

短時間スペクトラム解析の計算アーキテクチャの検討

佐々木 秀* 小林 春夫 (群馬大学)

Study of Computation Architecture for Short-Time Spectrum Analysis

Shu Sasaki*, Haruo Kobayashi (Gunma University)

キーワード：スペクトラム解析，乗算器，加算器/減算器，ルックアップテーブル，
(Spectrum Analysis, Multipliers, Adder/ Subtractor, Look-up Table)

1. はじめに

本論文では、少ないサンプリング点数で高精度高分解能スペクトラム推定アルゴリズムと、その FPGA 実現に適した構成を検討したので報告する。

周波数は時間によらず統計的性質が一定の波形に対して定義される。[1,2] しかしノイズや信号の過度状態など、ある条件下においてはその周波数が一瞬だけ上昇、下降などの急激な変化が生じることがある。[2] それらの波形の過渡現象のスペクトル測定する際、波形をサンプリングして良く用いられている周波数解析手法 DFT (離散フーリエ変換)、FFT (高速フーリエ変換) を用いるのは次の問題がある。[1]

- 1) 細かな周波数分解能を得るためには短い過渡時間で多くのサンプリング点数が必要なので高速サンプリングが必要になる。
- 2) 必要な部分の周波数帯域のみならず周波数全体を測定してしまうため多くの計算時間とコストを要する。

ここでは少ないサンプリング点数で高周波数分解能測定するため離散フーリエ変換の定義に戻って任意の細かい周波数分解能で計算するアルゴリズムを用いる。また周波数範囲を狭め必要な周波数帯の範囲を測定し計算時間を短縮する。また、その計算専用ハードウェアを少量回路・低消費電力で FPGA 実現する構成を検討した。すなわち乗算器を使わずに下記の公式と 2 乗を得るルックアップテーブル(Look Up Table: LUT) を用いる。

$$AB = \frac{1}{2}[(A+B)^2 - A^2 - B^2]$$

2. 検討した周波数のスペクトラム推定アルゴリズム

検討した周波数推定アルゴリズムを示す。波形を一定時間(Ts)間隔でサンプリングする (図 1)。それらのサンプリング点をたとえば (5 点) 順にデータ a_0, a_1, a_2, a_3, a_4 とおく。そのスペクトルは次のように計算できる。

$$F(j\omega) = a_0 + a_1e^{-j\omega Ts} + a_2e^{-j\omega 2Ts} + a_3e^{-j\omega 3Ts} + a_4e^{-j\omega 4Ts} \dots (1)$$

この式 (1) に基づき ある特定帯域 $\omega_1 < \omega < \omega_2$ で細かな周波数分解能 $\Delta\omega$ で $F(j\omega)$ を計算する。

$F(j\omega_1), F(j(\omega_1+\Delta\omega)), F(j(\omega_1+2\Delta\omega)), F(j(\omega_1+3\Delta\omega)), \dots F(j\omega_2)$.

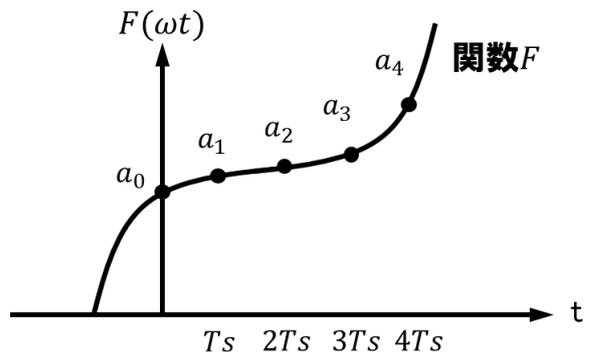


図 1 波形のサンプリング

Fig. 1. Sampling of waveform.

3. スペクトル推定アルゴリズムの専用ハードウェア構成の検討

3.1 構成法 1：乗算器を使用し計算式を直接構成

オイラーの公式を用いて (1) 式を整理し次式を得る。
 $|F(j\omega)| = (a_0^2 + a_1^2 + a_2^2 + a_3^2 + a_4^2 - 2a_0a_4 \cos(\omega 4Ts) + 2(a_0a_3 + a_4a_1) \cos(\omega 3Ts) + 2(a_2a_0 - a_1a_3 + a_2a_4) \cos(\omega 2Ts) + 2(a_0a_1 + a_1a_2 + a_2a_3 + a_3a_4) \cos(\omega Ts))^{\frac{1}{2}}$
(2)

これをハードウェア実現する構成を図 2 に示す。

3.2 構成法 2: 少ない乗算器で実現する構成

(2) 式を

$$AB = \frac{1}{2}[(A+B)^2 - A^2 - B^2] \dots\dots\dots(3)$$

の式を用いて整理する。

$$|F(j\omega)| = (a_0^2 + a_1^2 + a_2^2 + a_3^2 + a_4^2 - \{(a_0 + a_4)^2 - a_0^2 - a_4^2\} \cos(\omega 4Ts) + \{(a_0 + a_3)^2 + (a_1 + a_4)^2 - a_0^2 - a_1^2 - a_3^2 - a_4^2\} \cos(\omega 3Ts) + \{(a_0 + a_2)^2 - (a_1 + a_3)^2 + (a_2 + a_4)^2 - a_0^2 + a_1^2 - 2a_2^2 + a_3^2 - a_4^2\} \cos(\omega 2Ts) + \{(a_0 + a_1)^2 + (a_1 + a_2)^2 + (a_2 + a_3)^2 + (a_3 + a_4)^2 - a_0^2 - 2a_1^2 - 2a_2^2 - 2a_3^2 - a_4^2\} \cos(\omega Ts))^{\frac{1}{2}} \dots\dots\dots(4)$$

この式に基づき、専用計算構成ブロックを図3のよう設計した。

3.3 構成法 3: LUT による 2 乗演算を使用し乗算器を使わない構成

(4) 式をさらに整理して加減算と 2 乗算のみにする。

$$|F(j\omega)| = \left(a_0^2 + a_1^2 + a_2^2 + a_3^2 + a_4^2 - \frac{1}{2} \{ (A + \cos(\omega 4Ts))^2 - (B + \cos(\omega 3Ts))^2 - (C + \cos(\omega 2Ts))^2 - (D + \cos(\omega Ts))^2 - A^2 + B^2 + C^2 + D^2 - \cos^2(\omega 4Ts) + \cos^2(\omega 3Ts) + \cos^2(\omega 2Ts) + \cos^2(\omega Ts) \} \right)^{\frac{1}{2}} \dots\dots\dots(5)$$

ここで

$$A = (a_0 + a_4)^2 - a_0^2 - a_4^2 \dots\dots\dots(6)$$

$$B = (a_0 + a_3)^2 + (a_1 + a_4)^2 - a_0^2 - a_1^2 - a_3^2 - a_4^2 \dots(7)$$

$$C = (a_0 + a_2)^2 - (a_1 + a_3)^2 + (a_2 + a_4)^2 - a_0^2 + a_1^2 - 2a_2^2 + a_3^2 - a_4^2 \dots\dots\dots(8)$$

$$D = (a_0 + a_1)^2 + (a_1 + a_2)^2 + (a_2 + a_3)^2 + (a_3 + a_4)^2 - a_0^2 - 2a_1^2 - 2a_2^2 - 2a_3^2 - a_4^2 \dots\dots\dots(9)$$

である。

データの 2 乗値は LUT で得る構成をとると乗算器が不要になる。(5) 式に基づき専用計算構成ブロックを図4のよう設計した。

4. 考察

● 最も回路規模・消費電力が大きく、スピードのボトルネックになる可能性が高い回路ブロックは乗算器であるが、図2の構成では乗算器を 14 個、図3では 3 個、図4では一個も使用していない。回路規模は図3が一番大きく見えるが、乗算器を使用していないため実装したときの規模は小さい。むしろ LUT と加算器、減算器のみにしたことで乗算器を使用した従来の方式よりも回路量が

少なくなり、計算時間の短縮が見込める。

● このブロック図の設計の正当性は、 $a_0 \sim a_4$ それぞれに適切な値をおいて計算式とブロック図の二通りで計算して結果を比較して検証した。検証した結果、計算式においてもブロック図においても同じ出力値を得ることが出来た。

● このハードウェア構成を FPGA 実現することで周波数スペクトル解析がリアルタイムで処理でき、メモリサイズ(回路規模)が小さくなり、低コスト化・低消費電力実現が期待できる。

● 2つのデータ A, B に対して AB の乗算を、乗算器を用いずに加算器と LUT を用いて計算するためのハードウェアは多くの場合次の構成をとる。

- ① A, B の対数 $\log A, \log B$ を対数データの LUT を用いて得る。
- ② $\log A + \log B (= \log AB)$ を加算器を用いて得る。
- ③ $\log AB$ から AB を指数データの LUT を用いて得る。

高精度で対数、指数を得るためにはビット数を大きく(すなわち LUT のサイズが大きく)しなければならない。一方、構成法 2, 3 では (3) 式を用いる。

$$AB = \frac{1}{2}[(A+B)^2 - A^2 - B^2]$$

2 乗のデータ、1/2 乗のデータを得る LUT はビット数が小さくても上記演算が比較的小さくても AB 計算が高精度に計算できることが期待できる。(定量的評価はこれからである。)

● Cosine の計算も LUT を用いる。余弦波生成用 Direct Digital Synthesizer でよく用いられている技術であるのでそれを援用できる。[3]

5. まとめと今後の課題

短時間スペクトル推定のアルゴリズムとそのハードウェア実現法を検討した。今後は実際に FPGA 実装を行い提案構成の実機検証していく。

具体的には次を重点的に検証していく。

- ・計算精度と演算器・LUT のビット数の明確化
- ・実装 FPGA 動作クロック周波数) と計算速度の明確化
- ・データが 5 点より多くなった場合の計算式とハードウェア構成の検討

文 献

- (1) A. V. Oppenheim, and R. W. Shafer, Digital Signal Processing, Prentice-Hall (1975)
- (2) L. Cohen, Time-Frequency Analysis, Prentice Hall
- (3) A Technical Tutorial on Digital Signal Synthesis, Analog Devices, Inc. (1999).

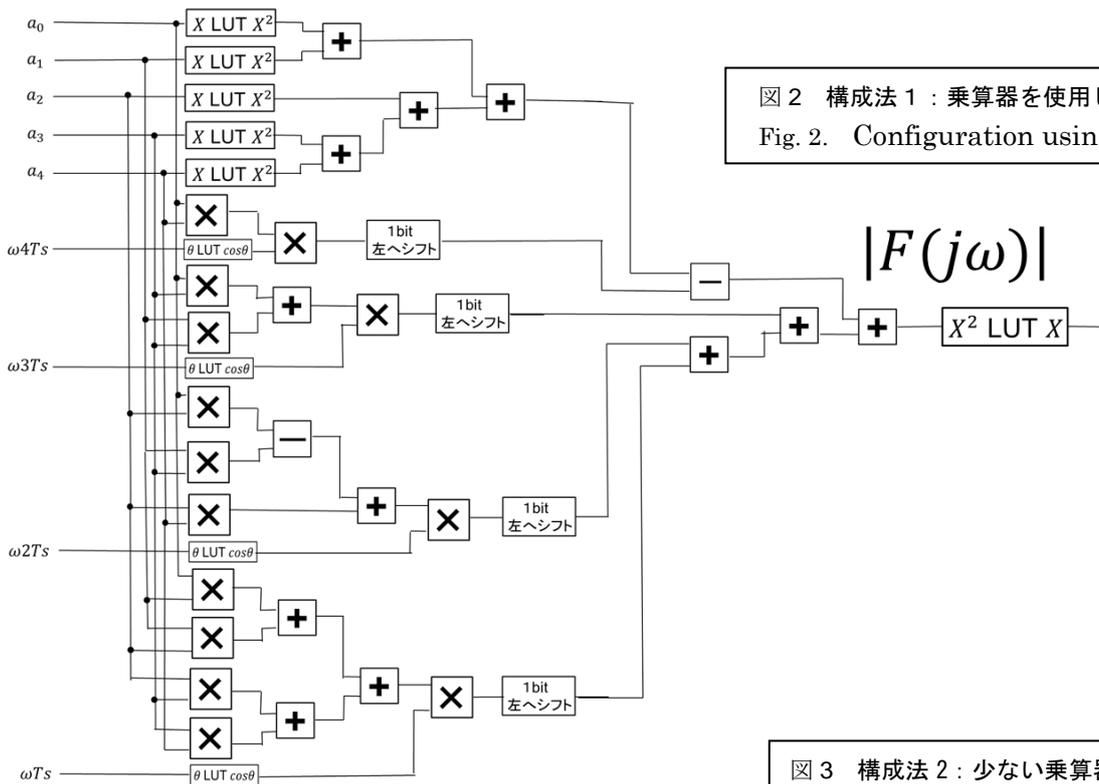


図2 構成法1：乗算器を使用し計算式を直接構成
Fig. 2. Configuration using multipliers

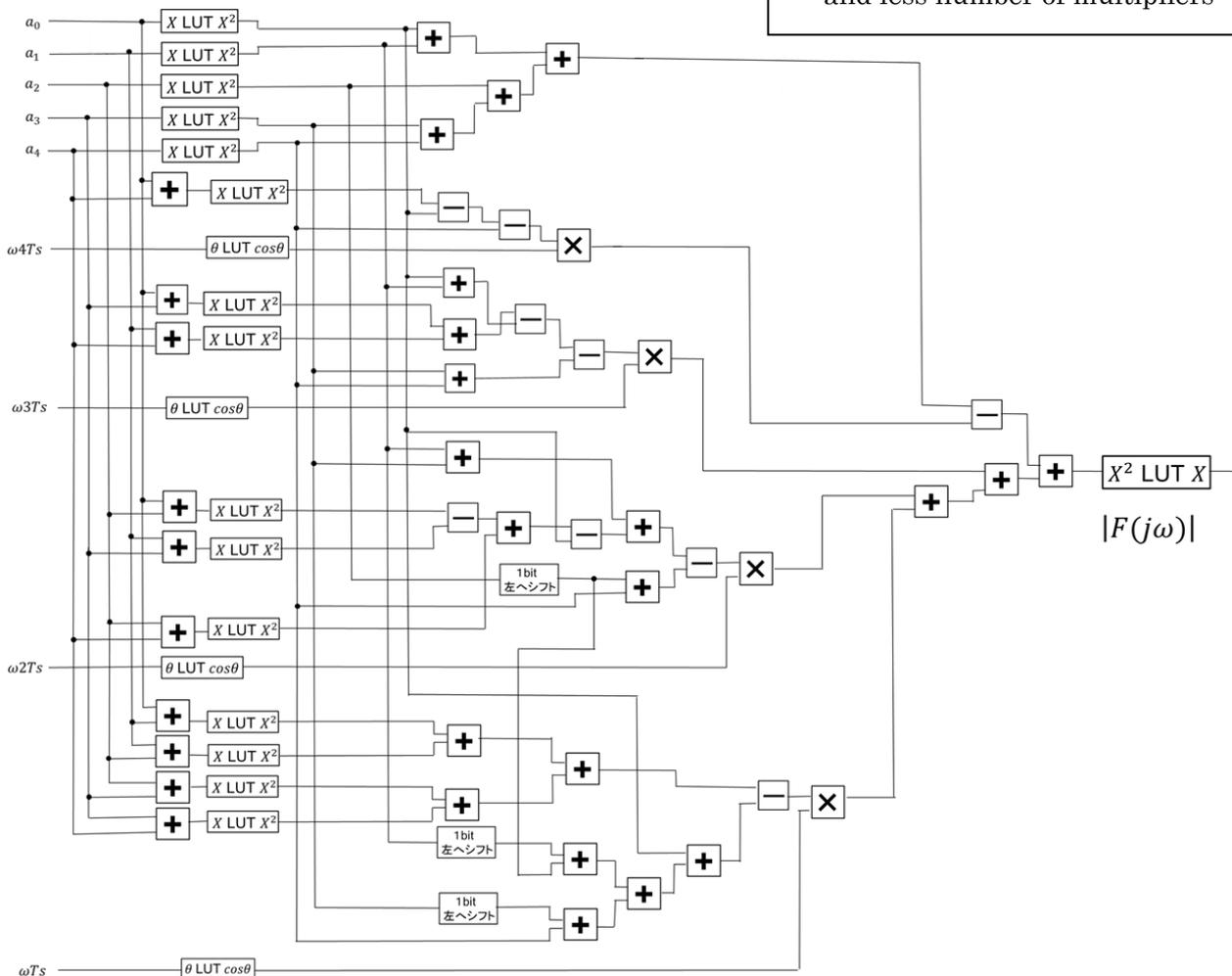


図3 構成法2：少ない乗算器で実現する構成
Fig. 3. Configuration with LUTs and less number of multipliers

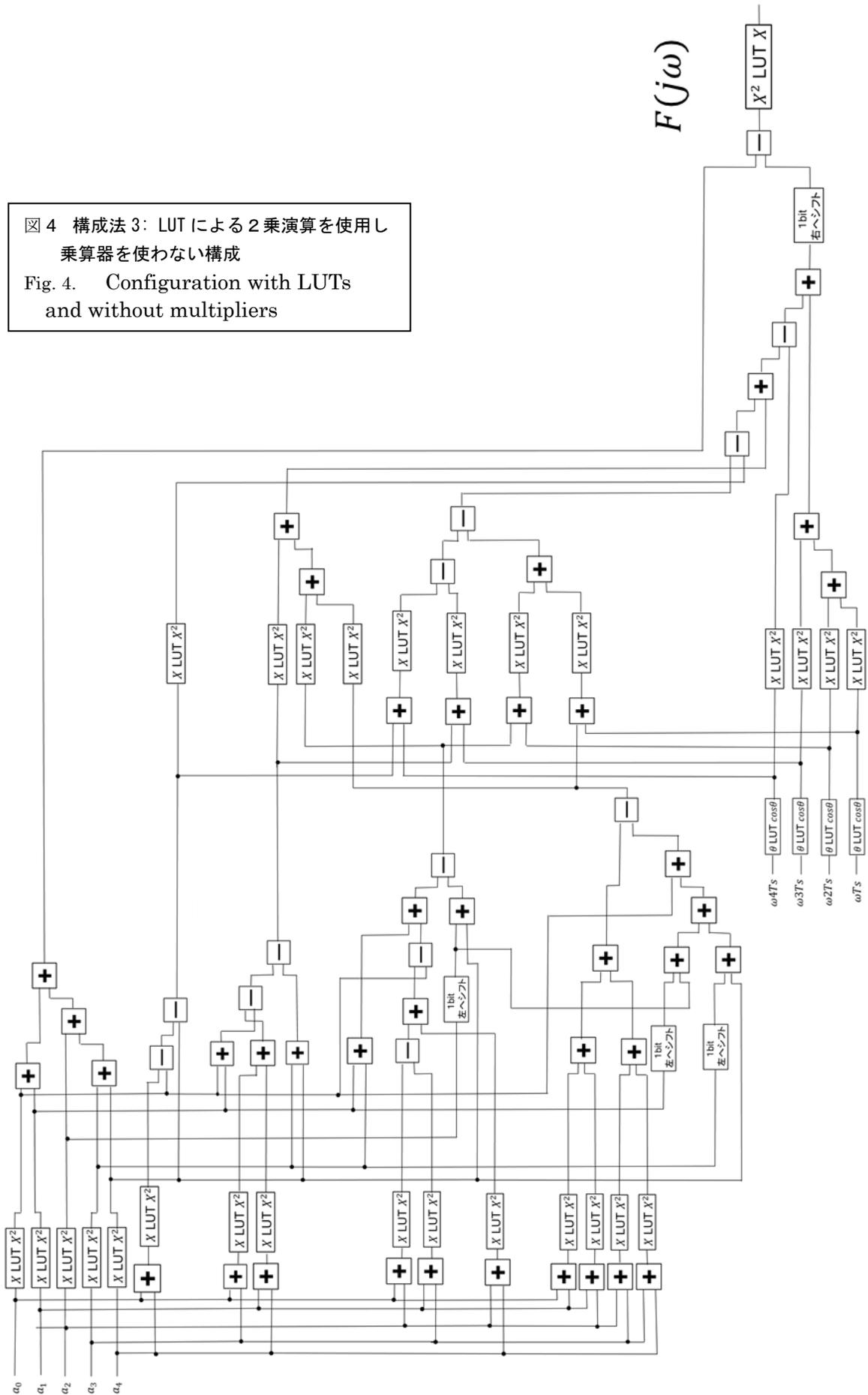


図4 構成法3: LUTによる2乗演算を使用し乗算器を使わない構成
 Fig. 4. Configuration with LUTs and without multipliers