらかじめ定義し

ておく必要がある。同様にこの手法(3)に従い、8
点 DFT 演算回路 DFT8()が設計できる。この Coding
定義方式を使って定義すると図7の様になる。

この手法、再帰的回路設計手続き(Recursive Circuit Design Procedure)を使って、さらに DFT16()回 路、DFT32()回路、DFT64()回路、そして DFT128() 回路も設計が可能である。回路規模が大きくなっても このように再帰的回路設計手続きを使って階層化す ることにより、シミュレーションも実装設計用の NET Coding の比較的単純に作業実行できる。

5. まとめ:3手法でSizeとDelay時間の比較

本研究の Coding 定義方式を使って、各 DFT4() か ら DFT128()までの並列演算処理回路を設計した。下 部階層 BLOCK 回路を再帰的手続きを使って定義し た。DFT64() 演算回路では3つの手法でその Size と Delay 時間を比較しその結果を表2に示した。手法 (3)でのP=4 点~128 点の場合を表3にまとめた。

表 2	DFT64()	演算回路規模と	遅延時間の比較
-----	---------	---------	---------

	手法(1)	手法(2)	手法(3)	
DFT2() 回路	192	192	192	
x() 回路	128	128	160	
rom() 回路	80	60	4	
delay(DFT2)	6	6	6	
delay(x)	2	4	5	

表3 手法(3)での DFT() 回路規模と遅延時間

P DFT2()回路 x()回路 rom()回路 delay(DFT2) delay(x)

4	4	1	0	2	1
8	12	2	1	3	2
16	32	12	2	4	3
32	80	48	3	5	4
64	192	160	4	6	5
128	448	480	5	7	6

手法(3)による DFT64()回路設計では回転因子を呼び 出す rom()回路は(W₂, W₁₆, W₃₂, W₃₄)の4つでよ い事がわかった。

文 献

 64 Point FFT Design Manual,64点高速フーリエ変換回路設計仕様 書,和田知久、Design Wave Magazine 2006年11月号、pp.143-156

 (2) 数値計算法、三井田惇郎,須田 宇宙共著、森北出版、第8章離散フ ーリエ変換、pp.96-110. 2015年、大学院生の修士論文指導し、研究発表

題目「非均等な時間間隔サンプリング されたデータの周波数成分 ベクトルを求める演算回路」

2015年8月24日~25日@熊本県熊本市の 熊本市民会館およびくまもと県民交流館パレアにて、 電子情報通信学会主催の集積回路 (ICD)研究会にて 現地主催幹事として奉仕しつつ、研究発表した。

非均等な時間間隔サンプリングされたデータの周波数成分 ベクトルを求める演算回路

田中 優* 梁 維焜 萩原良昭(崇城大学)

E-mail: hagiwara@cis.sojo-u.ac.jp

Digital Frequency Transformation Circuit for Time-wise Unequally Sampled Data

Masaru TANAKA, Weikun LIANG, and Yoshiaki HAGIWARA

DFT is essential for voice and picture recognition. Normally DFT is processed for Time-wise Equally Sampled Data. This paper reports a challenge to design a DFT circuit for Time-wise Unequally Sampled Data.

キーワード:離散フーリエ変換、再帰的手続き、デジタル回路設計、回路性能予想

Keywords DFT, Recursive Procedure, Digital Circuits Design,, Circuit Simulation

1. はじめに、

従来の離散フーリエ変換回路では均等な時間間隔でア ナログ情報を複数個サンプルし、その時間軸ベクトル を求め、次に回転因子行列と言われるもので行列演算 処理して周波数成分ベクトルを求めている。しかし、非 均等な時間間隔で、複数個サンプリングした場合にも 同様に周波数成分ベクトルを求める事ができ、その結 果回転因子行列に相当する行列式がある事を報告す る。その演算処理回路の長所と短所についても、従来の 離散フーリエ変換回路と対比して、報告する。

2. 離散フーリエ変換 (DFT) の定義

離散フーリエ変換 (DFT=Discrete Fourier Transformation)演算とは、ある一定の時間 区間 T を等間隔に N 区分し、区分した各離散 時間点ベクトル t[]の時点で、アナログ入 力信号 a(t)を sampling して、アナログ信号 ベクトル a[]を求めて、そのベクトル a[] と、回転因子行列といわれる行列式 W[][] とで、かけ算を実行し、周波数成分ベクトル A[]=W[][]a[]を求めることをいいます。 周波数成分ベクトル A[]の各要素値 A[m]は、 次の行列式演算をすることにより求まりま す。m の値は 1 から N 迄の整数とします。

$$A[m] = \sum_{n=1}^{N} W[m][n]a[n] \dots \dots (1)$$

ここで、この回転因子行列 W[][]の各要素値 W[m][n]は次式で与えられます。

$$W[m][n] = \left(\frac{1}{N}\right) \exp(-2\pi m n j/N) \dots \dots (2)$$

また、この周波数成分ベクトルA[]の値から、 もとのアナログ信号ベクトルa[]の値を、この回 転行列式 W[][]の逆行列 invW[][]を求めるこ とにより、次式の様に求まります。

$$a[n] = \sum_{m=1}^{N} invW[n][m]A[m].....(3)$$

ここで、この逆回転因子行列 invW[][]の各 要素値 invW[n][m]は次式で与えられます。

 $invW[n][m] = exp(2\pi nmj/N) \dots \dots (4)$

3. 2 点離散フーリエ変換回路 DFT2()の定義

まず2点離散フーリエ変換の場合、N=2 とします と、演算回路DFT2()を次式で定義できます。

$$A[1] = (1/2) \{ -a[1] + a[2] \} \dots (5)$$

$$A[2] = (1/2) \{ a[1] + a[2] \} \dots (6)$$

この単純な DFT2()回路は、図1の様に、加算 回路 ADD()1個と減算回路 SUB()1個と2個の 1 bit Shift 回路 Half()で構成されまず。



図1 2点入力 DFT 演算回路 DFT2()の定義

4. DFT4()回路の定義

4 点離散フーリエ変換回路 DFT4()回路は、回転因子 行列の定義式(1)と(2)により、図2に示す様な4 x 4 行列式の演算回路と定義できます。



図2 4x4の回転因子行列を使った 4点入力のDFT4()演算回路の定義

図3に示す様に、DFT4()回路は、DFT2()回路を4 つコピーしたDFT2(1)回路、DFT2(2)回路、DFT2(3)回路、 DFT2(4)回路を基本回路 BLOCK として構成できます。 DFT4()回路はこの行列演算を実行する複素数演算回路 です。ここで、回転因子ベクトル W[]と Bo[]ベクトル は次式で定義しています。For m=1→Nの値で、

$$W[m] = \exp\left(-\frac{2\pi jm}{N}\right)...(7)$$

 $Bo[m] = W[m]Ao[m] = \exp\left(-\frac{2 \pi jm}{N}\right)Ao[m] \dots (8)$

ここで、WN()回路を、このN個の複素数のかけ算 を実行する演算回路と定義します。しかし、実際には、 奇数項のみに適応されるもので、DFT4()回路の場合 は、W[1]=-jとW[2]=1のみが、図3のDFT4()回路 の中の複素数かけ算回路W2(1)回路の入力となります。 実際には、Bo[2]=Ao[2]のままで、Bo[1]= -jAo[1]とな ります。



図3 4 点入力 DFT 演算回路 DFT4()回路

実際には、次の関係が成り立ちます。

 $Ae[1] = (1/2) \{ -ae[1] + ae[2] \} \qquad \dots \qquad (9)$

 $Ae[2] = (1/2) \{ ae[1] + ae[2] \} \dots \dots (10)$

 $Ao[1] = (1/2) \{ -ao[1] + ao[2] \} \dots (11)$

 $Ao[2] = (1/2) \{ ao[1] + ao[2] \} \dots (12)$

すなわち、ae[1]=a[2], ae[2]=a[4], ao[1]=a[1], ao[2]=a[3]で あるので、次式を得ます。

 $Ae[1] = (1/2) \{ -a[2] + a[4] \} \dots \dots (13)$

 $Ae[2] = (1/2) \{ a [2] + a[4] \} \dots (14)$

 $Ao[1] = (1/2) \{ -a [1] + a[3] \} \dots (15)$

 $Ao[2] = (1/2) \{ a[1] + a[3] \} \dots (16)$

また、図2に示した回転因子行列の値により、次式の関係を 得ます。

$$A[1]=(1/2) \{ -Bo[1] + Ae[1] \} \cdots (17)$$
$$A[3]=(1/2) \{ Bo[1] + Ae[1] \} \cdots (18)$$

$$A[2]=(1/2) \{ -Bo[2] + Ae[2] \} \cdots (19)$$

 $A[4]=(1/2) \{ Bo[2] + Ae[2] \} \cdots (20)$

従って、図3において、DFT2(3)回路は式(17)と(18)を演算 実行する演算回路と考えられます。また、DFT2(4)回路は式 (19)と(20)を演算実行する演算回路と考えられます。

5. 奇数偶数分割法による DFT2N()回路の定義

同様に、8 点離散フーリエ変換回路 DFT8()回路、16 点離散フーリエ変換回路 DFT16()回路と、順次に設計 構築していきたいところです。実際、64 点の離散フー リエ変換(DFT)の場合は、すでに実用化され、既に IEEE802.11a/g/n 等の Wireless LAN にも応用されて います。現在 64bit のパソコンが主流ですが、2020 年 には 128bit パソコンの出現が期待され、離散フーリエ 変換回路も DFT128()の設計構築されることになると 期待します。

そこで、奇数偶数分割法という方法を使って、DFT 演算回路のサンプリング数を2のべき乗の数2^Lとし て、再帰的(recursive)に階層定義することにします。

たとえば、 $2^{L}=2N$ 点の実時間の入力信号ベクトルa[]を処理する DFT2N()回路を考えます。まず、この2N個のサンプリング点を持つ入力ベクトルa[]を、偶数ベクトルae[]と奇数ベクトルao[]に2分割します。

そして、それぞれを、図4の様に2つの DFTN()回 路で処理します。N bit の2つ DFTN()回路で演算処 理し、それぞれ、N bit の周波数成分ベクトル Ae[] と Ao[]を求めます。

さらにその2つのベクトルをN個のDFT2()回路を 使って、最終的に2N個の要素値を持つA[]ベクトルを 演算処理して求める回路になります。ここでN個の DFT2()回路の前段でAo[]信号には回転因子ベクトル の重みづけ演算WN()回路が必要となります。

実際には、回転行列式の定義(1)と(2)から、導入計算 は複雑ですが、最終的に次の関係が成り立ちます。

For m=1→N の値で、

$$A[m] = \left(\frac{1}{2}\right) \left\{ Ae[m] + exp\left(-\frac{2\pi jm}{N}\right) Ao[m] \right\} \dots \dots (21)$$

$$A[m+N] = \left(\frac{1}{2}\right) \left\{ -Ae[m] + \exp\left(-\frac{2\pi jm}{N}\right) Ao[m] \right\} \dots (22)$$

ここで Ao[m] と Ae[m]を次式で定義しています。
Ao[m] = $\left(\frac{1}{N}\right) \sum_{n=1}^{N} \exp\left(-\frac{2nm\pi j}{N}\right) a[2n-1] \dots (23)$
Ae[m] = $\left(\frac{1}{N}\right) \sum_{n=1}^{N} \exp\left(-\frac{2nm\pi j}{N}\right) a[2n] \dots (24)$

式(23)と式(24)の演算は、DFTN()回路を2つ使うことにより演算処理が可能です。

また、式(22)と式(1)の演算は、DFT2()回路をN個用 意することにより演算実行が可能です。

図 4 では N 個の DFT2()回路をまとめて DFT2xN() 回路として表記しています。



図4 偶数奇数2分割 DFT2N() 演算回路アーキテクチャ

ここで、回転因子ベクトル W[]と Bo[]ベクトルは式 (7)と(8)で定義しています。

WN()回路は N 個の複素数のかけ算演算回路です。 離散フーリエ変換回路の最大の問題点はこの複素数の かけ算回路 WN()回路の存在です。まずこの回転因子 ベクトル W[]の値を、特別に ROM の形などにして保 管しておき、それを読み出しかけ算する必要がありま す。

6. 非等間隔での Sampling 手法の提案

従来のDFT 変換は等間隔の離散時間点でのアナ ログ入力信号の sampling を行います。ここで発想 を転換して、等間隔で Sampling しなくても、つま り非等間隔の Sampling でも、有効な周波数成分ベ クトル A[]は求められないかと考えてみます。 そこでまず、sampling 開始時間 t \sim 0 の近 辺に多くの Sampling する離散時間点を持たせて みます。

 まず、アナログ入力信号の高速 sampling を可能にするパルス(sampling pulse)を発生 する回路が必要になります。

(2)次に当然、高速にアナログ入力信号を A/D 変換して 2 値ベクトル情報に変換する回路や、 (3)その前処理をするデジタル回路に、

(4) 重要な情報を抽出する回路と、

(5)その抽出情報を判定する回路や、

(6)出力保存する回路などが必要となります。

そこで、非等間隔 sampling 手法は自由度が 大きすぎますので、単純な非等間隔 sampling 手法として、時間区間[0, T]の間で、離散 時間点ベクトルt[]を次の簡単な式で定義さ れる場合における非等間隔 sampling 手法を 考えてみます。n の値は1から N 迄の整数と します。

$$t[n] = \frac{T}{(N+1-n)} \quad \text{for} \quad n = 1 \rightarrow N \quad \dots (25)$$

図5は N=8 の場合の sampling の timing を 示す図です。時間区間[0,T]の間で、合計 N 個 の離散時間点アナログ入力信号を sampling することにします。



図5 非等間隔 sampling 手法(N=8)

これでも t = 0 に近い時刻で、比較的多数 のアナログ入力信号 a(t)を sampling してい ます。この手法を適当に t=0 付近高密度 sampling 手法と呼ぶことにします。システム が sampling を開始しなさいと命令した時刻 を t=0 としています。

たとえば、標本点が N=8 の場合は、離散時間 点ベクトル t[]の値は以下に様になります。

$$t[] = \left\{ \frac{T}{8}, \frac{T}{7}, \frac{T}{6}, \frac{T}{5}, \frac{T}{4}, \frac{T}{3}, \frac{T}{2}, T \right\} \dots (26)$$

ここで、例えば、t[7] - t[8]=($\frac{T}{56}$) となり
ます。

非等間隔 Sampling 手法での周波数 変換行列 H[][]の定義

この非等間隔 sampling 手法での周波数変 換行列 H[][]が実際に存在します。その値 を図 6 に示します。



図 6 行列式 H[][]と逆行列 invH[][] 従来の DFT 変換の W[][]行列、すなわち

 $W[m][n] = \left(\frac{1}{N}\right) exp(-2\pi mnj/N)$ に対応するのが、 invH[][]行列になります。

すなわち、A[]=W[][]a[]に対応して A[]=invH[][]a[]となります。

また、invW[n][m] = exp(2*πmnj/N*) に対応 するのが、H[][]になります。

すなわち、a[]=invW[][]A[]に対応して a[]= H[][]A[]となります。

A[]従来の DFT 変換では、最終的に、

 $a(t) \sim b(t) = \sum_{m=1}^{N} A[m] \exp\left(\frac{2\pi m t j}{T}\right) \cdots (27)$

の形で、アナログ入力関数*a*(*t*)を近似関数 *b*(*t*)で近似できるものとします。

8. 非等間隔 Sampling 手法での

基本構成関数h_m(t)の定義

そこで同様に、ゼロ近辺高密度 sampling 手法で も、 $b(t) = \sum_{m=1}^{N} A[m] h_m(t)$ の形で、アナログ入 力関数 a(t)を近似できる関数群 $h_m(t)$ なるものを 考えます。a[]=H[][]A[]に対応させます。こ $の基本構成関数<math>h_m(t)$ の値は次式で定義できます。 その値は、単純に -1 か 0 か 1 の pulse 周期関 数 になります。

【1】
$$\left(\frac{T}{m}\right)$$
 を周期として繰り返す関数であり、

 $[2] \quad 0 \le t \le \left(\frac{T}{m}\right) \text{ kanch,}$

t=0及び $t=(\frac{T}{m})$ の近傍において1の値を取る。

- 【3】 $t = (\frac{T}{2m})$ の近傍において-1の値となり、
- 【4】他の t の値については $h_m(t) = 0$ となります。

図7に関数 $h_m(t)$ を定義する図を示します。



図7 関数h_m(t)の定義図

図 6 においては、 $h_m(t) = 1$ または $h_m(t) = -1$ となる箇所に d だけの幅 (t 軸方向の幅) を持たせていますが、幅がないものと考える ことが正しいです。 $d \rightarrow 0$ の極限が $h_m(t)$ を正確 に表すものとします。

図8にN=8の場合のt=t[n]の場合に求まる関数 $h_m(t)$ の値、すなわち、H[n][m]= $h_m(t[n])$ の値を示 します。



図8 N=8の場合のH[n][m]=h_m(t[n])の値

DFT 変換やコサイン変換やサイン変換の場 合、必ず三角関数の計算が必要でその演算処 理時間は無視できません。できれば ROM 回路 も不要な簡単な演算処理で済めば最高です。

図9は、時間領域 data を周波数領域 data に 変換する演算回路の構成図です。



 図 9 時間領域 data を周波数領域 data に 変換する演算回路の構成図

また図 10 は、sampling 回路を示す図です。



図 10 sampling 回路の block 図

図9の中での各部品番号の説明は以下のよう になります。

- 時間領域 data を周波数領域 data に 変換する演算回路全体
- 3 = analog 信号
- 21= clock 信号発生器
- 22= trigger 発生器
- 23= sampling 器
- 24= CR 回路
- 41= 時間領域 data 保存 register
- 42= 行列演算中間ベクトル data Vm[]の一時保存用 register
- 43= 周波数領域ベクトル data f[](=A[]) の一時保存用 register

- 51= 逆行列 invH[][]の要素数の分子 (整数成分)の記憶用回路
- 52= 逆行列 invH[][]の要素数の 共通分母の記憶用回路
- 53= 乗算器
- 54= 加算器
- 55= 除算器

実際に a(t)=cos(4 π t/T)の場合で,従来の DFT 変換で求めて周波数ベクトル A[]と今回 提案の t=0 付近高密度 sampling 方式で求めた ベクトル A[]の値を図 11 と図 12 に示します。



A[]=invH[][]a[]={1.032, 1.500, 0.0, 0.0, -0.809, -0.500, -0.223, 0.0}

図 12 t=0 付近高密度 sampling 方式

離散フーリエ変換は元来 cosine 関数等の周波 数成分を求めるのに最適ですが、図 12 の結果 ではかなりの aliasing が目立ちます。

文 献

- [1] 64 Point FFT Design Manual,64 点高速フーリエ変換回路設計仕様書,和田知久、Design Wave Magazine 2006年11月号、pp.143-156
- [2] 数値計算法、三井田惇郎,須田 宇宙共著、森北出版、第8章離散フーリエ変換、pp.96-110.