

## 可変基数 CORDIC プロセッサのハードウェアアルゴリズム

1 Q-2

北折 一郎

青木 孝文†

樋口 龍雄

東北大学大学院情報科学研究科

## 1. はじめに

初等関数演算のためのハードウェア向きアルゴリズムとして、CORDIC (COordinate Rotation DIgital Computer) アルゴリズムが知られている。このアルゴリズムによる演算は、シフトと加減算、および定数読み出しという単純な操作の繰り返しで実現できる。しかしながら、収束時間が演算語長に比例するため、DSP などへの応用にはその高速化が望まれる。本稿では、収束の各段階において基数を変化させることにより、高速な座標回転演算を実現する可変基数 CORDIC プロセッサのハードウェアアルゴリズムを示す。

## 2. 基数 2-4-8 CORDIC アルゴリズム

一般に、高基数の概念は算術演算の高速化に有効であるが、CORDIC への適用を考える場合には、(i) 各小回転のステップにおける回転桁が多値化するため、これをいかに高速に選択するか、さらに (ii) 回転桁の多値化に起因する座標ベクトルの伸び率 (スケールファクタ) の変化をいかに補正するか、の 2 点が問題になる。

筆者らは、冗長表現されたデータの一部を非冗長化することによって直接回転桁を決定するとともに、基数を段階的に変化させて近似的に定スケールファクタ演算を実現することにより、上記の問題を解決した基数 2-4-8 CORDIC アルゴリズムを提案している [1],[2]。本アルゴリズムによる 64 ビット精度座標回転演算は、次のように実現される。

≪ 基数 2 CORDIC アルゴリズム ( $i = 1, \dots, 22$ ) ≫

[入力]  $X_0, Y_0$ : 初期ベクトル,  $Z_0$ : 目的の回転角度

●座標回転演算

$$X_i = X_{i-1} - q_i \cdot 2^{-i} Y_{i-1} - p_i \cdot 2^{-2i-2} X_{i-1}$$

$$Y_i = Y_{i-1} + q_i \cdot 2^{-i} X_{i-1} - p_i \cdot 2^{-2i-2} Y_{i-1}$$

●角度演算

$$Z_i = Z_{i-1} - q_i \cdot 2 \cdot \tan^{-1}(2^{-i-1})$$

$$p_i, q_i \in \{0, \pm 1\}$$

≪ 基数 4 CORDIC アルゴリズム ( $i = 23, \dots, 28$ ) ≫

$$X'_{11} = X_{22}, Y'_{11} = Y_{22}, Z'_{11} = Z_{22}$$

$$j = i - 11$$

●座標回転演算

$$X'_j = X'_{j-1} - d'_j \cdot 4^{-j} Y'_{j-1} - \frac{1}{2} \cdot d_j'^2 \cdot 4^{-2j} X'_{j-1}$$

$$Y'_j = Y'_{j-1} + d'_j \cdot 4^{-j} X'_{j-1} - \frac{1}{2} \cdot d_j'^2 \cdot 4^{-2j} Y'_{j-1}$$

●角度演算

$$Z'_j = Z'_{j-1} - \tan^{-1}(d'_j \cdot 4^{-j})$$

$$d'_j \in \{0, \pm 2, \pm 4\}$$

≪ 基数 8 CORDIC アルゴリズム ( $i = 29, \dots, 39$ ) ≫

$$X''_{11} = X'_{17}, Y''_{11} = Y'_{17}, Z''_{11} = Z'_{17}$$

$$k = i - 17$$

●座標回転演算

$$X''_k = X''_{k-1} - d''_k \cdot 8^{-k} Y''_{k-1}$$

$$Y''_k = Y''_{k-1} + d''_k \cdot 8^{-k} X''_{k-1}$$

$$d''_k \in \{0, \pm 1, \dots, \pm 5\}$$

各基数における回転桁 ( $q_i, p_i, d'_j, d''_k$ ) の決定方法の詳細は、[1] および [2] を参照されたい。

( $X_0, Y_0$ ) に初期ベクトル,  $Z_0$  に目的とする回転角度を与えることにより、39 ステップ ( $i = 39$ ) 後に回転後の座標 ( $X_{39}, Y_{39}$ ) が得られる。

## 3. 可変基数 CORDIC プロセッサ

以下では、基数 2-4-8 CORDIC アルゴリズムを実現する可変基数 CORDIC プロセッサを提案する。上記アルゴリズムの実現に際しては、各基数ごとに異なるハードウェアを用意するのではなく、単一のハードウェアですべての基数の演算を実行する必要がある。そのためには、基数に依らず同一の漸化式でアルゴリズムを表現できることが望ましい。

座標回転演算の漸化式は以下のように書き換えることができる。

$$X_i = X_{i-1} - T_2 \cdot Y_{i-1} - \frac{1}{2} \cdot T_3 \cdot X_{i-1} \quad (1)$$

$$Y_i = Y_{i-1} + T_2 \cdot X_{i-1} - \frac{1}{2} \cdot \delta_i \cdot T_3 \cdot Y_{i-1} \quad (2)$$

$$(i = 1, \dots, 39)$$

ただし、 $\delta_i$  を次式で定める。

$$\delta_i = \begin{cases} +1 & (i = 1, \dots, 28) \\ -1 & (i = 29, \dots, 39) \end{cases}$$

ここで、 $T_2$  および  $T_3$  のとり得る値を表 1 に示す。

表 1 から分かるように、 $T_2$  および  $T_3$  はすべての基数、回転桁において 0 または 2 のべき乗となっている。ゆえに、式 (1), (2) の第 2 項、第 3 項はすべてシフトと加算のみで実現できる。このため、基数によって回路

\*"A Hardware Algorithm of a Variable-Radix CORDIC Processor", Ichiro KITAORI, Takafumi AOKI, and Tatsuo HIGUCHI. Graduate School of Information Sciences, Tohoku University, Sendai-shi, 980-8579 Japan.

† 科学技術振興事業団, さきがけ研究 21 研究者兼任。

表 1:  $T_2, T_3$  の値

基数	回転桁	$T_2$	$T_3$
$r = 2$ ( $i = 1 \sim 22$ )	$(q_i, p_i) = (-1, +1)$	$-2^{-i}$	$+2^{-2i-1}$
	$(q_i, p_i) = (+1, +1)$	$+2^{-i}$	$+2^{-2i-1}$
	$(q_i, p_i) = (0, -1)$	0	$-2^{-2i-1}$
$r = 4$ ( $i = 23 \sim 28$ )	$d_i = \pm 4$	$\pm 2^{-2(i-11)+2}$	$\pm 2^{-4(i-11)+4}$
	$d_i = \pm 2$	$\pm 2^{-2(i-11)+1}$	$\pm 2^{-4(i-11)+2}$
	$d_i = 0$	0	0
$r = 8$ ( $i = 29 \sim 39$ )	$d_i = \pm 5$	$\pm 2^{-3(i-17)}$	$\pm 2^{-3(i-17)+3}$
	$d_i = \pm 4$	0	$\pm 2^{-3(i-17)+3}$
	$d_i = \pm 3$	$\pm 2^{-3(i-17)}$	$\pm 2^{-3(i-17)+2}$
	$d_i = \pm 2$	0	$\pm 2^{-3(i-17)+2}$
	$d_i = \pm 1$	$\pm 2^{-3(i-17)}$	0
	$d_i = 0$	0	0

構成が大きく変化することがなく、回路の一部を動的に再構成することにより、すべての基数における演算が図1に示すような単一のハードウェアで実現可能である。図のT2およびT3のブロックでは、表1の $T_2$ および $T_3$ の値によってシフト演算を実行する。参照テーブル選択回路（ATS）は回転桁の決定方法が基数によって異なるため、それぞれの基数について異なる回路が必要となる。

以上で述べた考え方に基づいて、可変基数CORDICプロセッサをVerilog-HDLを用いて設計した。図2は、ある目的角度(0.55755447431978[rad])を入力したときの回転演算の収束状況を表している。最終的に収束すべき座標の理論値とステップ*i*における座標との間の誤差(距離)を計算した。扱う語長が64ビットの場合、基数2の冗長CORDICでは64ステップ必要であるが、本プロセッサでは図から分かるように約39ステップで収束し、演算ステップ数が約60%に減少している。

4. まとめ

座標回転演算の漸化式を基数に依らず同一の式で表すことによって、回路の再構成を可能とし、オーバーヘッドの少ないコンパクトな回路が実現できることを示した。今後は、本プロセッサの回路構造の詳細について更に検討していく必要がある。

参考文献

[1] T. Aoki, H. Nogi, and T. Higuchi, "High-Radix CORDIC Algorithms for VLSI Signal Processing," *Proc. of the 1997 IEEE Workshop on Signal Processing Systems*, pp.183-192, November 1997.  
 [2] 青木孝文, 野木均, 樋口龍雄, "高基数CORDICアルゴリズム," 電子情報通信学会論文誌, D-I Vol. J81-D-I, No.4, pp.359-367, April 1998.

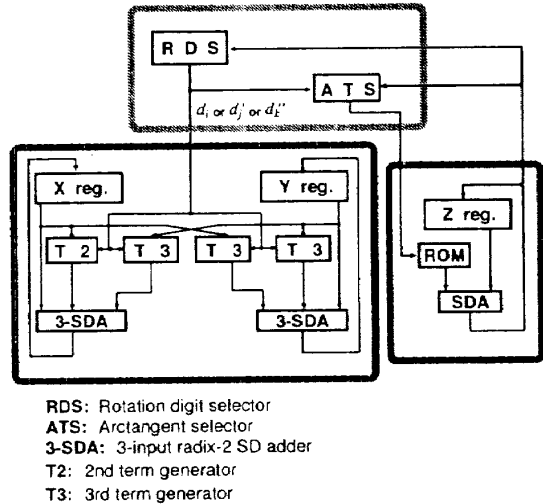


図 1: 可変基数CORDICプロセッサのブロック図

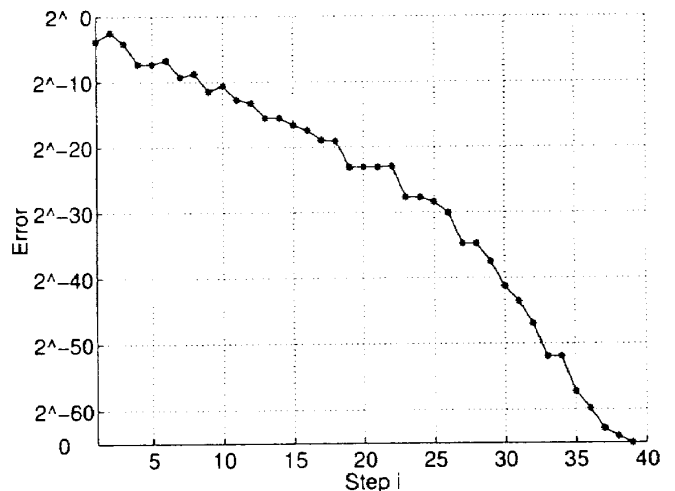


図 2: 回転演算の収束状況