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)

https://www.ieej-ect.org/ect/ectp/2014/p14-07-03.html

電気学会 電子回路研究会

日時

2014年7月4日(金) 09:30~12:05

@ニューウェルシティ出雲(7/4) 島根県出雲市塩冶有原町2-15-1

http://www.nwci.jp/access

論文番号 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 維焜,吉田侑司,深草紀志,萩原良昭(崇城大学)

電子回路研究会

〔委 員 長〕 島 健(神奈川大学) 〔幹 事〕 西城和幸,松元藤彦(防衛大学校),傘昊(東京都市大学) 〔幹事補佐〕 ニコデムス レディアン(神奈川大学)

日時 2014年7月3日(木) 13:00~16:25

2014年7月4日(金) 09:30~12:05

場 所 大社文化プレイス(7/3) 島根県出雲市大社町杵築南1338-9 出雲大社前駅下車

http://www.city.izumo.shimane.jp/www/contents/1310104798186/html/common/52c89d9b021.html

ニューウェルシティ出雲(7/4) 島根県出雲市塩冶有原町2-15-1 http://www.nwci.jp/access

(大社文化プレイス) 7月3日(木) 13:00~14:30

座長 奥田裕一 (ルネサスエレクトロニクス)

ECT-14-052 対称型フローティングインピーダンススケーリング回路の諸特性に関する考察

◎西岡周造,松元藤彦,藤井達哉,大淵武史(防衛大学校)

ECT-14-053 集積化可変RCポリフェーズフィルタの設計

◎伴内優貴,谷本 洋,吉澤真吾(北見工業大学)

ECT-14-054 遺伝的アルゴリズムを用いた複素フィルタの自動設計

◎ 関 洋明,高井伸和,小林春夫,根岸孝行,加藤雅人,菅原誉士紀,鈴木研人(群馬大学)

ECT-14-055 Active Inductorを用いたMixerの消費電流の低減

◎古谷 吏,兵庫 明,松浦達治(東京理科大学)

(大社文化プレイス) 7月3日(木) 14:45~16:25

座長 松野隼也(東芝)

ECT-14-056 β展開に基づくパイプラインADCにおける?β推定手法の検討

◎成田隼斗,内田順平,山田俊毅(東京都市大学),松浦達治(東京理科大学),傘 昊,堀田正生(東京都市大学)

ECT-14-057 カレントミラーアンプを用いたβ変換サイクリック形AD変換器の試作および評価結果

◎内山亜沙人,吉田勇太,山田俊毅,吉田昴右(東京都市大学),松浦達治(東京理科大学) ,傘 昊,堀田正生(東京都市大学)

ECT-14-058 CMOS標準ロジックを用いた確率的フラッシュ型AD変換器の試作と評価

◎竹端久登,谷本 洋,吉澤真吾(北見工業大学)

ECT-14-059 逐次比較近似ADCの整数論に基づく冗長アルゴリズム設計

◎小林佑太朗,小林春夫(群馬大学)

電子回路研究会

〔委 員 長〕 島 健(神奈川大学) 〔幹 事〕 西城和幸,松元藤彦(防衛大学校),傘昊(東京都市大学) 〔幹事補佐〕 ニコデムス レディアン(神奈川大学)

日時 2014年7月3日(木) 13:00~16:25

2014年7月4日(金) 09:30~12:05

場所 ニューウェルシティ出雲(7/4) 島根県出雲市塩冶有原町2-15-1 http://www.nwci.jp/access

((ニューウェルシティ出雲) 7月4日(金) 9:30~10:45

座長 香川景一郎(静岡大学)

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 維焜,吉田侑司,深草紀志,萩原良昭(崇城大学)

ECT-14-061 CMOSイメージセンサにおける画素信号高速読出のための列回路の回路定数決定に関する一考察

◎高橋渓輔,兵庫 明,松浦達治(東京理科大学),新谷 悟,井上俊輔,譲原浩(キヤノン株式会社)

ECT-14-062 BSIM4による90nm n-channel MOSFETの Hot Electron の劣化特性モデル化に関する研究

◎ 戸塚拓也,青木 均,安部文隆, Khatami Ramin,新井薫子,轟俊一郎,香積正基,王 太峰,小林春夫(群馬大学)

(ニューウェルシティ出雲) 7月4日(金) 11:00~12:05

座長 萩原良昭(崇城大学)

ECT-14-063 カレントブリーディングを用いたソースフォロワ負帰還LNAの高利得化・広帯域化の検討

◎倉田圭輔,兵庫 明,松浦達治(東京理科大学)

ECT-14-064 デルタシグマTDCを用いた位相ノイズ測定

◎大澤優介,平林大樹,針谷尚裕,小林春夫(群馬大学),新津葵一(名古屋大学), 小林 修(半導体理工学研究センター)

ECT-14-065 並列数可変制御型降圧コンバータに関する研究

◎三次将太,兵庫 明,松浦達治(東京理科大学)

※原則はA講演で25分、一部B講演で20分(共に質疑応答5分を含む)

※講演者氏名の ◎は代表発表者であり且つ若手表彰選考対象者 ○は代表発表者

公開後の変更やその他の情報は、電子回路研究会のホームページをご覧下さい。

http://ieej-ect.org/ect/です。

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。

 $\sum \subset \mathcal{C}, M = \mathbf{m_0} + 4 \mathbf{m_1} + 16 \mathbf{m_2} ; \dots (5)$ 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4 x 4 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 [$m_0 + 4 m_1 + 16 m_2$]

 $= R3 [m_2 + 4 m_1 + 16m_0] \dots (1 4)$

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。

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

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

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

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。

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

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

...(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.