

## FPGA に適した複素乗算器構造の検討および評価

○ 佐藤 啓一<sup>†</sup> 多田 十兵衛<sup>‡</sup> 後藤 源助<sup>‡</sup> 田村 安孝<sup>‡</sup><sup>†</sup>山形大学大学院理工学研究科システム情報工学専攻 <sup>‡</sup>山形大学大学院理工学研究科情報科学専攻

## 1. はじめに

複素乗算は FFT や DCT 等で多用されており [1],[2], 信号処理にとって重要な演算である。また, アルゴリズムの高速計算手法としてハードウェア実装が挙げられ, 低コストで設計・開発が可能な FPGA がよく用いられる。しかしながら FPGA は回路構造が固定されているため, ASIC での設計技術を FPGA に導入すると必ずしも高性能な回路を設計できるとは限らない。本研究では FPGA の基本構造である Look up table(LUT)および Carry-chain を考慮した複素乗算器構造を検討および評価する。

## 2. 複素乗算

複素数  $(a+jb)$  および  $(c+jd)$  との複素乗算は一般的に (1) 式のように表される。  $j$  は虚数 ( $j^2=-1$ ) を示す。

$$(a + jb)(c + jd) = ac - bd + j(ad + bc) \quad (1)$$

実数部  $R=ac-bd$  および虚数部  $I=ad+bc$  である。(1) 式の複素乗算は 4 つの乗算および 2 つの加減算から構成されるので, 複素乗算器の形態は Fig. 1 (a) のように 4 つの乗算器および 2 つの加算器が必要となる。

複素乗算の別形態として (2) 式が挙げられる。

$$R = a(c+d) - d(a+b), I = a(c+d) + c(b-a) \quad (2)$$

この複素乗算では  $a(c+d)$  の項が実数および虚数部に対して共通項なので 3 つの乗算および 6 つの加減算から構成される。従ってこの複素乗算器の形態は Fig. 1 (b) であり, 3 つの乗算器および 6 つの加算器が必要となる。

本研究ではこの 2 つの形態について複素乗算器を設計および評価を行う。

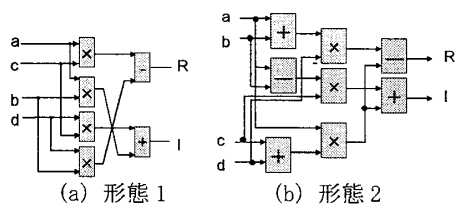


Fig. 1 複素乗算の形態

## 3. FPGA 構造

FPGA の構造の概略を Fig. 2 に示す。主に LUT, Carry-chain, および FF (Flip flop) から構成される [3]。任意の組み合わせ論理は 6 入力 1 出力ポートを持つ LUT (SRAM) により実装されるので入力信号は LUT へのアドレスとして扱われる。Carry-chain は加算な

どの演算時に高速キャリ伝播を行うための専用線である。HDL 記述において演算子 ('+', '\*', 等) の記述により使用可能である。本研究はこの LUT および Carry-chain に着目し, 複素乗算器の設計を行う。

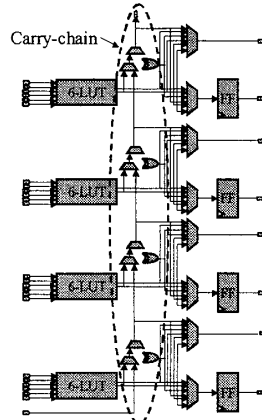


Fig. 2 FPGA 構造の概略 (Xilinx Virtex5)

## 4. 複素乗算器構造

Fig. 1 の各形態に基づいて複素乗算器を設計する。データ幅は 18 ビットである。Fig. 3 および Fig. 4 に複素乗算器構造の概略を示す。主に部分積生成部, 部分積圧縮部, および Carry-chain を用いた桁上げ伝播加算器 (CPA) から構成される。

各部において, 部分積生成部は 2 次 Booth アルゴリズムを導入した構造であり, 部分積数を約半分に削減する。部分積圧縮部においては Carry save adder (CSA) を用いた Wallace tree 構造を用いる。

形態 1 の回路 (Fig. 3) では, 4 つの積に対して部分積生成回路を導入する。Booth encoder は乗数ビットのエンコードを行い, Booth selector がその結果に応じて部分積ビットを出力する。但し "-bd" の積については負数なので, 負数の部分積を生成する N\_Booth selector を用いる。生成された部分積は Wallace tree による桁上げ保存加算が行われる。圧縮された 2 つの部分積は実数および虚数部用それぞれ Carry-chain による加算が行われ, 複素積が出力される。回路構造を Fig. 3 に示す。

形態 2 の回路 (Fig. 4) では, はじめに括弧内の項の加減算を Carry-chain で行う。その結果を 3 つの積に対して部分積生成回路を導入し, 生成された部分積を Wallace tree および Carry-chain により加算することで複素積がそれぞれ出力される。

Examination and evaluation of complex multiplier structure suited for FPGA

<sup>†</sup>Keiichi Satoh, Yamagata university graduate school of science and engineering, department of system and information engineering

<sup>‡</sup>Jubei Tada, Gensuke Goto, and Yasutaka Tamura, Yamagata university graduate school of science and engineering, department of informatics

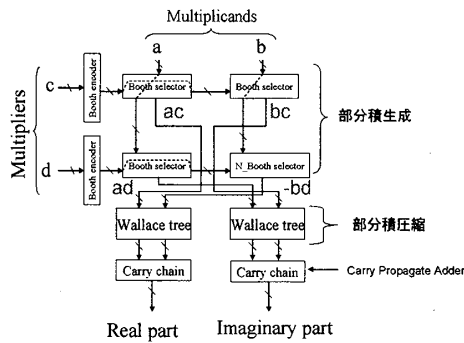


Fig. 3 形態1の複素乗算器構造

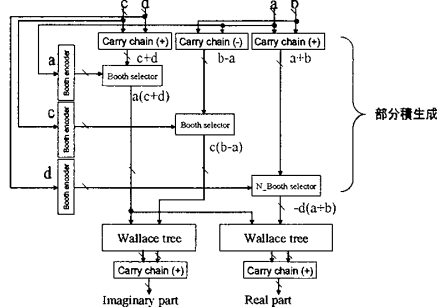


Fig. 4 形態2の複素乗算器構造

### 5. LUTに基づく部分積圧縮構造

FPGAに適した圧縮器構造として6:3-compressor [4]を提案する。Fig. 5にその構造を示す。これは6入力3出力ポートを持ち6入力ビット( $P_0$ - $P_5$ )を加算し、信号Sum, Carry<sub>0</sub>,およびCarry<sub>1</sub>を出力する。従って3つの6入力LUTが用いられる。入力ビットを単純に加算して3ビットを出力するため下位桁のビットを用いない。その結果1段分のLUT配置で済む。この圧縮器をWallace treeに導入して部分積圧縮部を構成する(Fig. 6)。このツリーにより圧縮される部分積数が3つになった時点でFull adder (FA)により2つに圧縮される。単一のFAは3入力2出力ポートを持つ構造なので、6入力LUTを二つ用いることで構成できる。従ってFAはFPGAに適した構造である。

2つに圧縮された部分積はCarry-chainにより加算された後複素積が出力される。

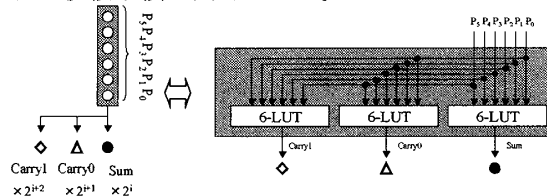


Fig. 5 6:3-compressor構造  
(左図：部分積圧縮の概略, 右図：内部構造)

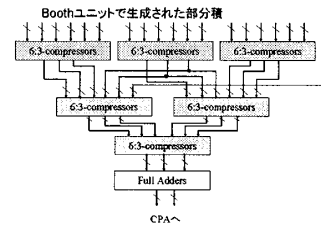


Fig. 6 6:3-compressorを用いたWallace tree構造

### 6. 設計および評価

形態1および2に基づく複素乗算器を設計および論理合成を行う。更にそれらの回路の比較対象として\*および+の演算子をVHDL記述により生成される複素乗算器も含める。開発環境はXilinx ISE v9.1.03iを用いる。設計後、論理合成された回路のパスディレイTおよび回路規模(LUT数)Nを調査し、それらを総合した回路性能  $P=1/(NT)$ を算出して評価する。それらの結果をTable 1に示す。

回路性能Pから、両形態の複素乗算器は演算子から生成されたものに比べ高性能であった。形態1および2の複素乗算器の比較において、形態2は形態1に比べパスディレイが約13%増加したものの回路規模が約5%減少した。従って回路規模だけを見ると形態2の方が有利である。しかしながらPを比較すると形態1の方が高い値を示した。従って形態1に基づく複素乗算器が高性能であることが確認された。これは形態2の演算過程が形態1よりも1段分多いことに起因すると考察される。

Table 1 各形態の複素乗算器の合成結果比較

複素乗算器形態	パスディレイT[ns]	回路規模N	$P=1/(NT)[1/s \times 10^3]$
形態1	5.62	1605	111
形態2	6.46	1520	101
演算子	9.68	2010	51

### 7. おわりに

FPGA構造に基づき複素乗算器構造を検討および評価を行った。その結果、演算子から生成された回路に比べ高性能な回路を得られ、形態1の複素乗算器が本研究の中では高性能である。また、FPGAに適した回路を得るのにその構造をうまく利用することは有効な手法であることが確認された。

### 参考文献

- [1] Steven W. Smith, "Digital Signal Processing: A Practical Guide for Engineers and Scientists - chapter9 Applications of the DFT", p180-184, Newnes, Sept. 2002.
- [2] X. Shao and S. G. Johnson, "Type-II/III DCT/DST algorithms with reduced number of arithmetic operations," arXiv.org e-Print archive, p. arXiv:cs.DS/0703150, Mar. 2007.
- [3] Xilinx Co., "Xcell journal vol.58.59", 2007 Spring.
- [4] Keiichi Satoh, Jubee Tada, Kenta Yamaguchi, and Yasutaka Tamura, "Complex Multiplier suited for FPGA structure", pp. 341 - 344, Proc. ITC - CSCC 2008, July. 2008