

高位合成を用いた 2 次元 FFT 計算回路の作成と検証

design and verification of two-dimensional FFT calculation circuit using high-level synthesis

佐藤 紘将*

Hiromasa Sato

廣田 祐輔*

Yusuke Hirota

山本 洋太*

Yota Yamamoto

増田 信之*

Nobuyuki Masuda

1. まえがき

近年、大規模シミュレーションを高速に実現するために FPGA(Field Programmable Gate Array) を用いる研究が盛んに行われている。増田研究室では FPGA を用いた専用計算機の開発を行っており、その一つとしてデジタルホログラフィ専用計算機が開発されている。デジタルホログラフィ専用計算機内部には 2 次元 FFT(Fast Fourier Transform) の計算回路が実装されており、CGH(Computer Generated Holography) の再生像を計算する際に 2 次元 FFT を用いることにより、4 変数の計算式を 2 変数に減らすことが出来るため、計算を高速化させることに成功している。しかし、現在のデジタルホログラフィ専用計算機では、1 次元 FFT を 2 回用いることによって 2 次元 FFT を実現している。1 次元 FFT を 2 回用いるのではなく、2 次元 FFT のアルゴリズムを FPGA 内に実装することにより、さらなる高速化が図れると考えられる。そこで本研究では高位合成を用いた 2 次元 FFT 計算回路の作製及び検証を目的とした。

2 次元 FFT アルゴリズムを FPGA に実装する先駆けとして本研究では高位合成を用いて、C 言語で記述した 2 次元 FFT を HDL(Hardware Description Language) 化し、シミュレーション上での速度比較を行った。結果として 2 次元 FFT は、1 次元 FFT を 2 回用いて 2 次元 FFT を実装した方法より高速に動作することが確認できた。

2. 高速フーリエ変換

FFT(Fast Fourier Transform)

$n \times n$ 個の入力データ $f(i, j)$ ($i, j = 0, 1, \dots, n-1$) が与えられたとき、2 次元の離散フーリエ変換 $F(p, q)$ 及び逆変換 $f(i, j)$ は、

$$F(p, q) = \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} f(i, j) W_n^{ip+jq} \quad (1)$$

$$f(i, j) = \frac{1}{n^2} \sum_{p=0}^{n-1} \sum_{q=0}^{n-1} F(p, q) W_n^{-(ip+jq)} \quad (2)$$

ただし、

$$W_n = \exp(-\sqrt{-1} \cdot 2\pi/n) \quad (3)$$

で定義される。2 次元 FFT を実行する方法の一つとして、次式のように 1 次元 FFT アルゴリズムを繰り返す

使用する手法がある。

$$\begin{aligned} F(p, q) &= \sum_{j=0}^{n-1} \left[\sum_{i=0}^{n-1} f(i, j) W_n^{ip} \right] W_n^{jq} \\ &= \sum_{j=0}^{n-1} f_1(p, j) W_n^{jp} \end{aligned} \quad (4)$$

この手法は 2 次元の 1 次元 FFT スキャン法と呼ばれている。増田研究室で作成されたデジタルホログラフィ専用計算機はこのアルゴリズムを用いて 2 次元 FFT を実装している。次に 2 次元 FFT のアルゴリズムについて述べる。まず、 p, q, i, j の 2 進数表示は次のようになる。

$$\begin{aligned} p &= (p_{m-1} p_{m-2} \cdots p_1 p_0) \\ q &= (q_{m-1} q_{m-2} \cdots q_1 q_0) \\ i &= (i_{m-1} i_{m-2} \cdots i_1 i_0) \\ j &= (j_{m-1} j_{m-2} \cdots j_1 j_0) \end{aligned} \quad (5)$$

これを式 1 に代入すると、

$$\begin{aligned} F(p, q) &= F(p_{m-1} \cdots p_0, q_{m-1} \cdots q_0) \\ &= \sum_{j_0} \cdots \sum_{j_{m-1}} \sum_{i_0} \cdots \sum_{i_{m-1}} f(i_{m-1} \cdots i_0, j_{m-1} \cdots j_0) \\ &\quad \times W_n^{p(i_{m-1} 2^{m-1} + \cdots + i_0) + q(j_{m-1} 2^{m-1} + \cdots + j_0)} \end{aligned} \quad (6)$$

を得る。次に、加算の順序を交換して次式のように変形する。

$$\begin{aligned} &f_1(p_0 i_{m-2} \cdots i_0, q_0, j_{m-2} \cdots j_0) \\ &= \sum_{j_{m-1}} \sum_{i_{m-1}} f(i_{m-1} \cdots i_0, j_{m-1} \cdots j_0) \times W_n^{p i_{m-1} 2^{m-1} + q j_{m-1} 2^{m-1}} \\ &f_2(p_0 p_1 i_{m-3} \cdots i_0, q_0 q_1 j_{m-3} \cdots j_0) \\ &= \sum_{j_{m-2}} \sum_{i_{m-2}} f_1(p_0 i_{m-2} \cdots i_0, q_0 j_{m-2} \cdots j_0) \times W_n^{p i_{m-2} 2^{m-2} + q j_{m-2} 2^{m-2}} \\ &\quad \dots \dots \dots \\ &f_m(p_0 \cdots p_{m-1}, q_0 \cdots q_{m-1}) \\ &= \sum_{j_0} \sum_{i_0} f_{m-1}(p_0 \cdots p_{m-2}, q_0 \cdots q_{m-1}, j_0) \times W_n^{p i_0 + q j_0} \\ &\equiv F(p_{m-1} \cdots p_0, q_{m-1} \cdots q_0) \end{aligned} \quad (7)$$

以上の式を m 回繰り返すことで、フーリエ変換が終了する。上式より、1 次元 FFT と同様にビット逆転を行う必要があることが分かる。本研究で使用したプログラムでは、入力データに対してビット逆転を行ってから 2 次元 FFT を実行させている。

*東京理科大学基礎工学研究所

†千葉大学大学院工学研究科

3. 実験結果

3.1 シミュレーション環境

シミュレーションを行う上でのターゲットデバイスとして、Xilinx 社が提供する Virtex®-7 FPGA VC707 評価キットを用いた。2次元FFT計算回路を作成する際に使用したシミュレーション環境を表1に示す。

表1: シミュレーション環境

CPU	Intel Core i5-4460 3.20GHz
メモリ	8.00GB
OS	Windows7 Pro
開発環境	Vivado HLS 2017.2
コンパイラ	gcc 5.4.0
FFTW ライブラリ	FFTW 3.3.6

次に入力データ 8×8 に対する2次元の1次元FFTスキャン法及び、2次元FFTの合成結果及びシミュレーション結果を表2及び表3に示す。

表2: double型 高位合成結果 リソース使用料

	1次元スキャン法	2次元FFT
DSP48E[個]	524	258
FF[個]	58,673	45,623
LUT[個]	65,168	54,914

表3: double型 高位合成結果 動作時間

	1次元スキャン法	2次元FFT
レイテンシ	140	102
動作周波数 [MHz]	111.98	101.52
動作時間 [ns]	1,250.20	1,004.70

ここで言うレイテンシとは、最初の入力データがFPGAのレジスタに格納されてから、計算が終わり全ての計算結果がFPGAのレジスタに格納されるまでにかかるクロック数を表している。また、動作時間はレイテンシに動作周期を掛けることで導出した。表2及び表3より、2次元FFTアルゴリズムを用いた場合の方がリソース使用料が少なく、短い動作時間で計算が完了することが分かる。

次に2次元FFT計算回路の構成を16bit整数型にしたときのリソース使用量及び、CPUとの比較を表4及び表5に示す。入力データは 8×8 とし、比較対象としたCPUでの2次元FFTにおいては、オープンソースのFFTライブラリであるFFTWを使用して速度計測を行った。

表4: 16bit整数型におけるリソース使用量

	double型	16bit整数型
DSP48E[個]	258	60
FF[個]	45,623	6,985
LUT[個]	54,914	10,314

表5: 2次元FFT計算回路のCPUとの速度比較

	計算時間 [ns]	速度比
CPU(FFTW)	4,665	1
double型	1,005	4.6
16bit整数型	704	6.6

表4及び表5よりデータ幅を小さくすることでリソース及び動作時間が改善されることが確認できる。また、PCによる計算と比較して6.6倍の高速化を達成している。

4. まとめと今後の課題

2次元FFTアルゴリズムをデジタルホログラフィ専用計算機に実装する先駆けとして、Vivado HLSを用いた高位合成による2次元FFT計算回路の作成及び検証を行った。1次元FFT及び2次元FFTのプログラムを高位合成を用いてHDL化し、2次元の1次元スキャン法と2次元FFTアルゴリズムの速度比較をC/RTLシミュレーション上で行った結果、2次元FFTアルゴリズムを用いた場合の方が、リソース使用量が少なく、動作時間も短いことが分かった。以上の結果より、デジタルホログラフィ専用計算機に2次元FFTアルゴリズムを導入することによってさらなる高速化が期待できると言える。

今後は高位合成によって得られたFFT計算回路のさらなる改良と共に、手書きHDLでの2次元FFTの回路設計を進めFPGAへの実装を目指す。

参考文献

- [1] J.W.Cooley, J.W.Tukey, "An Algorithm for the Machine Calculation of Complex Fourier Series", Mathematics of Computation, Vol.19, pp.297-301, Apr.1965
- [2] 町田東一, 小島紀男, "FORTRAN 応用数値計算", 東海大学出版会, pp.94-141, 1989
- [3] 天野英晴, "FPGAの原理と構成", オーム社, pp.28-33, pp.86-87, pp.165-169, 2016
- [4] 小林優 "FPGA プログラミング大全", 秀和システム, pp.470-481, 2016
- [5] Shin-ichi Satake, Yoshiaki Hiroi, Yuya Suzuki, Nobuyuki Masuda, Tomoyoshi Ito "Special-purpose computer for two-dimensional FFT", Computer physics Communications, Vol.179, pp.404-408, 2008
- [6] Xilinx Corporation, "Vivado Design Suite ユーザーガイド 高位合成", https://japan.xilinx.com/support/documentation/sw_manuals_j/xilinx2017_1/ug902-vivado-high-level-synthesis.pdf, (参照 2018-6-26)