

79-80

論 説

- デジタル画像におけるフーリエ記述子の回転不変性の検証
..... 淡 誠一郎 1
- 組込CPUにおける割込・分岐処理の高速化に関する研究
..... 岩出 秀平 21

研究ノート

- 根菜の画像解析のための形状記述についての研究ノート
-フーリエ記述子による形状記述- 淡 誠一郎 55

資 料

- 市民ランナーに対するマラソン前の30km走の効果検証のための研究プロ
トコル -ランダム化比較試験デザインを用いた研究-
..... 山内 武・高橋 秀人・得居 雅人・足立 哲司
杉山 喜一・串間 敦郎・中嶋 南紀・前田 和良
元根 朋美・松尾信之介 69

- 彙 報 81

- 掲載論文要旨 84

大阪学院大学
人文自然学会

2020

デジタル画像におけるフーリエ記述子の 回転不変性の検証

淡 誠一郎

A Study on Rotation Invariant Features of Fourier Descriptors on Digital Images

Seiichiro Dan

【Abstract】

In object recognition and clustering, silhouette of object is one of the most useful features. In most problems, invariances of scaling and rotation are required. The Fourier descriptors are a series of techniques for shape descriptions with various invariant features by treating the coordinates or the declination angles of the point sequence constituting the silhouette as a periodic function and apply Fourier series expansion on them. They are used for morphological analysis of animals and plants, pathological diagnosis, and design of industrial products.

Theoretically, Fourier descriptors have scale and rotation invariance. Furthermore, if the image is discretized with a sufficiently short sampling interval and sufficiently many quantization levels, even digital images have these properties approximately. However, conversely, for rasterized digital images, it also means that the invariances are somewhat disrupted.

In this study, we investigated the actual extent of rotation invariance using actual digital images. As a result, it has been found that the rotation invariance can be greatly lost in digital images.

概 要

物体の識別やクラスタリングにおいて、シルエットはもっとも重要な特徴の一つであり、多くの場合にスケーリングや回転に対する不変性が求められる。フーリエ記述子は、シルエットを構成する点列の座標、あるいは点列の偏角を周期関数とみなしてフーリエ級数展開することによって、スケーリングや回転に対して不変な形状記述を得ようとする手法であり、いくつかの形式化のバリエーションが提案されてきた。それらは動植物の形態学的解析や病理診断、工業製品のデザインなどに利用されている。

理論的には、フーリエ記述子がスケール不変性や回転不変性をもつことに疑いの余地はない。また十分短い標準化間隔と十分な量子化レベルをもって離散化されているならば、デジタル画像（ここではラスタ表現を前提としている）であっても、近似的にそれらの性質を持つであろうことも予想できる。しかし逆に言えば、ラスタ表現されたデジタル画像に対しては、フーリエ記述子の回転不変性は多少なりとも崩れるということも意味する。

本研究は実際の画像をもちいて、フーリエ記述の回転不変性がどの程度成り立っているのかを調査するものである。結果として、デジタル画像では回転不変性が大きく崩れることがあることが分かった。

1. はじめに

シルエットは対象を認識したり分類したりするための重要な手がかりの一つである。動植物の形態学的解析や種の分類に用いられ^[1,2]、医学分野での臓器の形状解析や病理診断^[3,4]、工業製品のデザインが人に与える印象の分析に用いられ^[5,6]たりもする。

たとえば、動植物の形態学的なシルエット解析は、旧来は目視による判断や人手での作業に依存していたため、容易に判断できる定性的な属性や、花卉の数などの離散的特徴量、全長や幅などの少数の定量属性に基づく分析にとどまらざるをえなかった。しかし、近年ではデジタル画像処理が手軽に利用できるようになり、シルエット解析の専用ツールが公開されるなど、形状を定量的に分析するための環境が整ってきており、計算機を用いた解析が主流となってきている。上に挙げた医学や工業分野においても同様である。

シルエット形状の記述方法にはさまざまなものがあるが、ここではフーリエ記述子を取り上げる。

フーリエ記述子の利点は、

- (1) スケール不変、回転不変の形状特徴量がえられる。
- (2) 低次の記述子のみを用いることにより形状の概形を記述できる。
- (3) 記述子と識別や診断結果との関係进行分析することで、画像による識別や診断を実現

できる。

といったことである。

一つ目に挙げた回転不変性は、フーリエ記述子の基礎であるフーリエ級数展開やフーリエ変換の性質からくる当然の性質と考えられており、実際に回転不変であるかどうか疑問視されることはない。同じ形状であれば回転させても周波数領域での記述は同じであると推測されるからである。

ただし、その性質が連続系において成り立つことは自明であっても、離散化した際にも成立するという保証はない。解像度を限りなく高く、すなわち標本化間隔を限りなく細かくし、量子化間隔も限りなく小さくしてやれば離散系で連続系を近似できるに違いない。しかし、実際の画像の解像度や量子化レベルが十分であるかどうかの検証は必要であり、盲目的にフーリエ記述子を使えば回転不変性が成り立つと考えるのは危険であろう。

そこで、本研究では、実際に人間であれば目視で識別が可能な解像度のシルエット画像を用い、フーリエ記述子の回転不変性がどの程度成り立っているのか、あるいは成り立っていないのかを検証してみたい。

2. フーリエ記述子

フーリエ記述子では、平面上の連続曲線を、端点から曲線に沿ってはかった道のりの長さ s の関数として記述し、その関数をフーリエ級数展開した際の係数を形状の記述ととらえる。

フーリエ級数展開は連続無限の周期関数を前提とした数学的操作であり、フーリエ記述子は、曲線の何をどう周期関数とみなすかによっていくつかのバリエーションが存在する。

以下、

- Z型フーリエ記述子
- 楕円フーリエ記述子
- G型フーリエ記述子
- P型フーリエ記述子

の概要を述べる。

① Z型フーリエ記述子

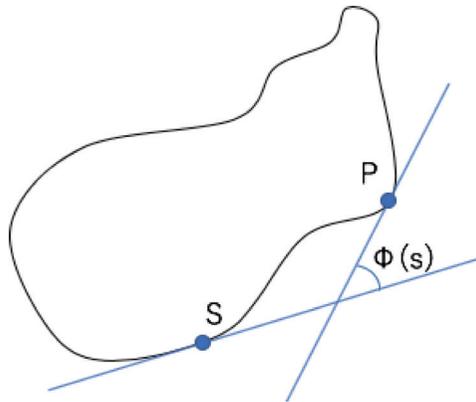


図1 偏角

まず、Z型フーリエ記述子でフーリエ記述子の基本的な考え方について説明しよう。

Z型フーリエ記述子では、対象を1つの閉曲線に限定して考える。

まず閉曲線上に任意に点Sを定め、これを始点として一定方向に曲線上をたどる。曲線上の点Pにおける接線と始点Sにおける接線のなす角を $\phi(s)$ で表し、偏角と呼ぶ(図1)。ただし、 s は始点から点Pまでの弧長(経路に沿って測った道のりの長さ)である。

曲線上をたどって再び始点に戻ったあとも、続けて曲線上をぐるぐると回り続けるものとし、始点に戻っても s や偏角はリセットせず積算する。そうすると、 $\phi(s)$ は1周ごとに 2π 増加していくような関数となる。

ここで、

$$t = \frac{2\pi s}{L} \quad \dots(1)$$

と変数変換し、新たに

$$\Phi(t) = \phi(t) - t \quad \dots(2)$$

という関数を定義して、**正規化偏角関数**とよぶことにする。変数 t の値は周回ごとに 2π 増加していき、 $\Phi(t)$ は周期 2π の連続周期関数となる。仮に図形の形状が真円だとすれば、 $\phi(t) = t, \Phi(t) = 0$ となる。

例えば、図2(a)に示す図形の偏角と正規化偏角関数は(b)のようになる。図のグラフは2周分をプロットしてある。

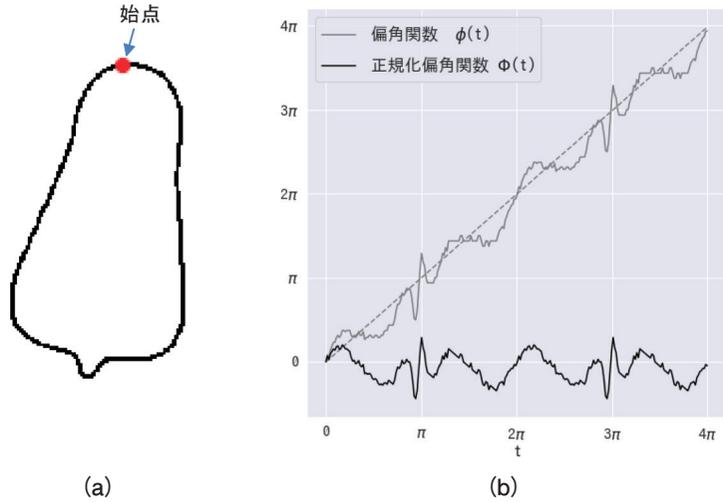


図2 偏角と正規化偏角関数

正規化偏角関数 $\Phi(t)$ は周期 2π の周期関数となるので、

$$\Phi(t) = \sum_{k=-\infty}^{\infty} c_k \exp(jkt) \quad \dots(3)$$

とフーリエ級数展開できる。このとき、 c_k を Z 型フーリエ記述子とよぶ。ただし、 j は虚数単位であり、

$$c_k = \frac{1}{2\pi} \int_{-\pi}^{\pi} \Phi(t) \exp(-jkt) dt \quad \dots(4)$$

である。

Z 型フーリエ記述子の離散表現

ここまで述べてきた理論は連続閉曲線を前提としているが、われわれが実際に扱うのはデジタル画像である。そこで、閉曲線上に等間隔に N 個の標本点 P_0, P_1, \dots, P_{N-1} を取って、閉曲線を等辺多角形で近似することを考え、各辺の方向ベクトル $\overrightarrow{P_i P_{i+1}}$ が基準辺の方向ベクトル $\overrightarrow{P_0 P_1}$ となす角 θ_i を「偏角」と考えて、離散表現する。

離散系の正規化偏角関数は、

$$\Phi_i = \theta_i - \frac{2\pi(i \bmod N)}{N} \quad \dots(5)$$

と記述できる。 $\Phi_i; i = 0, 1, \dots$ は周期 N の周期関数となり、

$$\Phi_i = \sum_{k=0}^{N-1} c_k \exp\left(j \frac{2\pi k i}{N}\right) \quad \dots(6)$$

と表せる。ここで、

$$c_k = \frac{1}{N} \sum_{i=0}^{N-1} \Phi_i \exp\left(-j \frac{2\pi k i}{N}\right) \quad \dots(7)$$

である。

注意しないといけないのは、離散フーリエ変換ではで k が大きいほど高調波に対応するとは限らないという点である。

$$\exp\left(j \frac{2\pi(N-k)}{N}\right) = \exp\left(j \frac{2\pi N}{N}\right) \exp\left(j \frac{-2\pi k}{N}\right) = \exp\left(j \frac{-2\pi k}{N}\right) \quad \dots(8)$$

であるから、

$$c_{N-k} = c_{-k} \quad \dots(9)$$

の関係がある。簡単のため、 N を偶数とすると、(6)式は、

$$\Phi_i = \sum_{k=-\frac{N}{2}+1}^{\frac{N}{2}} c_k \exp\left(j \frac{2\pi k i}{N}\right) \quad \dots(6)'$$

と書ける。

始点 (x_0, y_0) とフーリエ記述子 c_k がわかっているならば、次の計算式により、すべての標本点の座標 $(x_i, y_i) : i = 0, 1, \dots, N-1$ を復元することができ、標本点を順次繋いでいくことによって、元の曲線の近似多角形が得られる。

$$\left. \begin{aligned} \theta_i &= \Phi_i + \frac{2\pi(i \bmod N)}{N} \\ x_i &= x_0 + \delta \sum_{k=0}^i \cos\theta_k \\ y_i &= y_0 + \delta \sum_{k=0}^i \sin\theta_k \end{aligned} \right\} \quad \dots(10)$$

ここで、 δ は標本点間の距離、つまり多角形近似における 1 辺の長さである。

Z型フーリエ記述子の性質

理論上、Z型フーリエ記述子は、以下の性質を持つ。

- 記述子の大きさ $|c_k|$ は平行移動、拡大縮小、回転に対して不変である。
- 低い周波数のみを用いて逆変換をほどこすことで、元の図形の近似形状が得られる。
高周波成分を加えていくことによって、任意に詳細なレベルの形状変化まで再現できる。

平行移動と回転に関する不変性はフーリエ級数展開やフーリエ変換に共通する特徴である。理屈の上では、波形の周波数成分は平行移動や回転に左右されないため、この不変性は当然と言える。Z型フーリエ記述子は変換対象が角度であるため、拡大縮小に関しても不変性を持つ点特徴的である。

ただし、注意しないといけないのは、上に挙げた不変性はあくまでも連続関数に対する性質であり、離散化されたデータに対して成立するという保証はないという点である。

一般にフーリエ級数展開において高次の項をカットし、低次の項だけを用いて逆変換を行うと、高周波ノイズが除去され、平滑化された波形を得ることができる。Z型フーリエ記述子についても同様であるが、Z型フーリエ記述子では変換・逆変換の対象はシルエットではなく、正規化された偏角であり、形状は偏角から間接的に再構成しなければならない。このとき得られる曲線が閉じるという保証はない。

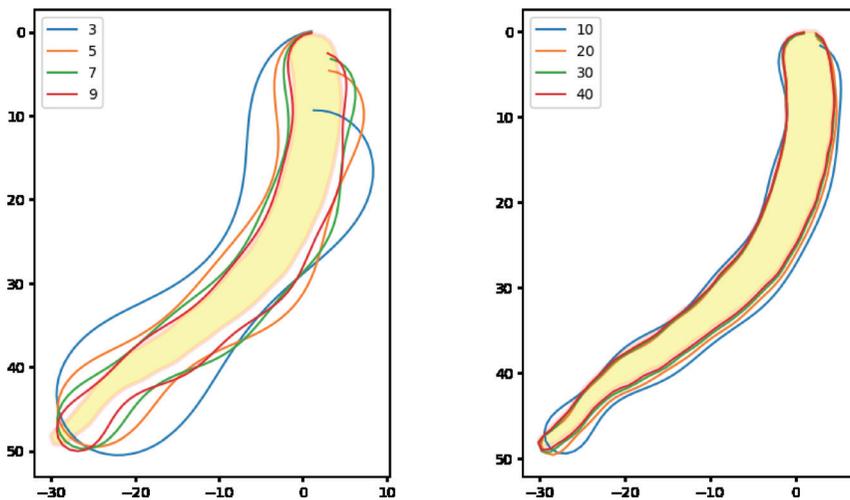


図3 Z型フーリエ記述子による復元形状

図3にZ型フーリエ記述子からの形状復元の例を示す。左は復元に用いる係数の最大次数を3,5,7,9とした場合、右は10,20,30,40とした場合である。次数が低いほど元の形状から離れると同時に、終点の位置が始点から大きくずれていく。

② 楕円フーリエ記述子

楕円フーリエ記述子も Z 型同様閉曲線を前提とした変換である。

曲線上に任意の点 S を定め、S から一定速度で曲線上を移動しながら周回する動点 P を考えると、点 P の x 座標、 y 座標はそれぞれ独立した周期関数とみなせる (図 4)。これをフーリエ級数展開することで得られる三角関数の係数が楕円フーリエ記述子である。

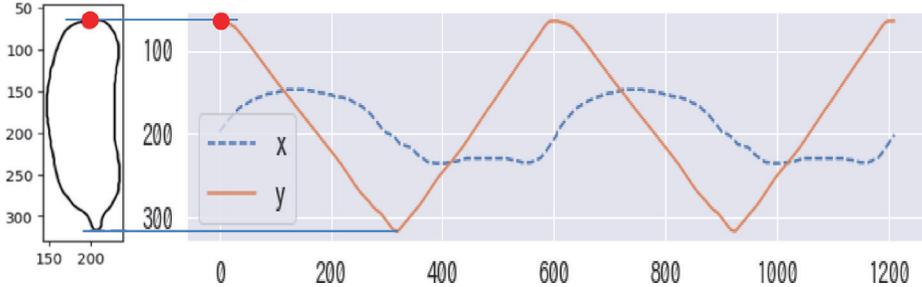


図 4 閉曲線上の動点の x 座標と y 座標

楕円フーリエ記述子は次に述べる G 型フーリエ記述子と数学的には同等である。G 型フーリエ記述子の方が楕円フーリエ記述子より理論の記述がコンパクトになるので、楕円フーリエ記述子の理論説明は省略する。

なお、楕円フーリエ記述子は回転不変性を持たない。 x 座標の周期関数と y 座標の周期関数の形が図形の回転によって変化するからである。このため、楕円フーリエ記述子を形状分析や識別等に用いる場合は、図形の近似楕円の方向と大きさを基準に記述子を正規化しなければならない。

③ G 型フーリエ記述子

楕円フーリエ記述子では、 x 座標と y 座標をそれぞれ別々に弧長の関数と考えたが、G 型フーリエ記述子では、 x 座標が実部、 y 座標が虚部であるような複素関数ととらえてフーリエ級数展開を施す。

平面座標 (x, y) と複素数 $z = x + jy$ を同一視して考える。

N 個の標本点、すなわち等辺 N 角形の頂点の集合は N 個の複素数の集合 $z_i; i = 0, 1, \dots, N - 1$ と同一視される。ただし、 $z_i; i = 0, 1, \dots, N - 1$ は閉曲線上をたどった時の出現順に並んでおり、先頭の頂点と末尾の頂点は、他の隣接頂点と同じ間隔だけ離れて隣接しているものとする。

いま、 $z_i; i = 0, 1, \dots$ を周期 N で 1 周する周期関数とみなし、 N は閉曲線を近似するのに十分な数であるものとするれば、

$$z_i = \sum_{k=-\frac{N}{2}+1}^{\frac{N}{2}} c_k \exp\left(\frac{j2\pi ki}{N}\right) \quad \dots(11)$$

とフーリエ級数展開できる。ここで、

$$c_k = \frac{1}{N} \sum_{i=0}^{N-1} z_i \exp\left(-j\frac{2\pi ki}{N}\right) \quad \dots(12)$$

である。

このとき、係数 c_k を **G 型フーリエ記述子** とよぶ。

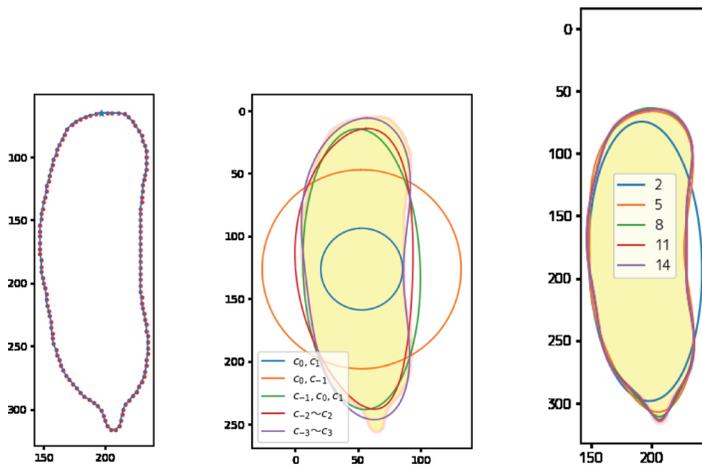


図5 G型フーリエ記述子から得られる復元形状

G 型フーリエ記述子の性質

以上のことをふまえ、G 型フーリエ記述子の特徴をまとめておく。

- 直流成分 c_0 は図形の重心位置を表す。
- c_0 以外の $|c_k|$ は、平行移動、回転に関する不変性を持つ。
- 高次成分をカットして逆変換することで近似曲線を得ることができる。 c_1 と c_{-1} のみを用いて逆変換すると原図形の近似楕円がえられる。
- この近似楕円の中心、軸方向、長径の大きさを基準として図形を正規化することで、G 型フーリエ記述子は平行移動、回転、スケーリングに対して不変となる。

近似楕円による正規化まで含めて考えれば、楕円フーリエ記述子も同様の不変性が成り立つ。

図5にG型フーリエ記述子をもとに逆変換して得られる近似図形の例を示す。左の図

が原図形，中央の図は記述子の組 $\{c_0, c_1\}, \{c_{-1}, c_0\}, \{c_0, c_{\pm 1}\}, \{c_0, c_{\pm 1}, c_{\pm 2}\}, \{c_0, c_{\pm 1}, c_{\pm 2}, c_{\pm 3}\}$ からの逆変換で得られる形状，右の図は，2次，5次，8次，11次，14次までの記述子からの逆変換で得られる形状である。なお，この実験で用いた標本点の数は128である。

④ P型フーリエ記述子

図6にP型フーリエ記述子の考え方を示す。P型フーリエ記述子は等辺N角形近似における辺の方向ベクトルの変化を周期関数とみなす。

まず図形を図6(a)のように等辺多角形近似する。任意の辺を最初の辺0として選び，その方向を基準方向と考える。辺0から順に近似多角形の辺を接続順に一定方向に巡回していくときに，出現する辺の方向ベクトルを始点が原点となるように複素平面上に描いていくと(b)のようになる。辺を巡って1周すると元の位置に戻るが，それ以降も辺をたどり続けると考えて，辺の方向ベクトルの列を周期関数とみなし，フーリエ級数展開する，というのがP型の基本的なアイデアである。

輪郭上に等間隔に $N+1$ 個の標本点 $P_i; i = 0, 1, \dots, N$ を取る。ただし， $P_N = P_0$ である。 $P_i = (x_i, y_i)$ に複素平面上の点 $z_i = x_i + jy_i$ を対応づけ，両者を同一視することにする。

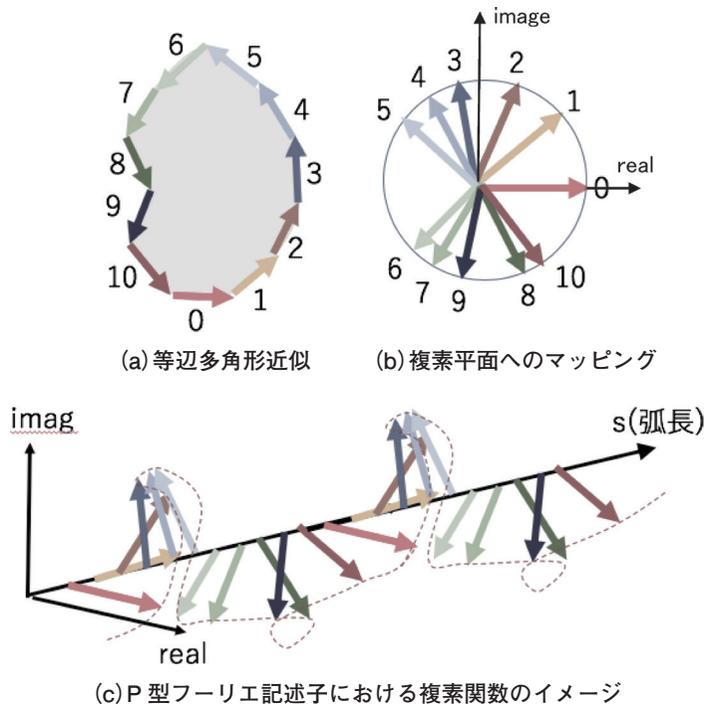


図6 P型フーリエ記述子

ここでは，ベクトル $\overrightarrow{P_i P_{i+1}}$ と x 軸（実数軸）のなす角 θ_i をベクトル $\overrightarrow{P_i P_{i+1}}$ の偏角とよ

ぶことにする。ここで、 $w_i = \cos\theta_i + j\sin\theta_i$ と定義すると、偏角 θ_i は複素平面上的の複素数 w_i の偏角に一致する。

いま、等辺 N 角形の一つの辺の長さを δ とすれば、

$$w_i = \cos\theta_i + j\sin\theta_i = \frac{x_{i+1} - x_i}{\delta} + j\frac{y_{i+1} - y_i}{\delta} = \frac{z_{i+1} - z_i}{\delta} \quad \dots(13)$$

である。

逆に、 $w_i: i = 0, 1, \dots, n$ と始点 z_0 が与えられたならば、 $z_n = z_0 + \delta \sum_{i=0}^n w_i$ として、近似多角形の頂点の位置が求まり、原図形は、位置・形ともに復元できる。また、始点 z_0 がわからなくても、任意の点を始点と定めることで、図形の形状は復元できる。

輪郭を巡回することを考えると、 $w_n: n = 0, 1, \dots$ は周期 N の複素周期関数とみなすことができるので、

$$w_i = \sum_{k=-\frac{N}{2}+1}^{\frac{N}{2}} W_k \exp\left(j\frac{2\pi ki}{N}\right) \quad \dots(14)$$

とフーリエ級数展開することができるはずである。ここで、

$$W_k = \frac{1}{N} \sum_{i=0}^{N-1} w_i \exp\left(-j\frac{2\pi ki}{N}\right) \quad \dots(15)$$

である。この、 W_k を **P 型フーリエ記述子** と呼ぶ。

P 型フーリエ記述子の性質

偏角は図形的位置に左右されないので、P 型フーリエ記述子は G 型の項で述べたのと同じ理屈により、回転、拡大縮小に関して不変である。この性質を含め、P 型フーリエ記述子は、

- 平行移動、回転、拡大縮小に関して不変である。
- 低い周波数のみを用いて逆変換をほどこすことで、近似形状が得られる。高周波を加えていくにつれて任意に詳細なレベルの形状変化まで再現できる。
- P 型には、Z 型の短所であった逆変換図形が閉曲線とならない、という欠点はない。
- 開曲線に対しても適用可能である。

などの特徴を有する。P 型フーリエ記述子は偏角をフーリエ変換するという意味では Z 型フーリエ記述子と考え方も結果として得られる性質も類似している。

図 7 に P 型フーリエ記述子の一部からの逆変換により得られた復元形状の例を示す。

凡例の意味はこれまでと同様なので説明は省く。記述子の一部を用いて得られる近似形状については G 型記述子の場合と大きな違いはない。近似図形の位置が異なるのは、P 型の結果が始点位置を基準に描かれているのに対し、G 型の結果は重心位置が保存されるような位置に描かれているからである。G 型は 0 次の記述子が重心を表すが、P 型は角度の変換であるため、重心位置の情報を含まない。

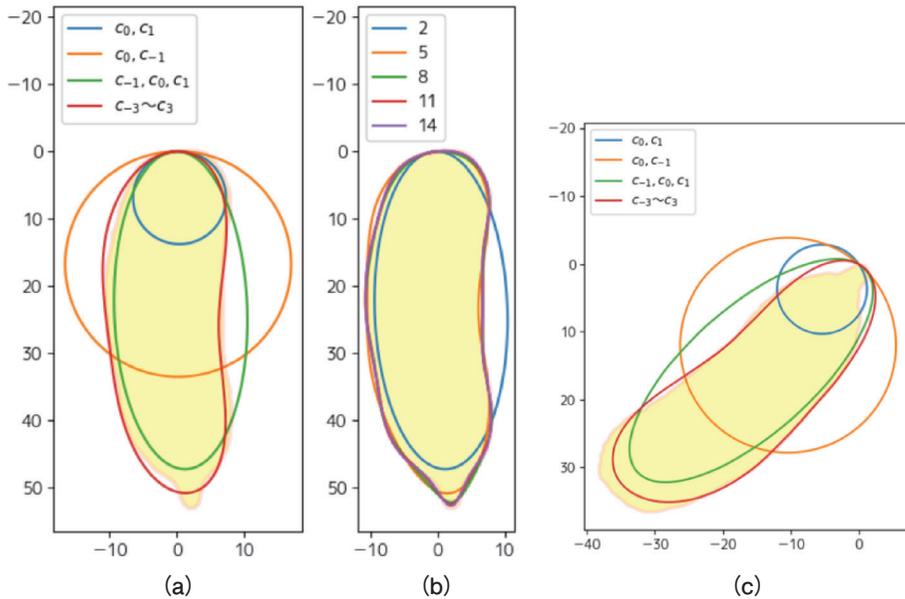


図7 P型フーリエ記述子から得られる復元形状の例

3. デジタル画像に対するフーリエ記述子の回転不変性の検証

検証実験に入る前に、フーリエ記述子が回転不変性をもつとされる理由を確認しておこう。

複素平面上の点 z を原点周りに θ 回転させた点を z' とすれば、

$$z' = z \exp(j\theta) \quad \dots(16)$$

と表せる。N 個の標本点全体、 $z_i: i = 0, 1, \dots, N-1$ を原点周りに θ 回転させたものを $z'_i: i = 0, 1, \dots, N-1$ として、そのフーリエ記述子を c'_k とすれば、

$$c'_k = \frac{1}{N} \sum_{i=0}^{N-1} z'_i \exp\left(-j \frac{2\pi k i}{N}\right) = \frac{\exp(j\theta)}{N} \sum_{i=0}^{N-1} z_i \exp\left(-j \frac{2\pi k i}{N}\right) = c_k \exp(j\theta) \quad \dots(17)$$

となり、回転後の記述子は回転前の記述子を原点周りに θ 回転させた値をとることがわかる。回転によって変わるのは位相だけで絶対値は変化しない。原点以外を中心とした回転については、原点周りの回転と平行移動に分解して考える。平行移動は0次の項にしか影響せず、1次以上の記述子の不変性は損なわれない。

楕円フーリエ記述子は x 座標と y 座標を別々に周期関数化する。それらの関数は回転によって形状が変化してしまうため、そのままでは回転不変とならないが、応用の際には1次の項から近似楕円を算出し、その軸方向と径で記述子を正規化することで2次的に不変な記述子を得る。

以上がフーリエ記述子における回転不変性の理論的説明である。

この理論は、回転前と回転後とで図形輪郭上の同じ点が標本点として選ばれていることを前提としていることに注意が必要である。現実には標本点の組み合わせは始点位置に左右される。回転に不変な始点の選び方があればいいが、一般的な方法はない。

さらに、デジタル画像においては標本点の位置は格子点しか選べない。回転角によって輪郭形状はわずかでもブレが生じる。近似精度を上げるために標本間隔を狭めると、逆に角度の量子化誤差が大きくなってしまう。

以下では、実用例の多い、G型とP型フーリエ記述子を対象として回転不変性の検証を行う。

① G型フーリエ記述子の回転不変性

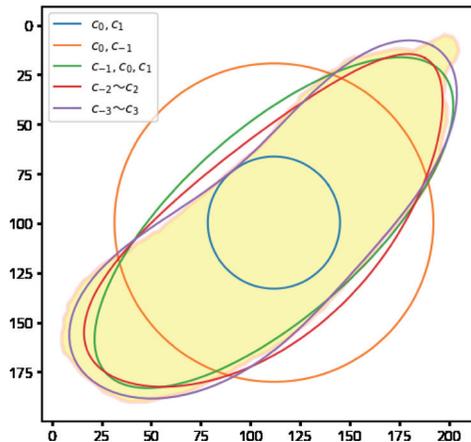


図8 G型フーリエ記述子からの復元形状（回転させた場合）

図8に図5と同じ図形を反時計回りに130度回転させた図形とその復元結果を示す。

目視では図5 (b)を回転させれば、ほぼ図8に一致しそうに見える。単なる概形記述が目的であるならばそれで十分であるが、シルエット解析やフーリエ記述子による識別が目的であるならば話は変わる。シルエット解析や識別においては、記述子そのものの類似性が問われるからである。復元形状がオリジナルの概形を捉えていることは必要条件にすぎない。記述子を用いた解析や識別においては、特徴ベクトルとしての記述子そのものが回転や拡大縮小などの幾何学的操作に対して不変であるかどうかの方が重要である。

表1 回転による記述子の変化

| 記述子 | 図5の図形 | | 図8の図形 | |
|----------|---------|-------|---------|-------|
| | 偏角(deg) | 大きさ | 偏角(deg) | 大きさ |
| c_0 | 67.3 | 17546 | 41.7 | 19173 |
| c_{-1} | -90.6 | 10154 | -49.0 | 10288 |
| c_1 | 93.2 | 4173 | -34.5 | 4266 |
| c_{-2} | -35.2 | 330 | 179.4 | 464 |
| c_2 | 16.4 | 518 | -97.8 | 469 |

表1に、図5と図8について、2次までのG型フーリエ記述子の実際の値を示す。G型フーリエ記述子は複素数であるので、大きさと複素平面における偏角を示した。

まず、G型フーリエ記述子は回転不変とされるが、不変なのは複素数としての記述子の大きさ(絶対値)であって、位相は回転に伴って変化する。1次の記述子(c_1, c_{-1})は近似楕円を規定する量であるが、図5と図8で、 c_{-1} は約40°、 c_1 は約140°のずれがある(40°と140°は方向としては同一と考えられる)。実際には図8は図5を130°回転させて描かれているのだが、130°に対して約140°なので、誤差の範囲と言えるかもしれない。大きさについては c_{-2} 以外は±10%程度の誤差の範囲に収まっているが、 c_{-2} は図5が330であるのに対して、130°回転した図8は464と約4割増しとなっている。これは一例にすぎないが、フーリエ記述子の回転不変性が実際のデジタル画像では必ずしも成立していないことを示すには十分である。

次に、フーリエ記述子の不変性がどの程度成立しているのか、逆に回転角によってどのように変化するかをもっと詳細に調査してみる。

先の図形について、5度単位で画像を回転させた場合の記述子の変化を図9に示す。回転角をグラフのx軸からの角度に、記述子の絶対値を中心からの距離に対応させてプロットしてある。次数によって振幅に大きな差異があるので、2つのグラフに分けて示してある。 c_2 と c_{-3} は両方のグラフに示した。

この図形の場合、楕円近似を表す c_{-1} と c_1 は図形を回転させていっても変動はわずか

デジタル画像におけるフーリエ記述子の回転不変性の検証

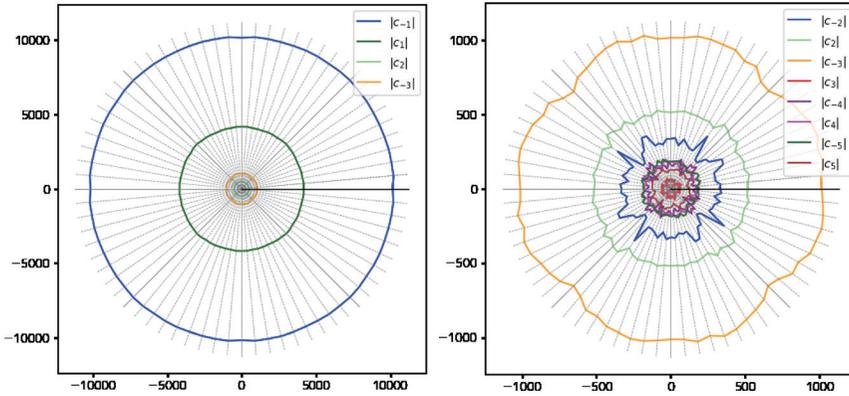


図9 回転角によるG型フーリエ記述子の変動

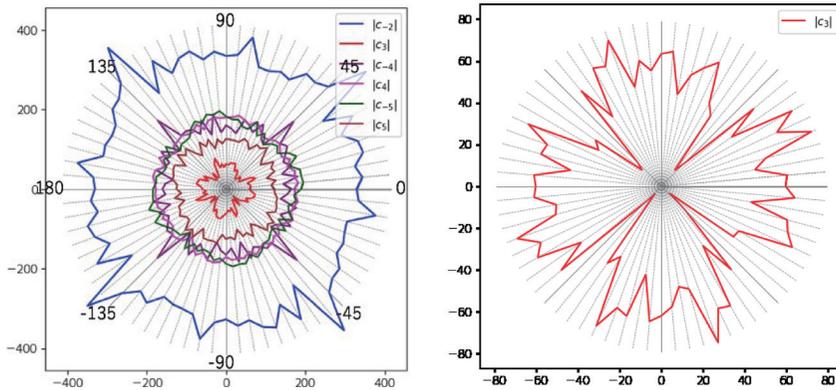


図10 回転角による変動幅の大きい記述子 (G型)

である。また、 c_2 と c_{-3} も比較的安定している。この程度の変動であれば回転不変と言ってもよいかもしれない。それ以外の記述子は回転角に対する変動が結構大きく、グラフを見ると概ね90度周期の変化がみて取れる。

図10に記述子のうちで絶対値の小さい項だけ抜き出したグラフを示す。図5(a)の図形の場合、 $|c_{-2}|, |c_3|$ の変動幅が大きく、特に45+90n度 ($n = 0, 1, 2, \dots$) 付近で著しく変化する。これはおそらくデジタル画像では画素が格子状に配置されていることと関係するもので、避けられない現象であろうと推察される。

1例だけではこの図形に限られた現象の可能性があるので、複数の図形について同様の実験を試みた。1次の記述子に関してはどの図形も最初の例と同様に回転に対して安定した値となる。 c_2 と c_{-3} も比較的安定している。最初の例で観察されたのと同様、90度周期で記述子の値が変動することも確認できた。追加実験の結果例を図11に示す。

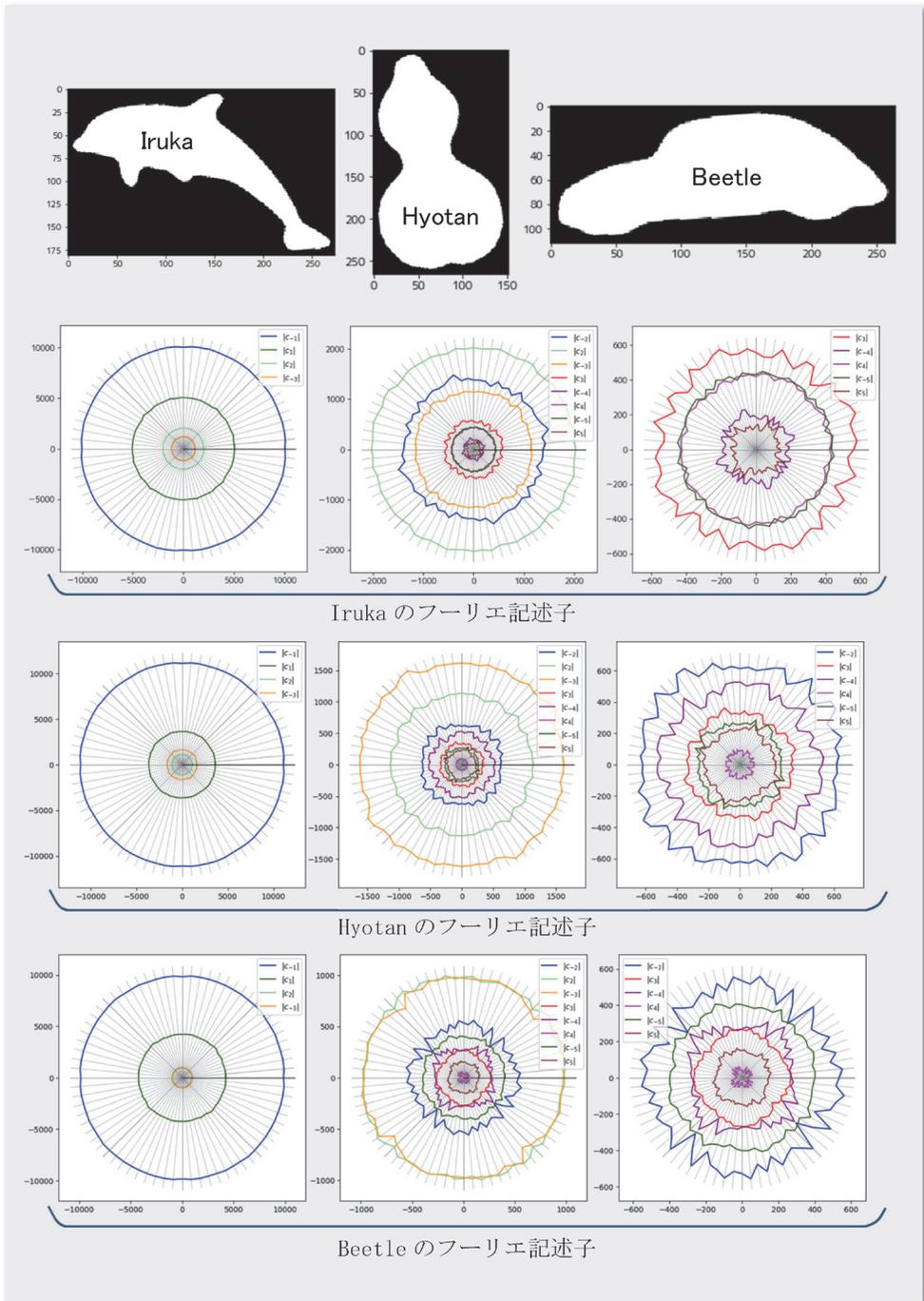


図11 回転角による G 型フーリエ記述子の変動 (Iruka, Hyotan, Beetle)

② P 型フーリエ記述子の回転不変性

P 型フーリエ記述子についても G 型同様の方法で回転不変性を調査した。

130度回転させた場合の形状復元結果については、すでに図7(c)に示してある。図7(a)と(c)とで近似図形の描画位置が違っているのは、両者の標本点の始点位置が異なるためである。先にも述べたが、P型フーリエ記述子には位置情報が含まれないため、近似図形は標本点の始点を基準として描いている。標本点の始点はラスタスキャンした際に最初に見つかる点であり、回転に伴って始点として選ばれる箇所そのものが変化する。目視では、近似形状自体は回転前とほぼ同一で、ちょうど130度回転しているように見える。

G型のとくと同様に、5度単位で画像を回転させた場合の記述子の変化を調べてみた。結果を図12～図14に示す。 c_{-1} と c_1 についての結果をG型と比べると、G型の結果のグラフが真円に近いのに対し、P型の結果は正八角形に近いグラフとなっているのがわかる。これはG型と比べると回転不変の度合いが若干下がることを意味するが、正八角形と円の差はわずかであるので、 c_{-1} と c_1 についてはほぼ回転不変性が成り立つと言って差し支えなからう。 c_2 と c_{-3} が他の高次の項と比べて安定している点も同じである。記述子の値に90度周期での変動がある点もG型と同じである。

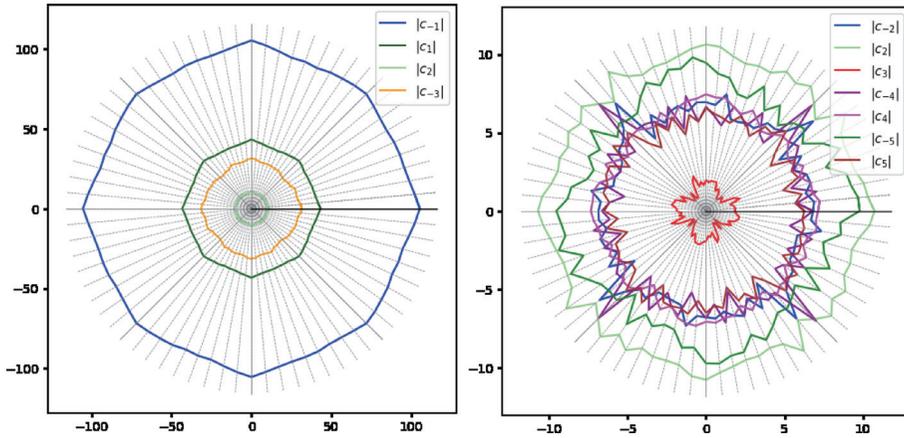


図12 回転角によるP型フーリエ記述子の変動

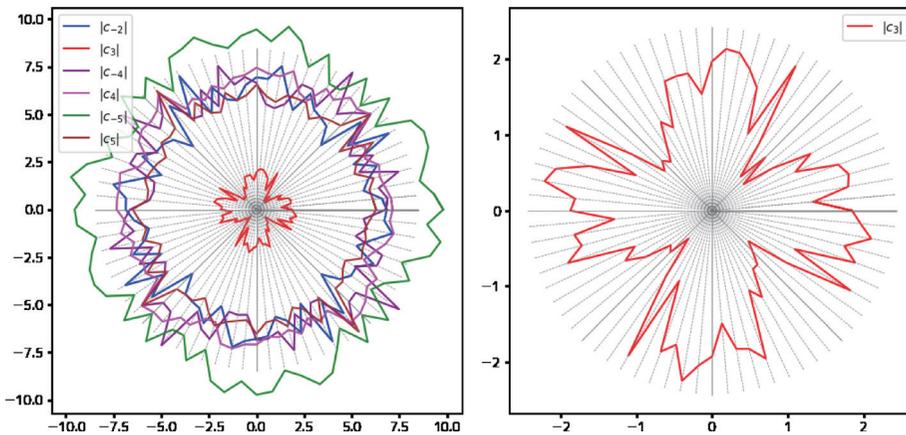


図13 回転角による変動幅の大きい記述子 (P型)

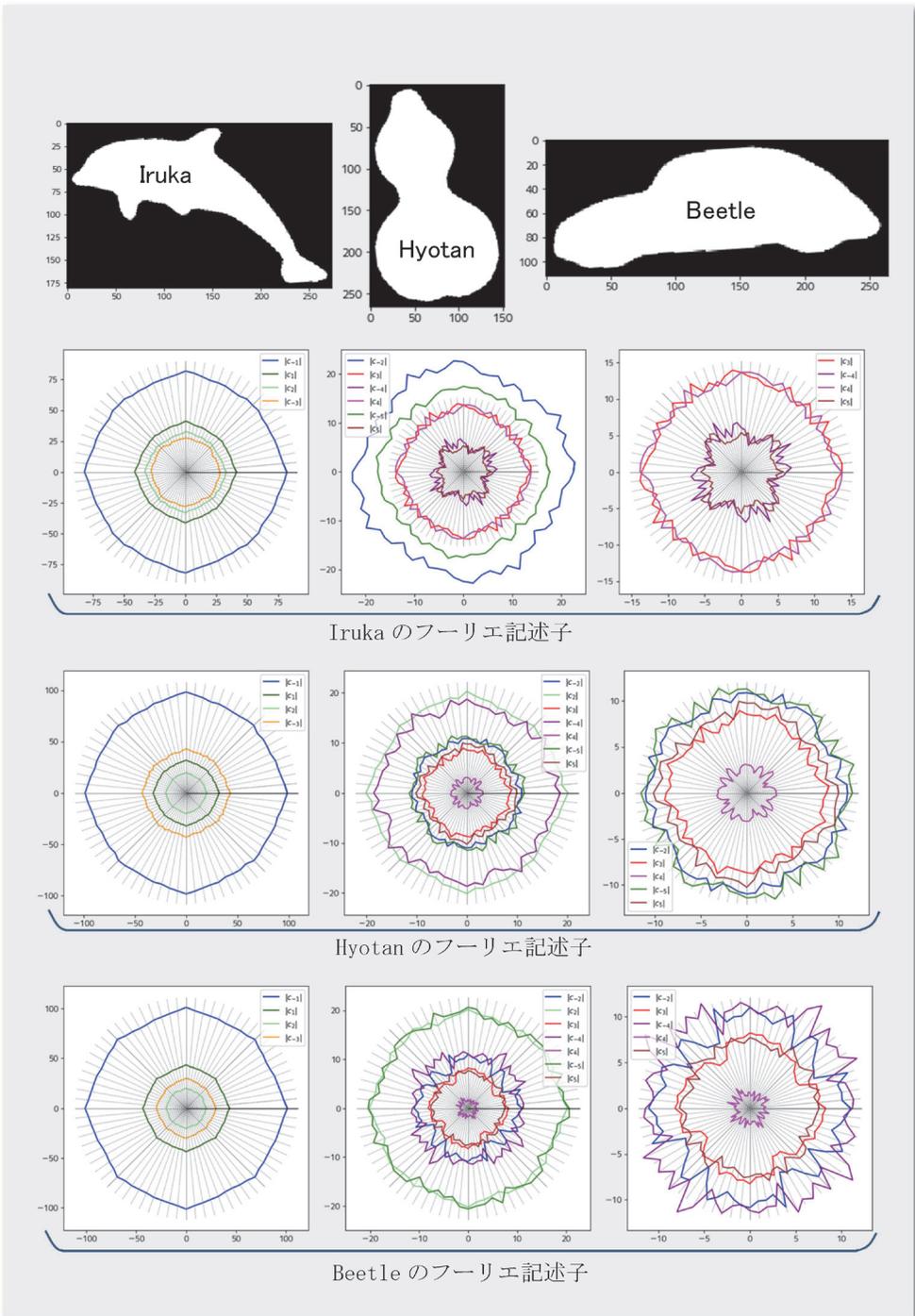


図14 回転角によるP型フーリエ記述子の変動 (Iruka, Hyotan, Beetle)

4. まとめ

フーリエ記述子のうち回転不変性を有する G 型と P 型について、デジタル画像を対象とした時に回転不変性がどの程度成立するのかを実験的に検証してみた。

実際にシルエット画像から G 型と P 型のフーリエ記述子を求めるプログラムと、ローパスフィルタを通した逆変換によって近似図形を生成するプログラムを作成し、いくつかの図形を 5 度刻みで回転させ、回転に対してフーリエ記述子の各項がどう変化するかを調べた。

実験の結果をまとめると以下のとおりである。

- (1) 1 次の記述子については、G 型、P 型ともに、回転不変性が認められる。ただし、G 型と P 型の比較では、G 型の方が回転に対する変動が少ない。G 型は 360 度どの方向に回転しても記述子の値がほとんど変化しないが P 型は特性グラフが正八角形に近い形になる。
- (2) 2 次以上の記述子は、G 型、P 型ともに角度によって大きく変動する場合がある。どの次数の記述子が増動するかは図形によってまちまちであるが、角度によっては同じ記述子の値の変動が最大最小比で 2 倍以上となることもある。変動は回転角に対して 90 度周期となる。

以上の結果より、デジタル画像に対して、フーリエ記述子の回転不変性を盲目的に信じて利用するのは危険であると結論づけられる。特にフーリエ記述子を識別のための特徴として用いる場合、まず楕円近似をして図形の方向を正規化した上で記述子を再計算するとか、0～90 度の範囲で回転させたときの記述子の平均値を特徴として採用するなどという措置が必要であろう。

今回の実験はフーリエ記述子のうちの G 型、P 型のみについて調査したが、90 度周期での変動はほかの記述子についても起こるであろうと推測できる。楕円フーリエ記述子を応用する場合に正規化が行われるという話を述べたが、その正規化といままで述べた正規化とは別である。楕円フーリエ記述子の場合には正規化してはじめて理論的に回転不変となる。G 型の実験結果は正規化後の楕円フーリエ記述子の振る舞いに相当する。

どの程度の次数の記述子までなら安定的であるのか、どれ程の振幅で記述子の値が増動するのかは、画像の解像度や量子化間隔、図形の複雑さなどに左右されるものと予想される。今回の実験に用いたプログラムでは、シルエットを囲む矩形の長辺が 256 画素となるように正規化した図形を用い、標本点の数を 128 とした。これは著者が神戸大学との共同研究として行っているダイコンの形状解析で用いている数値であり、特に粗すぎるわけではないことを記しておく。

参考文献

- [1] 鄭澤宇, 岩田洋佳, 二宮正士, 田村義保: “P形フーリエ記述子に基づくハナハス花弁の部分形状特徴の定量的評価”, 育種学研究 7, 3, pp.133-142 (2005).
- [2] 古田尚也, 二宮正士, 高橋信夫, 大森 宏, 鷓飼保雄: “ダイズ小葉形の楕円フーリエ記述子に基づく主成分スコアによる定量的評価”, Breeding science 45, 3, pp.315-320 (1995).
- [3] 川村陽子: “フーリエ記述子を用いた脳梁形状の性差の解析”, 岐阜大学工学部 博士学位論文 (2006).
- [4] 中嶋典子, 石井晃太郎, 岩田洋佳, 大鹿哲郎: “楕円フーリエ記述子に基づく水晶体形状発達の定量的解析”, 臨床眼科 65, 7, pp.137-1142 (2011).
- [5] 小林智美, 篠田一馬, 石川智治, 長谷川まどか, 加藤茂夫: “楕円フーリエ記述子を用いた衣服の輪郭形状の分類に関する一検討”, 映像情報メディア学会技術報告 39, 31, pp.5-8 (2015).
- [6] 小森政嗣, 川村 智, 横山卓未, 森下佳昌: “楕円フーリエ記述子を用いたボディソープボトル形状の分析と評価”, 日本包装学会誌 21, 6, pp.479-491 (2012).

組込 CPU における割込・分岐処理の 高速化に関する研究

岩 出 秀 平

Study of High-speed Embedded CPU for EIT Processing

ShuheI Iwade

【Abstract】

This study is concerned with the high-speed operation technology of embedded CPU for EIT processing. EIT is the abbreviation of Exception, Interrupt and Trap. The aim of the high-speed technology is the reduction of the number of data transfer instructions which are related to saving or setting data. In order to achieve the purpose, three technologies are developed. First, BPC file which includes plural BPCs is developed. BPC is the return address register and the abbreviation of Backup Program Counter. Secondly, the register file bank is introduced. A register bank consists of plural register files and each register file includes 2^n registers. These hardware can be controlled by a pointer which is generated by the file pointer developed. As a result, data transfer instructions for saving and setting both BPC and registers are reduced to zero. Additionally, the global registers are designed in order to transfer data before and after a branch interaction. The novel CPU which is equipped with these three hardware has been simulated by HDL simulator and can be assured the normal operation. Further the logic synthesis has been carried out with QuartusII for the novel CPU and the normal CPU. As a result, the novel CPU is proved to be the available for embedded application.

研究概要

本研究はリアルタイムシステムにおける割込処理に対応する組込用 CPU の高速化に関するものである。高速化の狙いは割込処理時に必要となるレジスタ等の退避や復帰処理に係る命令の削減である。そのため割込処理プログラムからの戻り先アドレスを保存する BPC (Backup Program Counter) レジスタを複数個保存可能な BPC ファイルを開発し、レジスタの集合体であるレジスタ・ファイルを多バンク化し、これらのハードウェアをポインタで制御可能にした。そして割込や復帰に対応してポインタを発生するファイル・ポインタを開発した。この結果、割込や復帰時の BPC やレジスタの退避・復帰のための命令数を 0 にすることに成功した。また分岐前プログラムと分岐後プログラムとの間でデータをやり取りする機能を付加するため、このポインタを利用したグローバル・レジスタも考案した。以上のハードウェアを搭載した新方式 CPU に対して HDL シミュレーションを実施し、全ての命令の正常動作を確認した。最後に QuartusII で新方式 CPU と旧方式 CPU の論理合成をおこない、論理合成レポートを比較した結果、バンク化によるレジスタ数の増加は想定を下回り且つ遅延時間に変化がないことが判明し、新方式 CPU がリアルタイムシステムに有効であることが分かった。

1. はじめに

CPU における割り込み（以下割込）にはタイマやセンサ等、外部ハードウェアから CPU に割込をかける外部割込とプログラム中の割込命令により自ら割込をかけるソフトウェア割込があり、割込処理後に元のプログラムに復帰する。

サブルーチン処理に使われる分岐命令は、分岐後に元のプログラムに復帰する機能をもつリンク付き分岐命令であり、ハードウェア構成の観点から割込処理と同一である。

割込・分岐命令による割込・分岐処理終了後に元のプログラムに復帰して処理を継続するには以下の条件が必要である。

- ① 命令メモリの戻り先アドレスが保存されていて、そのアドレスに分岐できること。
- ② レジスタ・ファイルのデータが、割込や分岐が起きる前の値になっていること。

戻り先アドレスは、割込や分岐を起こした命令の次の命令のアドレスで BPC というレジスタに保存する。レジスタ・ファイルは、2 のべき乗個のレジスタから成るレジスタの集合体で一般的には 16 個、32 個または 64 個等である。上記①②の条件を満たすためには、BPC やレジスタ・ファイル内の全レジスタの値を割込や分岐処理をする前にどこかに退避し、戻るときに復帰させなければならない。

これを実現するために従来の CPU ではデータメモリ内にスタック領域を設けて、割込や分岐が起こったときには BPC とレジスタ・ファイルの値をスタックに退避（ストア）

させ、戻るときにはスタックから復帰（ロード）させている。例えば三菱電機の M32R プロセッサでは、汎用で使われる15番レジスタ R15をスタックポインタに固定し、R15のデクリメントをハードウェアで自動的に行って ST Rsrc1,@R15命令を繰り返すことにより R15番地のスタックへの退避を、また R15のインクリメントをハードウェアで自動的に行って LD Rdest,@R15+ 命令を繰り返すことにより R15番地のスタックからの復帰を行っている[1]。MIPS Technologies 社の MIPS プロセッサでは、スタックポインタ :sp を汎用レジスタとは別に備え、sw \$s0 0 (\$sp), sw \$s1 4 (\$sp), sw \$s2 8 (\$sp)・・・と固定した SP に相対アドレス（0 や 4 や 8・・・）を命令のオペランドで与えることにより、それぞれの番地のスタックへの退避を、復帰させる場合は、lw \$s0 0 (\$sp), lw \$s1 4 (\$sp), lw \$s2 8 (\$sp)・・・のように sw と同様に命令のオペランドで与えることにより、それぞれの番地のスタックからの復帰を実現している[2]。スタック技術は1980年頃に提案され現在でもスタックに関する研究は行われている。[3]-[11]

具体的には割込・分岐処理前に、例えば16個のレジスタから成るレジスタ・ファイルの場合、BPC と合わせて17個のデータをスタック領域にストアし、処理が終わると同じく17個のデータをスタックからロードしなければならない。即ち1つの割込または分岐に対して（割込や分岐処理と無関係な）計34個のレジスタとデータメモリ間の転送命令を実行する必要がある。リアルタイム組込 CPU では割込や分岐の回数が多く、次の割込まで前割込処理が終わらず次の割込を取りこぼすことが問題となっているので、スタック方式を適用すると本来の割込処理命令に加えて上記復帰・退避命令のために全体の命令数が増加し、割込処理速度を更に低下させることになる。特にリアルタイム処理に使われる比較的小規模なマイクロプロセッサではコストや消費電力の制約があるため最先端の超高速デバイスが使えず、上記 M32R や MIPS も含めて命令数の増加による処理速度の低下をデバイスで吸収することは難しい。

本研究では、スタック方式の弱点を改善するため退避や復帰時の転送命令を使用せず割込・分岐命令で BPC やレジスタ・ファイルを保存できる新方式のハードウェア・アーキテクチャを考案し、16bit-CPU に実装して動作を確認した。また同規模のスタック方式 CPU も設計して QuartusII で論理合成を行い、論理合成結果の比較を行った。

以下では従来方式であるスタックによる退避・復帰方式の動作について説明した後に新方式ハードウェア・アーキテクチャ、それを実装した新方式 CPU 回路の設計および検証結果について述べる。

2. スタック方式による従来割込・分岐処理技術

表2.1に従来型 CPU（以下スタック方式 CPU）の命令セットを示す。表2.1において、塗りつぶしで強調されている命令や機能は、新方式 CPU にはないスタック方式 CPU の

命令や機能である。表2.1で強調されていない命令や命令表記は、新方式CPUと同じ命令または命令表記である。

表2.2にスタック方式CPU回路を実行順に5段に分割したパイプラインを示す。表2.1の各命令は、表2.2のパイプラインに従って順に実行される。例えばn番目の命令がDステージに入ったとき、n+1番目の命令はFステージに投入される。

表2.1で強調されている命令について説明する。

表2.1 スタック方式CPU命令セット

| 命令分類 | 命令表記 | 命令機能 |
|------|-------------------------|--|
| 演算命令 | ADD Rdest Rsrc | $Rdest \leftarrow Rdest + Rsrc$ |
| | SUB Rdest Rsrc | $Rdest \leftarrow Rdest - Rsrc$ |
| | AND Rdest Rsrc | $Rdest \leftarrow Rdest \& Rsrc$ |
| | OR Rdest Rsrc | $Rdest \leftarrow Rdest Rsrc$ |
| 転送命令 | LD Rdest @(Rsrc disp4) | $Rdest \leftarrow M[Rsrc + (disp4)_{16}][7:0]$ |
| | LDI Rdest imm9 | $Rdest \leftarrow (imm8)_{16}$ |
| | POP Rdest | $Rdest \leftarrow M[SP]; SP++$ |
| | LDbpc | $BPC \leftarrow M[SP]; SP++$ |
| | ST Rdest @(Rsrc disp4) | $M[Rsrc + (disp4)_{16}][7:0] \leftarrow Rdest$ |
| | PUSH Rdest | $SP--; M[SP] \leftarrow Rdest$ |
| | STbpc | $SP--; M[SP] \leftarrow BPC$ |
| 分岐命令 | BEQ Rsrc1 Rsrc2 pcdisp4 | $if(Rsrc1==Rsrc2) \quad PC \leftarrow PC + (pcdisp4)_{12}$ |
| | BRA pcdrect12 | $PC \leftarrow pcdrect12$ |
| 割込命令 | TRAP pcdrect3 | $BPC \leftarrow PC[TRAP]+1; \text{BRA } pcdrect3; \text{次命令}=\text{NOP}$ |
| | RTE | $PC \leftarrow BPC$ |
| | NOP | No Operation |

表2.2 スタック方式CPUのパイプライン

| ステージ名 | 記号 | 内容 |
|----------|-----------------------|-------------------------|
| 命令フェッチ | F(instruction Fetch) | 命令取出;BPC読出書込 |
| 命令デコード | D(instruction Decode) | 命令解読;アドレス/データ準備;次アドレス設定 |
| 命令実行 | E(Execution) | 演算実行;スタックポインタ生成 |
| メモリ・アクセス | M(Memory access) | データメモリ読出書込 |
| ライトバック | W(Write back) | レジスタへのデータ書込;BPC←戻り番地 |

2.1 スタック関連命令の機能

2.1.1 TRAP 命令 (ソフトウェア割込)

TRAP 命令機能で、PCは発行した命令のアドレスを保持するレジスタである。PC [TRAP]はTRAP命令のアドレス、PC[TRAP]+1はTRAP命令の次命令のアドレスである。BRA pcdrect3は分岐命令のBRA命令で、 $PC \leftarrow pcdrect3$ のことであるので次のclkが入ると

pcdrc3番地の命令が実行される。次命令 =NOP は F ステージに読み出された TRAP 命令の次命令を無効にすることである。

TRAP pcdrc3は、表2.2の D ステージで解読され、以下の順に処理が行われる。

- ① F ステージで TRAP 命令の次命令のアドレス：PC[TRAP]+1を BPC に保存する。
- ② D ステージで TRAP 処理への分岐命令が格納されている pcdrc3番地へ BRA pcdrc3により無条件分岐させるため、pcdrc3番地を次アドレスとして設定する。
- ③ TRAP 命令が D ステージに移行したため F ステージに発行された TRAP 命令の次命令を無効化する (NOP 命令にする)。

2.1.2 RTE 命令 (割込・分岐からの復帰)

RTE 命令機能で、PC ← BPC は、戻り先アドレスを PC に入力することを示す。

RTE 命令は、表2.2の D ステージで解読され、F ステージに対して戻り先アドレスが格納されている BPC を PC に入力する指示を出し、次の clk で戻り先アドレスの命令 (TRAP 命令の次の命令) が発行され割込処理プログラムから元の実行プログラムに戻る。

2.1.3 PUSH 命令 (スタックへのレジスタ退避)

PUSH 命令機能で、SP はスタックポインタで、スタック領域のアドレスを保持するレジスタである。SP-- は、SP を 1 だけデクリメントすることを示す。M はデータメモリで、M[SP]は SP 番地のデータである。M[SP]← Rdest は Rdest の値をデータメモリの SP 番地に書込むことを示す。Rdest の R はレジスタで dest はレジスタ番号 (0~15) である。

PUSH Rdest は、表2.2の D ステージで解読され、以下の順に処理が行われる。

- ① E ステージで SP を 1 だけデクリメントする。
- ② M ステージでデータメモリの SP 番地に Rdest を保存する。
- ③ 例えば、 $0 \leq \text{dest} \leq 15$ の場合、R0~ R15の各々について上記①→②を繰り返す。

2.1.4 STbpc 命令 (スタックへの BPC 退避)

STbpc は、PUSH と同じく表2.2の D ステージで解読され、E ステージで SP をデクリメントして、M ステージでデータメモリの SP 番地に BPC を退避させる。

2.1.5 POP 命令 (スタックからのレジスタ復帰)

POP 命令機能で、SP++ は、SP を 1 だけインクリメントすることを示す。

POP Rdest は、表2.2の D ステージで解読され、以下の順に処理が行われる。

- ① M ステージでデータメモリの SP 番地のデータを Rdest に保存する。
- ② E ステージで SP を 1 だけインクリメントする。
- ③ 例えば、 $0 \leq \text{dest} \leq 15$ の場合、R0~ R15の各々について上記①→②を繰り返す。但

し PUSH 命令が R0 から R15 までレジスタ番号の昇順に退避させていた場合、POP 命令は R15 から R0 までレジスタ番号の降順に復帰させることになる。

2.1.6 LDbpc 命令 (スタックからの BPC 復帰)

LDbpc は、POP と同じく表2.2の D ステージで解読され、M ステージでデータメモリの SP 番地のデータを BPC に復帰させた後に E ステージで SP をインクリメントする。

2.2 スタック方式による割込処理

スタック方式 CPU の TRAP 命令による割込処理の例を図2.1に示す。図2.1において、実行プログラムの100番地の TRAP 5命令が発行されると5番地に分岐し、5番地の BRA 500命令により500番地に分岐する。500番地の STbpc 命令が BPC をスタック領域に、PUSH Rn (n=0~15) 命令が Rn レジスタの内容をスタック領域にそれぞれプッシュした後、命令 A からの割込処理が開始される。命令 Z で割込処理が終了すると、POP R15 からレジスタ番号の降順にスタックの内容を Rn レジスタに、LDbpc 命令がスタック領域から BPC にそれぞれポップした後、RTE 命令で割込処理からのリターンが実行されて元の実行プログラムの101番地の次命令に戻る。このようにスタックでは最後に退避したものを最初に復帰させるので、スタックは Last In/First Out の LIFO メモリである。

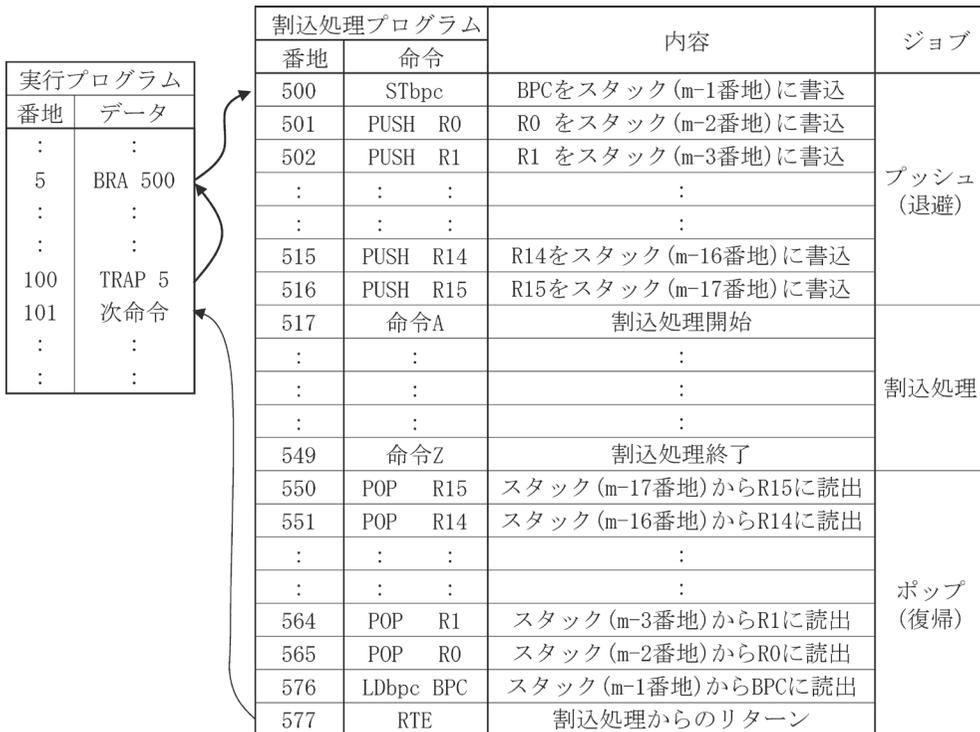


図2.1 スタック方式 CPU の TRAP 命令による割込処理 (SP 初期値 = m の場合)

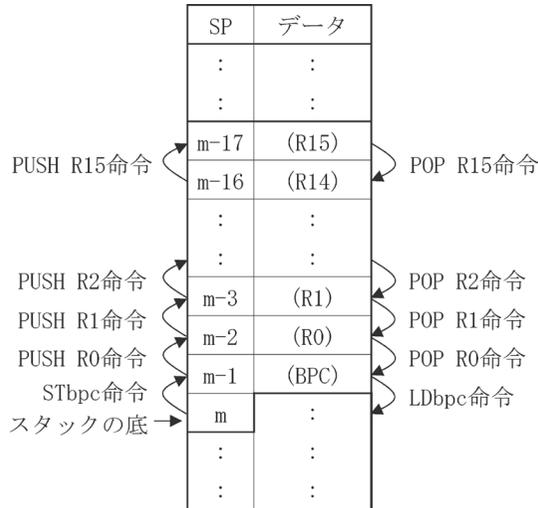


図2.2 図2.1の割込処理におけるスタック領域

図2.2は、図2.1の割込処理においてスタックの底が m 番地のときのデータメモリの状態である。図2.1の STbpc 命令により図2.2において $SP=m$ であった SP が1だけデクリメントされ、 $m-1$ 番地に BPC の内容が退避される。次の PUSH R0命令により SP が更に1だけデクリメントされ、 $m-2$ 番地に R0の内容が退避される。このように BPC と R0～R15までの全てのレジスタの内容が退避された結果 $SP=m-17$ 番地になっている。

割込処理（命令 A～命令 Z）が終了し、元の実行プログラムに戻るために図2.1における POP R15命令が実行されると、 $SP=m-17$ 番地のデータが読み出され R15に書き込まれた後、 SP は1だけインクリメントされ、 $SP=m-16$ 番地となって次の POP 命令を待つ。このようにして $m-1$ 番地にある最後のデータ BPC が復帰されると SP は1だけインクリメントされて $SP=m$ 番地に戻る。

3. 従来技術の問題点

前項2.で述べたことをまとめると、従来のスタック方式には以下の欠点がある。

- ① 割込・分岐が発生する度に戻り先アドレスやレジスタ群の退避・復帰命令を実行しなければならないため、ジョブを実行するための命令数が増加し、それによって処理速度が劣化する。
- ② 退避や復帰にはスタックポインタの管理が伴うので、通常のロード命令やストア命令では対応できず、スタック専用のストア命令（STbpc, PUSH）とロード命令（LDbpc, POP）を設けなければならない。リアルタイム処理対応の CPU では命令長が16ビットと短い場合が多く、命令の種類を決めるオペコードに割り当てるビッ

ト数が限られていて命令を増やせない場合が多い。従って4命令の追加により他の有用な命令を割愛せざるを得ない場合にはCPUの機能全体に悪影響を与える。

以上の問題を解決するため、スタック専用のストア命令やロード命令を必要としない回路方式を内蔵したCPUを設計した。

4. 新方式ハードウェア・アーキテクチャ

新方式CPUに適用された新方式ハードウェア・アーキテクチャについて述べる。

表4.1に新方式CPUの命令セットを示す。表4.1において、塗りつぶしで強調された命令や機能は、スタック方式CPUにはなく新方式ハードウェア・アーキテクチャを実現する新規命令（BL命令）、または命令表記は同じでもスタック方式CPUと異なる新規機能に変更した命令（TRAP命令、RTE命令）である。

表4.1の横ストライプで強調されたLDRR、LDRG、LDGR命令は、新方式ハードウェア・アーキテクチャ導入によるCPUのプログラミングを容易化するために新規に追加した副次的な命令でレジスタ間コピー命令である。表4.1で強調されていない命令は、スタック方式CPUと同じ命令である。 $(imm8)_{16}$ は8ビットのimmを16ビットに、 $(disp4)_{16}$ は4ビットのdispを16ビットに、 $(pcdisp4)_{12}$ は4ビットのpcdispを12ビットに、それぞれ符号拡張したものである。

以下では新方式ハードウェア・アーキテクチャに特化したパイプライン、新方式ハードウェア・アーキテクチャを実現する命令（表2.1で強調されている命令）の機能、それらの割込処理、そしてレジスタ間コピー命令の機能について述べる。

表4.1 新方式CPU命令セット

| 命令分類 | 命令表記 | 命令機能 |
|------|-------------------------|---|
| 演算命令 | ADD Rdest Rsrc | $Rdest \leftarrow Rdest + Rsrc$ |
| | SUB Rdest Rsrc | $Rdest \leftarrow Rdest - Rsrc$ |
| | AND Rdest Rsrc | $Rdest \leftarrow Rdest \& Rsrc$ |
| | OR Rdest Rsrc | $Rdest \leftarrow Rdest Rsrc$ |
| 転送命令 | LD Rdest @(Rsrc disp4) | $Rdest \leftarrow M[Rsrc + (disp4)_{16}][7:0]$ |
| | LDI Rdest imm8 | $Rdest \leftarrow (imm8)_{16}$ |
| | LDRR Rdest Rsrc | $Rdest \leftarrow Rsrc$ |
| | LDRG Rdest Gsrc | $Rdest \leftarrow GRsrc$ |
| | LDGR Gdest Rsrc | $GRdest \leftarrow Rsrc$ |
| | ST Rdest @(Rsrc disp4) | $M[Rsrc + (disp4)_{16}][7:0] \leftarrow Rdest$ |
| 分岐命令 | BEQ Rsrc1 Rsrc2 pcdisp4 | $if(Rsrc1==Rsrc2) PC \leftarrow PC + (pcdisp4)_{12}$ |
| | BRA pcdrc12 | $PC \leftarrow pcdrc12$ |
| | BL pcdrc12 | $rfp \leftarrow rfp+1; PC \leftarrow pcdrc12; BPC[rfp] \leftarrow PC+1; RF[rfp, n]$ |
| 割込命令 | TRAP pcdrc3 | $rfp \leftarrow rfp+1; BRA pcdrc3; BPC[rfp] \leftarrow PC[TRAP]+1; RF[rfp, n]$ |
| | RTE | $PC \leftarrow BPC[rfp]; rfp \leftarrow rfp-1; RF[rfp, n]$ |
| | NOP | No Operation |

4.1 新方式ハードウェア・アーキテクチャに特化したパイプライン

表4.2に新方式 CPU 回路を実行順に5段に分割したパイプラインを示す。表4.1の各命令は、表4.2のパイプラインに従って順に実行される。

表2.2に示すスタック方式 CPU のパイプラインでは、2.1.1項で述べたように割込・分岐命令の次に F ステージに投入された命令が、分岐先の命令ではなく、割込・分岐命令の次の命令になるという問題（制御ハザード）があり、F ステージの命令を NOP 化する必要があった。

しかし新方式 CPU において、表4.2に示すように通常 D ステージで行う無条件分岐命令の解読を F ステージで実施することにより制御ハザードを解消している。このことを図4.1と図4.2で説明する。

図4.1に分岐命令 BL の解読を D ステージで行う場合のパイプラインを示す。図において、100番地の BL 1000命令は表4.1に示すようにリンク付き分岐命令で、戻り先アドレスである101を退避させて1000番地に分岐する命令である。BL 命令は F ステージで取り出され、D ステージで解読されて次に実行すべき命令のアドレス：1000番地が F ステージに送られる。しかしパイプライン制御では、BL 命令が D ステージに入ったときに自動的に101番地の ADD 命令が F ステージに投入される。そのまま実行すると、本来戻ってきてから実行すべき101番地の ADD 命令が、BL 命令の次に実行すべき1000番地の SUB 命令の前に実行されてしまう。従って101番地の ADD 命令を F ステージで無効にする必要がある。この結果、図4.1から分かるように1000番地の SUB 命令発行は1ステージ遅れることになる。

新方式 CPU では上記問題を解決するため、図4.2に示すように分岐条件のない無分岐命

表4.2 新方式 CPU のパイプライン

| ステージ名 | 記号 | 内容 |
|----------|----|-------------------------------------|
| 命令フェッチ | F | 命令取出;無条件分岐先アドレス設定;rfpポインタ設定;BPCnへ保存 |
| 命令デコード | D | 命令解読;アドレス/データ準備;無条件分岐以外の次アドレス設定 |
| 命令実行 | E | 演算実行 |
| メモリ・アクセス | M | データメモリ読出書込 |
| ライトバック | W | レジスタへのデータ書込 |

| 番地 | 命令 | パイプライン | | | | | | |
|------|------------|--------|---------|---------|---------|---------|-----|--------|
| | | F | D | E | M | W | | |
| 100 | BL 1000 | F:BL取出 | D:BL解読 | E - | M - | W - | | |
| 101 | ADD R3, R4 | | F:ADD取出 | D - | E - | M - | W - | |
| 1000 | SUB R5, R2 | | | F:SUB取出 | D:SUB解読 | E:SUB実行 | M - | W:結果書込 |

図4.1 分岐命令解読を D ステージで行う場合のパイプライン

| 番地 | 命令 | パイプライン | | | | | | | | |
|------|------------|-----------|---------|---------|---------|---|---|--------|---|---|
| 100 | BL 1000 | F:BL取出/解読 | D | — | E | — | M | — | W | — |
| 1000 | SUB R5, R2 | | F:SUB取出 | D:SUB解読 | E:SUB実行 | M | — | W:結果書込 | | |

図4.2 分岐命令解読を F ステージで行う場合のパイプライン

令に対して解読を F ステージで行うようにした。図に示すように BL 命令は F ステージで取り出されて且つ解読もされるため、1000番地が F ステージ内で処理されて次に実行すべき命令のアドレスとなる。従って101番地が F ステージに投入されることはなく、BL 命令の次に1000番地の SUB 命令を発行することができるため、図4.1のように 1 ステージ分の遅れは生じない。このように分岐命令の F ステージでの命令解読は、割込・分岐処理の高速化をより効果的なものにする。

4.2 新方式ハードウェア・アーキテクチャ関連命令機能

4.2.1 TRAP 命令 (ソフトウェア割込)

表4.1の TRAP 命令機能で、rfp は register file pointer の略で 0～8 の値をとり、アドレスとして利用される。BPC[rfp]は、8本の BPC から構成されたファイルで BPC ファイルとよぶ。rfp は BPC ファイルのアドレスであるので、 $BPC[rfp] \leftarrow PC[TRAP]+1$ は、TRAP 命令の次命令のアドレスを BPC ファイルの rfp 番地に書込むことを示す。RF[rfp,n] ($0 \leq rfp \leq 8, 0 \leq n \leq 15$) は、2次元のレジスタ・ファイル・バンクで、rfp は 0～8 までのバンクアドレス、n は 0～15 までのレジスタ番号である。RF[rfp,n]は、rfp バンクの Rn レジスタを示す。

TRAP pcdrect3は、表4.2の F ステージで解読され、以下の順に処理が行われる。

- ① F ステージで rfp をインクリメントする。
- ② F ステージで TRAP 処理への分岐命令が格納されている pcdrect 3 番地へ BRA pcdrect3により無条件分岐させるため、pcdrect3番地を次アドレスとして設定する。
- ③ F ステージで BPC ファイルの rfp 番地に TRAP 命令の次の命令 (戻り先命令) の番地： $PC[TRAP]+1$ を保存する。
- ④ D ステージでレジスタ・ファイル・バンク：RF[バンク番号:rfp, レジスタ番号:n]のバンク番号が、上記①で更新された rfp に自動的に変更される。

以上の操作により、③の BPC の退避と④のレジスタの退避は TRAP 命令の中 (F ステージと D ステージ) で実行されるため、STbpc や PUSH 等の退避命令は不要になる。

4.2.2 BL 命令 (リンク付き分岐)

表4.1の BL pcdrect12命令は、サブルーチン分岐に使われる命令で以下の順に処理が行われる。

- ① TRAP 命令と同じ.
- ② F ステージで pcdrect12番地から始まるサブルーチンプログラムに制御を移す.
- ③ TRAP 命令と同じ.
- ④ TRAP 命令と同じ.

TRAP 命令との違いは②だけで、TRAP 命令では pcdrect3番地を介して間接的に割込処理に分岐するのに対し、BL 命令では分岐処理の先頭番地である pcdrect12番地に直接分岐する。

以上の操作により、③の BPC の退避と④のレジスタの退避は BL 命令の中 (F ステージと D ステージ) で実行されるため、STbpc や PUSH 等の退避命令は不要になる。

4.2.3 RTE 命令 (割込・分岐処理からの復帰)

表4.1の RTE 命令は、表4.2の F ステージで解釈され、以下の順に処理が行われる。

- ① F ステージで BPC ファイルの rfp 番地に保存されている戻り番地を PC に入力して、次の clk で元の実行プログラム中の TRAP 命令の次命令を発行させる。
- ② F ステージで rfp をデクリメントする。
- ③ D ステージで RF[バンク番号 :rfp, レジスタ番号 :n]のバンク番号が②で更新された rfp に自動的に変更される。

以上の操作により、①の BPC の復帰と③のレジスタの復帰は RTE 命令の中 (F ステージと D ステージ) で実行されるため、LDbpc や POP 等の復帰命令は不要になる。

4.3 新方式ハードウェア・アーキテクチャによる割込処理

新方式 CPU の TRAP 命令による割込処理の例を図4.3に示す。図4.3において、実行プログラムの100番地の TRAP 5命令が発行されると5番地に分岐し、5番地の BRA 500命令により500番地に分岐する。新方式では BPC やレジスタの退避は必要ないので、500番地から直ぐに割込処理を開始することができる。割込処理が終了すると BPC やレジスタ

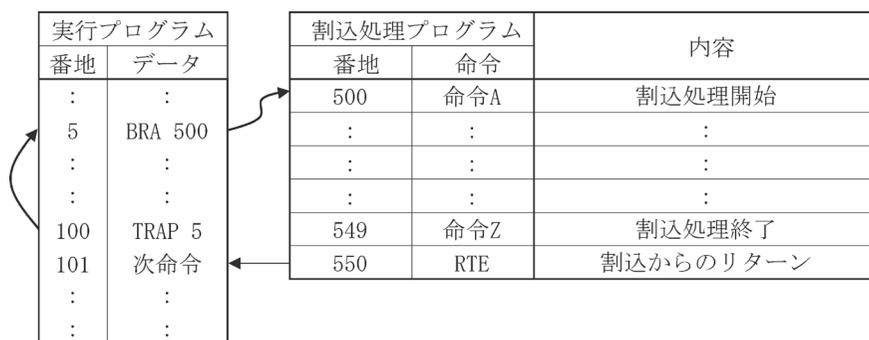


図4.3 新方式 CPU の TRAP 命令による割込処理

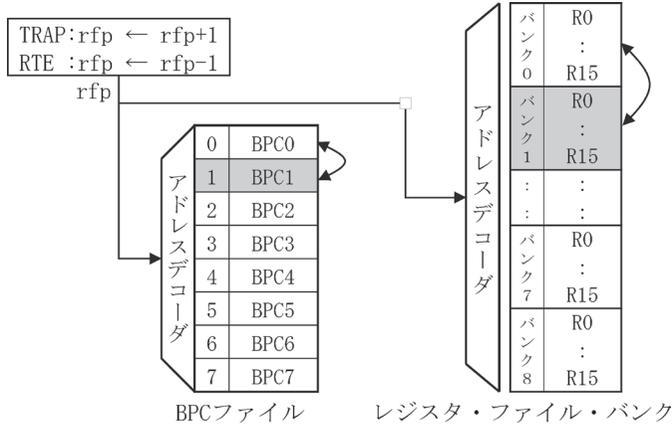


図4.4 新方式 CPU の割込処理における BPC およびレジスタの保護方式

への復帰は不要なので、割込処理が終了次第 RTE 命令で割込処理からのリターンが実行され元の実行プログラムの101番地の次命令に戻ることができる。

上記割込処理を可能とする BPC やレジスタの保護方式を図4.4に示す。

図4.4において、最初 rfp=0であったとする。その場合 BPC ファイルとレジスタ・ファイル・バンクのアドレスデコーダには0番地が入るので、BPC0とバンク0のレジスタ R0 ~ R15が使用される。

TRAP 命令による割込発生時、4.2.1①で述べたように rfp を 1 だけインクリメントするので、rfp=1となる。その結果、戻り先アドレスを BPC1に書込んで、割込処理プログラムはバンク1のR0~R15を使用することになる。この操作は TRAP 命令として実施されるので、スタック方式で必要であった退避命令 (STbpc や PUSH) は不要である。

割込処理が終了し、RTE 命令が実行されると4.2.3①で述べたように rfp=1のままなので、BPC1に保存した戻り番地を PC に送り元の実行プログラムに戻る。次に4.2.3②で述べたように rfp を 1 だけデクリメントするので、rfp=0となり、BPC0と実行プログラムで使用していたバンク0のレジスタ R0~R15が利用可能となる。この操作は RTE 命令として実施されるので、スタック方式で必要であった復帰命令 (LDbpc や POP) は不要である。

新方式 CPU に新規導入された BL 命令によるサブルーチン処理の例を図4.5に示す。BL 命令は TRAP 命令と異なり実行されると直接サブルーチンに分岐する。図4.5において、実行プログラムの100番地の BL 500命令が発行されると直接500番地に分岐し、割込処理が終了すると RTE 命令で割込処理からのリターンが実行され元の実行プログラムの101番地の次命令が実行される。BL 命令における BPC やレジスタの保護方式は図4.4と同じである。



図4.5 新方式 CPU の BL 命令によるサブルーチン処理

4.4 レジスタ間転送命令（レジスタコピー）

図4.4のレジスタ・ファイル・バンク図に示すように、新方式 CPU では、割込・分岐命令によりバンクが変更されレジスタ R0～R15のセットが強制的に変更されるので、実行プログラムで得たレジスタの内容を割込・分岐処理プログラムで利用するにはデータメモリに移しておく必要がある。（スタック方式の場合、R0～R15はスタックに退避されるが内容は変更されておらず、割込・分岐処理プログラムも同じレジスタを使うので、自動的にレジスタが引き継がれる。）

そこで新方式 CPU では、図4.4に示すバンク 8 をグローバル・レジスタとし、バンク 0 からバンク 7 との間でレジスタデータのコピーを可能にした。これにより実行プログラムから割込処理やサブルーチンへのデータ引継ぎはバンク 8 のレジスタに保存することで可能となる。

4.4.1 LDRG Rdest Gsrc

Rdest は実行中のプログラムに割り当てられているバンクに属する dest 番地のレジスタ、GRsrc は、グローバル・レジスタの src 番地のレジスタであり、その機能は、GRsrc の内容を Rdest にコピーすることである。

4.4.2 LDGR Gdest Rsrc

実行中のプログラムに割り当てられているバンクに属する Rsrc の内容をグローバル・レジスタの dest 番地のレジスタにコピーする。

4.4.3 LDRR Rdest Rsrc

実行中のプログラムに割り当てられている同一バンク内でのレジスタ間コピーである。

この命令は、グローバル・レジスタを用いたデータ転送とは関係ないが、LDRG や LDGR 命令をサポートする回路の制御線の値を 00 にするだけで実現できるため命令セットに加えた。

5. 新方式 CPU 回路の設計

新方式 CPU 回路の設計について述べる。CPU 回路は表4.1の新方式 CPU 命令セットの命令機能に基づいて設計された。

表5.1に新方式 CPU の命令フォーマットを示す。命令長は全て16ビットである。命令の種類を表す opcode はビット15～ビット12までの4ビットであるが、LDRR, LDRG, LDGRに限っては空いているビット1～ビット0の領域を第2の opcode として使用している。

レジスタは R0～R15の16本としたので、レジスタ番号である dest と src は4ビットである。未使用のビットは0とする。

disp は displacement の略で、アドレス間の距離を表す。imm は immediate の略で演算対象となる即値である。pcdisp は program counter displacement の略で、program counter 値（発行されている命令のアドレス）と分岐先アドレスとのアドレス間距離である。pcdrct は pc direct の略で分岐先の直接アドレスである。また disp4, imm8や pcdrc12の数字はビット数を示している。

設計手順を以下に示す。

手順1：各命令を実現する回路をRTL（Register Transfer Level）で設計。

手順2：回路図が似ている回路同士を統合して回路グループを設計し、最後に全ての回路グループを統合して CPU 全体回路を設計。

表5.1 新方式 CPU の命令フォーマット

| bit番号→ | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|--------|---------|----|----|----|----------|----|---|---|------|---|---|---|---------|---|---------|---|---|
| | opcode | | | | dest | | | | src | | | | 未使用 | | | | |
| ADD | 0 | 0 | 0 | 1 | | | | | | | | | | 0 | 0 | 0 | 0 |
| SUB | 0 | 0 | 1 | 0 | | | | | | | | | | 0 | 0 | 0 | 0 |
| AND | 0 | 0 | 1 | 1 | | | | | | | | | | 0 | 0 | 0 | 0 |
| OR | 0 | 1 | 0 | 0 | | | | | | | | | | 0 | 0 | 0 | 0 |
| | opcode | | | | dest | | | | src | | | | disp4 | | | | |
| LD | 1 | 0 | 0 | 0 | | | | | | | | | | | | | |
| ST | 1 | 0 | 0 | 1 | | | | | | | | | | | | | |
| BEQ | 1 | 0 | 1 | 0 | | | | | | | | | | | | | |
| | opcode1 | | | | dest | | | | src | | | | 未使用 | | opcode2 | | |
| LDRR | 1 | 1 | 0 | 1 | | | | | | | | | | 0 | 0 | 0 | 0 |
| LDRG | 1 | 1 | 0 | 1 | | | | | | | | | | 0 | 0 | 0 | 1 |
| LDGR | 1 | 1 | 0 | 1 | | | | | | | | | | 0 | 0 | 1 | 0 |
| | opcode | | | | 未使用 | | | | 未使用 | | | | pcdrct3 | | | | |
| TRAP | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | opcode | | | | dest | | | | imm8 | | | | | | | | |
| LDI | 0 | 1 | 1 | 1 | | | | | | | | | | | | | |
| | opcode | | | | pcdrct12 | | | | | | | | | | | | |
| BRA | 1 | 0 | 1 | 1 | | | | | | | | | | | | | |
| BL | 1 | 1 | 0 | 0 | | | | | | | | | | | | | |
| | opcode | | | | 未使用 | | | | | | | | | | | | |
| RTE | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| NOP | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

手順3：CPU 全体回路を F, D, E, M, W の5段のパイプライン・ステージに分割。

手順4：各 RTL 回路を Verilog HDL で記述

以下では手順1～3で得られた回路について F ステージから順に説明する。

5.1 F ステージ回路

図5.1に新方式 CPU の F ステージ回路を示す。このステージにおける新技術は、通常の回路に bpc_file 回路, file_pointer 回路と fetch_解読器を追加し、BPC の退避を TRAP 命令と BL 命令で、BPC の復帰を RTE 命令で実現したことである。

図5.1において、灰色に塗りつぶした回路は、クロックに同期する回路である。楕円の塗りつぶし回路はパイプラインのための D ステージ入口の同期レジスタで楕円内の数字はビット数を示す。また太枠回路は、新型 CPU 回路にのみ存在する回路である。

program counter は、命令メモリのアドレスを保持するレジスタで、その出力 pcout は命令アドレスである。

命令メモリに命令アドレス pcout が与えられると、そのアドレスに存在する命令が inst

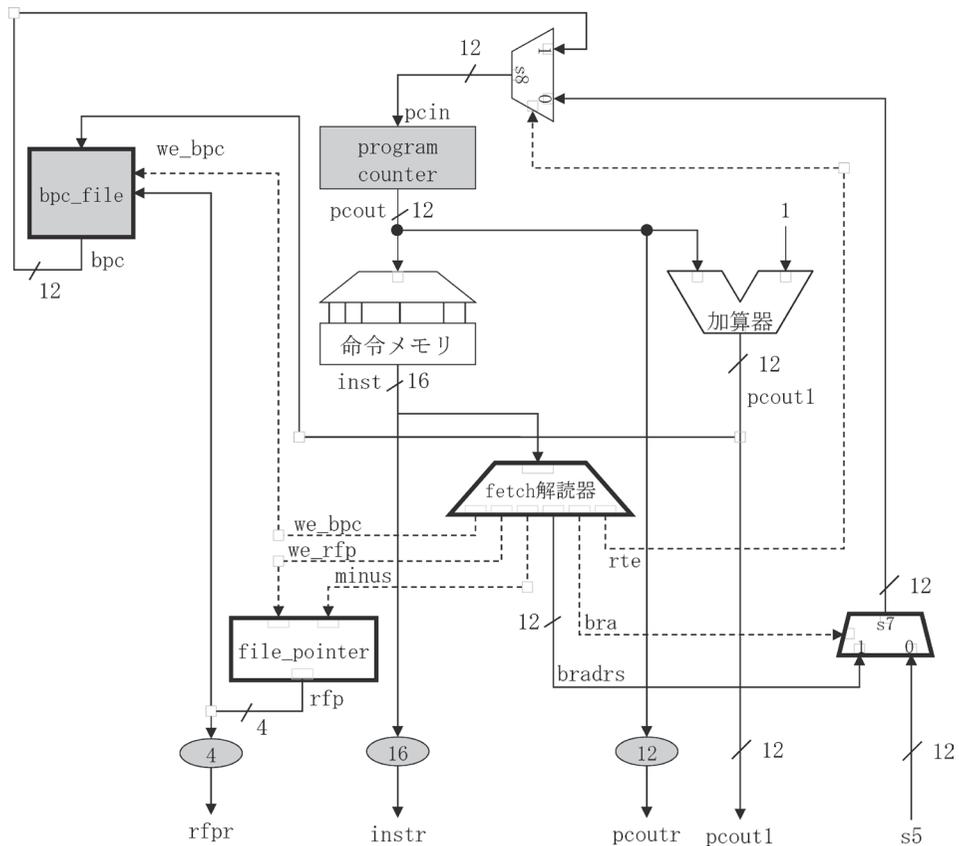


図5.1 新方式 CPU の F ステージ回路

に出力される。

加算器は、発行された命令のアドレス `pcout` に 1 だけ加算して、次に実行する命令のアドレス `pcout1` を計算する。

`fetch` 解読回路は、割込命令である `TRAP`、無条件分岐命令である `BL`、`BRA`、復帰命令である `RTE` を解読して制御線や分岐先アドレスを出力する回路である。

`file_pointer` は、`we_rfp=1` のときに動作する 4 ビットの `rfp` ポインタを出力する回路で、表 4.1 の `TRAP` 命令（割込）や `BL` 命令（リンク付き分岐）が発行されたときに 1 だけインクリメントされ、`RTE` 命令が実行されたときに 1 だけデクリメントされる。

`bpc_file` は、8 個の 12 ビット BPC を退避させることができる同期メモリの一種で、`TRAP`、`BL`、`RTE` 命令のときに `we_bpc=1` となり動作する。各 BPC のアドレス 0000~0111 は `rfp` で与えられ、`pcout1` がデータとして書き込まれる。理由は、`pcout1` は発行中の命令の次の命令アドレス（即ち戻り先アドレス）だからである。`s7` や `s8` は 12 ビットのセレクタである。

F ステージ回路の動作について説明する。

表 5.1 の `TRAP pcdrect3` 命令が `inst` に出力されると、`fetch` 解読器でデコードされ、`we_bpc=we_rfp=1`、`minus=0`、`bra=1`、`rte=0`、`bradrs` が $(pcdrect3)_{12}$ となるので、`s7` で $(pcdrect3)_{12}$ が選択され `s8` の 0 入力に入り、`s8` で 0 入力を選択され $(pcdrect3)_{12}$ が `pcin` となり `TRAP` 命令の次の `clk` で program counter から出力されて命令メモリに与えられ、 $(pcdrect3)_{12}$ 番地に分岐する。

このとき `file_pointer` に $(we_rfp, minus)=(1,0)$ が入力されるので、`rfp` が 1 だけインクリメントされて `bpc_file` に送られ、`we_bpc=1` より `pcout1` がデータとして `rfp` 番地に書き込まれる。

以上の動作より、`TRAP pcdrect3` 命令を実行すると、 $(pcdrect3)_{12}$ 番地に分岐するだけでなく次の `clk` で `bpc_file` において 1 だけインクリメントされたアドレスの BPC に `pcout1` が退避（保存）され、元の `rfp` 番地の BPC は消されずに保存される。即ち、スタック方式 CPU での `STbpc` 命令による BPC の退避が、新方式 CPU では `TRAP pcdrect3` 命令の D ステージで実行されたことになる。

表 5.1 の `BL pcdrect12` 命令の動作は上記 `TRAP pcdrect3` 命令とほぼ同じである。両命令の違いは、`BL` 命令のオペランドである `pcdrect12` が元々 12 ビットなので `TRAP pcdrect3` のときのように `fetch` 解読器内で `pcdrect3` を 12 ビットの $(pcdrect3)_{12}$ に拡張する必要がないことである。

表 5.1 の `RTE` 命令が `inst` に出力されると、`fetch` 解読器でデコードされ、`we_bpc=0`、`we_rfp=1`、`minus=1`、`bra=1`、`rte=1`、`bradrs=0` となるので、`s8` で 1 入力を選択され、`bpc_file` に最後に保存された BPC が `bpc` から `s8` の 1 入力を通して `pcin` となり `RTE` 命令の次の `clk` で program counter から出力されて命令メモリに与えられ分岐前の戻り先アドレスに分岐する。

このとき `file_pointer` に $(we_rfp, minus)=(1,1)$ が入力されるので、`file_pointer` 内で `rfp` が 1 だけデクリメントされるが内部で 1 clk 遅延用のレジスタを通すので直ぐに `rfp` に出力されることはない。次の clk でデクリメントされた `rfp` が `bpc_file` に送られ、RTE 命令実行時に `bpc` が保持していた戻り先アドレスの前に保存された戻り先アドレスが `bpc` に出力される。

以上の動作より、RTE 命令を実行すると、戻り先アドレスに分岐するだけでなく、次の clk で `file_pointer` から 1 だけデクリメントされた `rfp` が `bpc_file` に送られ、最後に保存された BPC の一つ前の BPC が非同期で `bpc` 出力に保持される。即ち、スタック方式 CPU での `LDbpc` 命令による BPC の復帰が、新方式 CPU では RTE 命令の D ステージで実行されたことになる。

表5.1の `BRA pcdrect12` が `inst` に出力されると、`fetch` 解読器でデコードされ、`we_bpc=0`, `we_rfp=0`, `minus=0`, `bra=1`, `rte=0`, `bradrs=pcdrect12` となるので、`s7` で `pcdrect12` が選択され `s8` の 0 入力に入り、`s8` で 0 入力を選択され `pcdrect12` が `pcin` となり `BRA pcdrect12` 命令の次の clk で `program counter` から出力されて命令メモリに与えられ、`pcdrect12` 番地に分岐する。

このとき `file_pointer` では `we_rfp=0` より `rfp` は変化せず、`bpc_file` でも `rfp` が不変で `we_bpc=0` ならば `bpc` は変化しない。

表5.1の上記以外の全ての命令が `inst` に出力されると、`fetch` 解読器の出力が全て 0 になるので、`file_pointer` や `bpc_file` は変化せず、`s7` と `s8` で 0 入力を選択されるので D ステージからの `s5` が `pcin` に転送される。即ち次の命令アドレスは F ステージではなく D ステージで決められる。

5.1.1 BPC ファイル (`bpc_file`)

図5.2に BPC ファイルの構成図を示す。図において `rst` はリセット、`clk` はクロック、`we_bpc` はデータ書込許可、`rfp` は 4bit のレジスタ・ファイル・ポインタで BPC ファイルのアドレス、`pcout1` は 12bit の BPC 入力、`bpc` は 12bit の BPC 出力である。内部には BPC0 ~ BPC7 の 8 個のレジスタが格納されている。各 BPC の bit 幅が 12bit である理由は、命令メモリのアドレス幅を 12bit としたからである。BPC が 8 個にも関わらずアドレスである `rfp` の bit 数が 4bit である理由は、`rfp` がレジスタ・ファイルの退避にも使用されるため、BPC ファイルで有効なポインタビットは、`rfp[2:0]` の 3bit である。

単体の動作について説明する。`rst` がかかると、BPC0 ~ BPC7 までの全ての BPC レジスタが 0 にリセットされる。書込は `clk` 同期である。即ち `clk = ↑` 前に `we_bpc`, `rfp`, `pcout1` が与えられ、`clk = ↑` になった瞬間に `pcout1` にある BPC 値が `rfp` アドレスに書き込まれる。読出は `clk` 非同期であるので、`rfp` にアドレスを与えると `clk` に無関係に BPC 値が `bpc` に出力される。

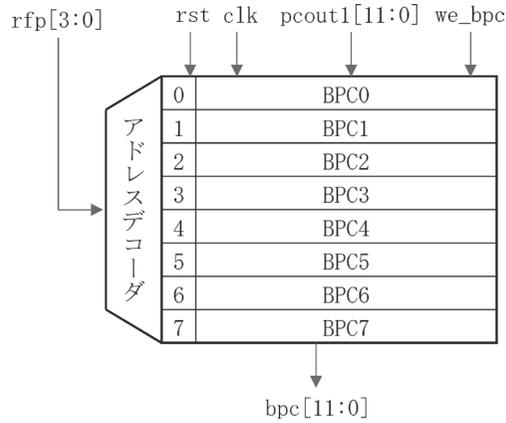


図5.2 BPC ファイルの構成図

次に CPU 中での BPC ファイルの動作について図5.3で説明する。図において BPC ファイルの信号は、rst ~ bpc[11:0]である。pcout は命令メモリから取り出した命令のアドレス、instはその命令である。we_rfp と minus は file_pointer の出力である。また図中の数字は全て16進数である。

<リセット時>

最初に rst=0でリセットされると、 rfp および bpc は 0 にリセットされ、 pcout と inst より 9 番地の 710B (表5.1 : LDI R1 B) という命令が発行される。 9 番地からスタートする理由は、 リセットをかけたときに 9 番地からスタートするように設定したからである。このとき pcout1 に与えられる BPC 値は 9 番地 + 1 より 00A となっている。この理由は、 A 番地の命令は発行された 9 番地の命令の次に実行される命令だからである。

< TRAP 命令実行 >

inst の 3 目 : 00B 番地の E007 (表5.1 : TRAP 7) 命令は、 ソフトウェア割込である。

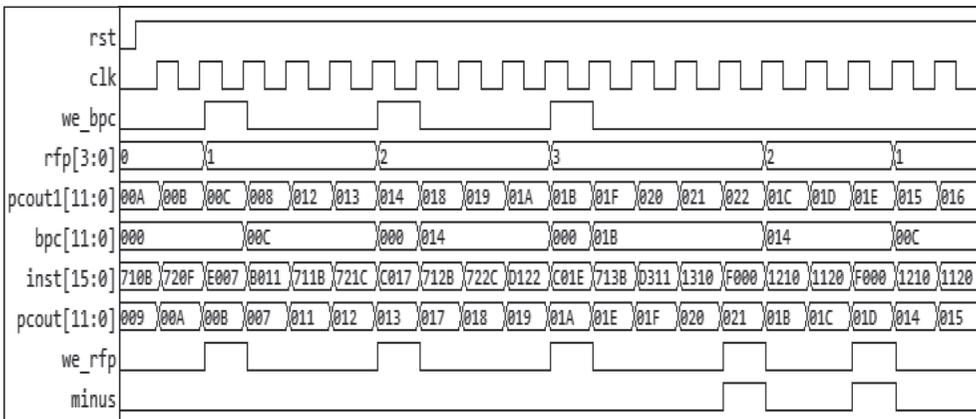


図5.3 BPC ファイルの動作

従って戻り先アドレスである (00B の次のアドレス) 00C を BPC ファイルに退避させなくてはならない。E007が発行されたとき、we_bpc=1, pcout1=00C, (file_pointer 入力 we_rfp=1, minus=0より rfp が 0 → 1) rfp=1であるので、clk= ↑となった瞬間に (rfp=) 1番地に00Cが書き込まれ、書き込まれた値00Cはbpcに出力されるはずである。

図に示すように、E007が発行された次の clk= ↑で、bpc=00Cとなっている。これは BPC ファイルの (rfp=) 1番地に戻り先アドレス00Cが退避されたことを示している。

このように TRAP 命令は、割込処理に分岐するだけでなく BPC の退避も実現している。
 < BL 命令実行 >

TRAP 命令による割込処理 (011番地～)に分岐して3clk後に発行される7番目の命令：(pcout=) 013番地の (inst=) C017 (表5.1：BL 17) という命令は、リンク付き分岐命令であるので戻り先アドレスである (013の次のアドレス) 014を BPC に退避させなければならない。C017が発行されたとき we_bpc=1, pcout1=014, (file_pointer 入力 we_rfp=1, minus=0より rfp が 1 → 2) rfp=2であるので、clk= ↑となった瞬間に (rfp=) 2番地に014が書き込まれ、書き込まれた値はbpcに出力されるはずである。

図に示すように、C017が発行された次の clk= ↑で、bpc=014となっている。これは BPC ファイルの (rfp=) 2番地に戻り先アドレス014が退避されたことを示している。

このように BL 命令は、割込処理に分岐するだけでなく BPC の退避も実現している。
 < 2度目の BL 命令実行 >

pcout=01A 番地の C01E (表5.1：BL 1E) の発行により同様に pcout1=01B が BPC ファイルの rfp=3番地に書き込まれ、C01Eの次の clk= ↑で bpc に書き込まれている。

< RTE 命令実行 >

(pcout=) 01E 番地に分岐後、(pcout=) 021番地の (inst=) F000 (表5.1：RTE) 命令が発行される。

この命令は分岐先からの復帰命令であるので、戻り先アドレスを BPC から読み出して、そのアドレスに分岐しなければならない。

図より最初の F000が発行されたとき、BPC ファイルの状態は、we_bpc=0 (読出)、bpc=01B (BPC データ：戻り先アドレス)、rfp=3 (BPC のアドレス) であるので、clk= ↑となった瞬間に pcout=01B となり01B番地の1210 (表5.1：ADD R2 R1) が発行されている。また (file_pointer 入力 we_rfp=1, minus=1より rfp が 3 → 2) rfp=2に変化するので (rfp=) 2番地に格納されている BPC 値014が bpc に出力されている。

このように RTE 命令は、BPC に分岐するだけでなく、BPC の復帰も実現している。
 < 2度目の RTE 命令実行 >

(pcout=) 01B 番地に分岐後、(pcout=) 01D 番地の (inst=) F000 (表5.1：RTE) 命令が発行される。

図より2度目の F000が発行されたとき、BPC ファイルの状態は、we_bpc=0 (読出)、

bpc=014 (BPC データ : 戻り先アドレス), rfp=2 (BPC のアドレス) であるので, clk= ↑ となった瞬間に pcout=014 となり 014 番地の 1210 (表5.1 : ADD R2 R1) が発行されている。また (file_pointer 入力 we_rfp=1, minus=1 より rfp が 2 → 1) rfp=1 に変化するので (rfp=) 1 番地に格納されている BPC 値 00C が bpc に出力されている。

5.1.2 ファイル・ポインタ (file_pointer)

図5.4にファイル・ポインタの回路図を示す。図において, count は, rst=0のときに 0 を出力し, we_rfp= ↑ のときに s1出力の flpt を読み込んで count に出力する。右の加算器は count を 1 だけインクリメントし, 左の加算器は 1 だけデクリメントする。register_4は 4 ビットレジスタで rst=0で 0 を出力し, clk= ↑ のとき flpt を読み込んで flptr を出力する。従って flptr は flpt より 1clk 過去の値になっている。セクタ s1と s2は共に minus により制御され, s1はリセット付きセクタで, minus=0のときインクリメントした値を, minus=1 のときデクリメントした値を選択する。s2は minus=0のときインクリメントした直ぐの値を, minus=1のときデクリメントした値の 1clk 過去の値を出力する。

次に CPU 中での file_pointer の動作について図5.5で説明する。図において file_pointer の信号は, rst ~ rfp[3:0]である。minus は rfp 出力をインクリメントするかデクリメントするかを指示する入力, we_rfp は file_pointer を動作させる信号, flpt はインクリメント時に rfp に出力する信号, flptr はデクリメント時に rfp に出力する信号である。また図中の数字は全て16進数である。

file_pointer が動作するのは, we_rfp=1のときである。図で we_rfp=1のときの命令 inst を確認すると, 順に E007 (表5.1 : TRAP 7), C017 (表5.1 : BL 17), C01E (表5.1 : BL 1E), F000 (表5.1 : RTE) である。即ち分岐・復帰命令のときに動作する。

pcout=00B の inst=E007のとき, we_rfp=1, minus=0なので, flpt=0→1になり rfp に出力される。BPC ファイルでは, これを受けて rfp=1番地に pcout1=00C が保存される。次の pcout=013, inst=C017のときも we_rfp=1, minus=0なので, flpt=1→2になり rfp に出力される。BPC ファイルでは, これを受けて rfp=2番地に pcout1=014が保存される。次の pcout=01A, inst=C01E のときも同じく rfp=2→3になる。BPC ファイルでは, これを受けて rfp=3番地に pcout1=01B が保存される。

pcout=021の inst=F000のとき, we_rfp=1, minus=1なので, flpt=3→2になるが, 図5.4で述べたように minus=1のとき flptr=3が rfp に出力され BPC ファイルに送られる。BPC ファイルでは, これを受けて rfp=3番地に bpc=01B を出力する。即ち rfp は変化せずに送られ, 次の clk= ↑ で flptr=3→2, rfp=3→2となりデクリメントされる。

次の pcout=01D の inst=F000のときも F000が発行されたときは rfp=2のまま, 次の clk= ↑ で rfp=2→1となりデクリメントされる。

以上の file_pointer 動作により BPC ファイルでは, 書き込むときは rfp をインクリメン

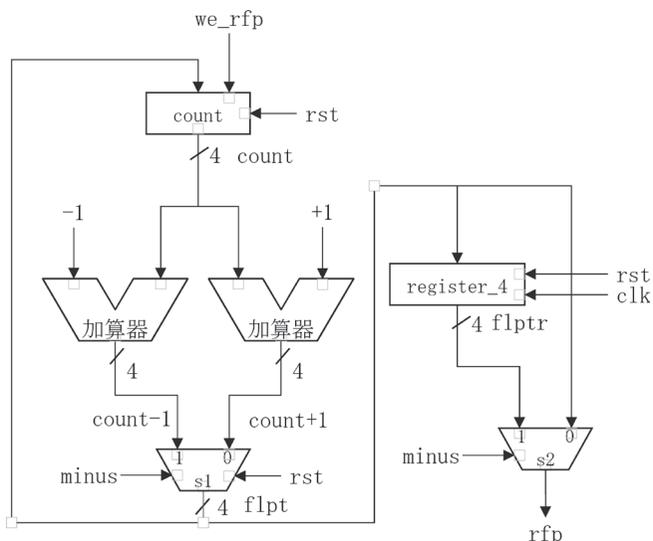


図5.4 ファイル・ポインタ回路図

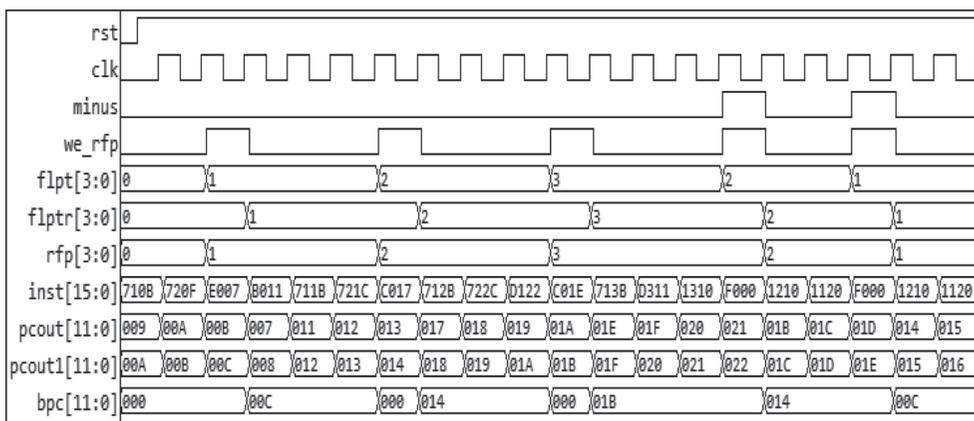


図5.5 file_pointer の動作

トしたアドレスに pcout1 を保存していき、読み出すときは rfp をインクリメントしたアドレスの pcout1 を bpc に出力した後に rfp をデクリメントしたアドレスの pcout1 を bpc に出力するという動作を保証している。

5.2 D ステージ回路

5.2.1 D ステージ回路の概要

図5.6に新型 CPU の D ステージ回路を示す。このステージにおける新技術は、レジスタの退避・復帰命令を削除するため、16個のレジスタから成るレジスタ・ファイルを1バンクとして9バンクから成るレジスタ・ファイル・バンクを開発したことである。

図5.6において、灰色に塗りつぶした回路は、クロックに同期する回路である。楕円の

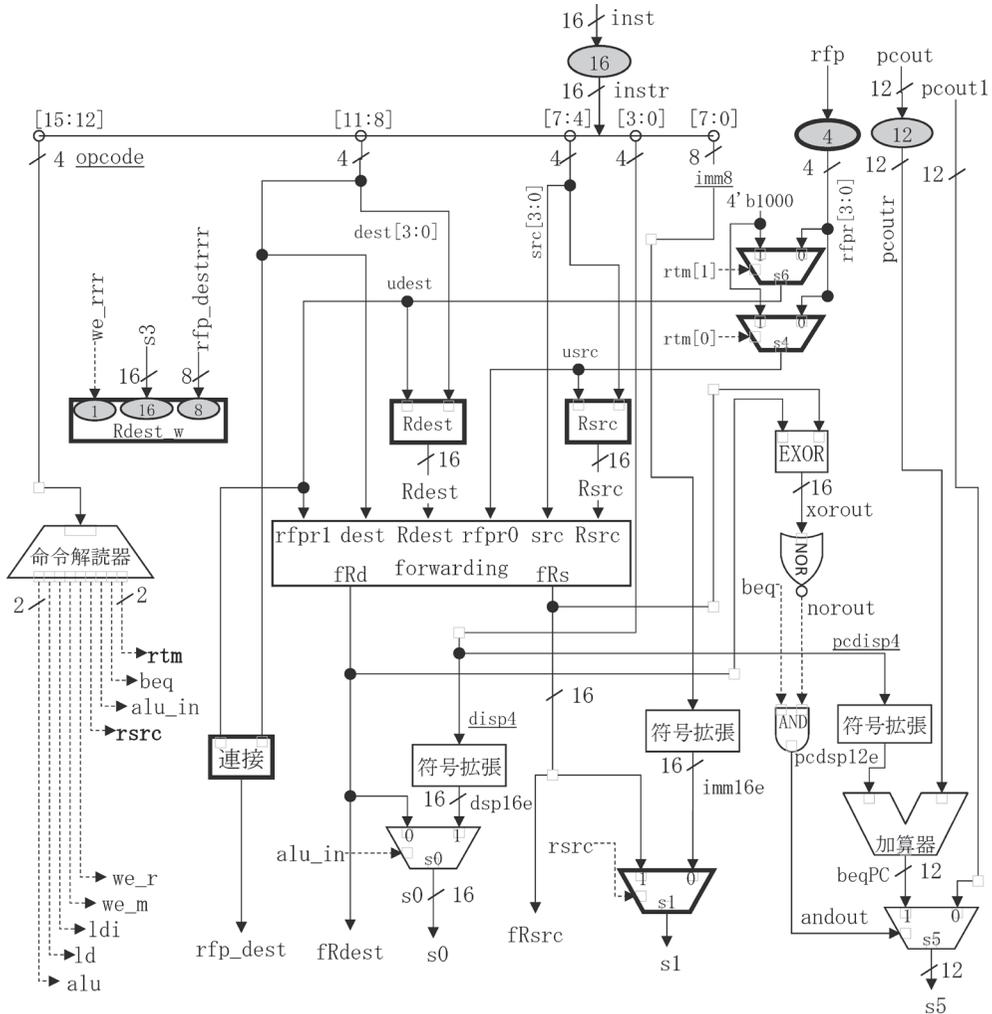


図5.6 新方式 CPU の D ステージ回路図

塗りつぶし回路はパイプラインのための D ステージ同期レジスタで楕円内の数字はビット数を示す。また太枠回路は、新型 CPU 回路にのみ存在する回路である。

命令解読器は F ステージで読解しなかった命令を解読する回路で 9 本の制御線を出力する。

レジスタ・ファイル・バンクは Rdest_w, Rdest, Rsrc の 3 つに分けて描かれているが、機能別に分けて表現しているだけで回路としては 1 つである。Rdest_w はレジスタへの書込回路, Rdest は表 5.1 の命令フォーマットにおける dest 番地からの読出回路, Rsrc は同じく src 番地からの読出回路である。dest や src は 4 ビットであるが、レジスタ・ファイル・バンクのアドレス入力は 8 ビットである。8 ビットのアドレスは、Rdest の場合は 4 ビットの udest と 4 ビットの dest の接続から、Rsrc の場合は 4 ビットの usrc と 4 ビットの src

の接続からそれぞれ生成される。s6とs4の0入力にはrfpr[3:0]が、s6とs4の1入力には10進数の8が入力され、2ビットのrtm[1:0]信号のうち、s6はrtm[1]により、s4はrtm[0]により選択される。

forwarding 回路は、データ・ハザードを解消する回路で、fRd から fRdest が、fRs より fRsrc が出力される。この回路は本研究内容ではないので説明は省略する。

EXOR → NOR は、EXOR に入力される fRdest と fRsrc が等しいかどうかを判定する回路で、等しい場合に 1 となる。

加算器は、表4.1に示す BEQ 命令での $PC+(pcdisp4)_{12}$ の加算に使用される。

5.2.2 レジスタ・ファイル・バンク

図5.7にレジスタ・ファイル・バンクの構成図を示す。rad1[7:0]は読出アドレス1で、そのアドレスに存在するレジスタ値がout1[15:0]から出力される。rad2とout2の関係も同じである。wad[7:0]は書き込みアドレスで、書き込み許可 we=1のときにin[15:0]に与えられているレジスタ値をwadアドレスに書き込む。

レジスタは144個あるが、R0～R127の128個について、16個ずつ8個の組に分けて、R0～R15をバンク0、R16～R31をバンク1、・・・R112～R127をバンク7とした。バンクアドレス（0～7）はrad1、rad2、wadのアドレスの[6:4]の3bit（000～111）で指定する。この上位3bitは、図5.6のDステージ回路図にあるrfpr[3:0]をs4およびs6セレクタで

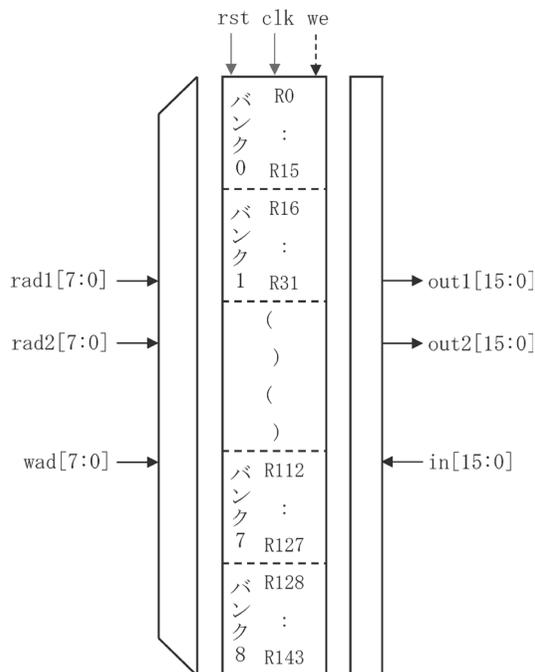


図5.7 レジスタ・ファイル・バンクの構成図

選択した4ビットである。各バンク内での16個のレジスタは図5.6の dest[3:0]と src[3:0]により指定される。まとめるとレジスタ・ファイルの3つのアドレス：rad1, rad2, wadには rfp と dest, rfp と src, rfp と dest' を接続したものが入力され、rad1={rfp,dest}, rad2={rfp,src}, wad={rfp,dest'} となる。上記5.1項で述べたように rfp は割込・分岐命令が入ったときにFステージで指定されるポインタで、図5.6の s4と s6で異なる rfp を選択することはできないので、同一バンク内の16本のレジスタを使ってプログラムが実行される。

図5.8にCPUのDステージにおけるレジスタ・ファイル・バンクの割込・分岐・グローバル・レジスタ動作を示す。レジスタ・ファイル・バンクの信号は clk ~ out2までである。図において数字は全て16進数である。

pcoutr=009番地の instr=710B (表5.1: LDI R1 B) 命令は、R1←000B を実行する。R1の1は、4ビットの udest と dest の接続であるが、図のように udest=0, dest=1より {udest, dest}=01となっている。次の pcoutr=00A番地の instr=720F 命令は、R2←000F を実行する。R2の2は同じく図で udest=0, dest=2より {udest, dest}=02となっている。udest=0はバンク0のレジスタ・ファイルである。usrc も0である。

次の pcoutr=00B番地の instr=E007 (表5.1: TRAP 7) 命令は割込命令なので、図のように udest=1に増加する。これは次の命令からバンク1のレジスタ・ファイルを使用することを示す。以降の711B (R1←1B), 721C (R2←1C) 命令での {udest, dest} は、図より =11, 12となっている。usrc も1に増加している。

次の pcoutr=013番地の instr=C017 (表5.1: BL 17) 命令は分岐命令なので、図のように

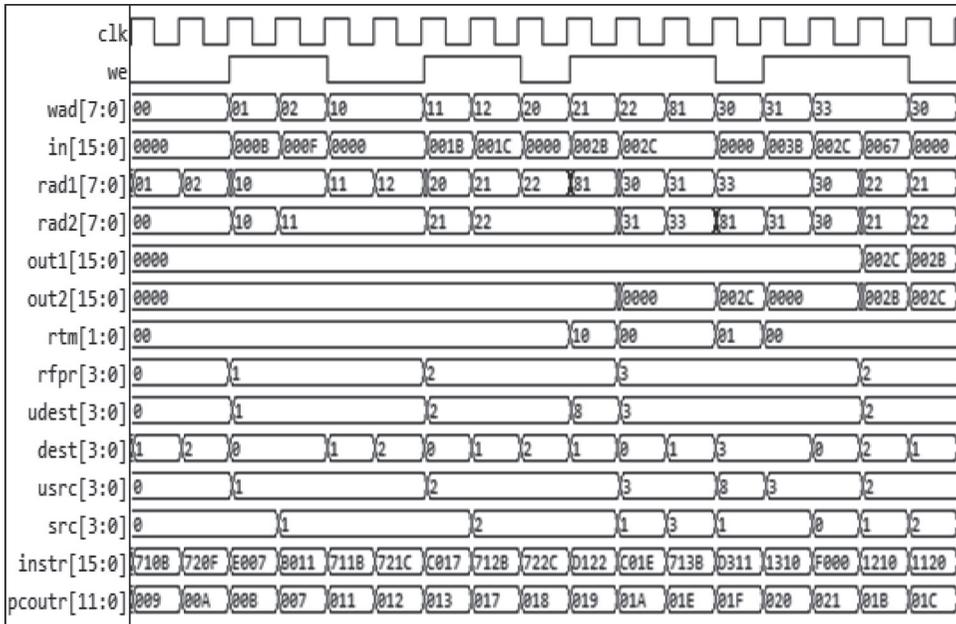


図5.8 レジスタ・ファイル・バンクの動作

udest=2に増加する。これは次の命令からバンク 2 のレジスタ・ファイルを使用することを示す。以降の712B (R1←2B), 722C (R2←2C) 命令での {udest, dest} は、図より =21, 22 となっている。usrc も 2 に増加している。

次の pcoutr=019 番地の instr=D112 (表5.1: LDGR GR1←R2) 命令は R2 を GR1 にコピーする命令である。Rdest や Rsrc はバンク 2 にあるので、R2 の usrc=2, src=2 である。GR1 はバンク 8 のグローバル・レジスタなので、図のように udest=8, dest=1 になっている。即ち、レジスタ・ファイル・バンクで、22 番地のデータを 81 番地にコピーするのである。

次の pcoutr=01A 番地の instr=C01E (表5.1: BL 1E) 命令は分岐命令なので、図のように udest=3 に増加する。これは次の命令からバンク 3 のレジスタ・ファイルを使用することを示す。以降の713B (R1←3B) 命令での {udest, dest} は、図より =31 となっている。usrc も 3 に増加している。

次の pcoutr=01F 番地の instr=D311 (表5.1: LDGR R3←GR1) 命令は GR1 を R3 にコピーする命令である。Rdest や Rsrc はバンク 3 にあるので、R3 の udest=3, dest=3 である。GR1 はバンク 8 のグローバル・レジスタなので、図のように usrc=8, src=1 になっている。即ち、レジスタ・ファイル・バンクで、81 番地のデータを 33 番地にコピーするのである。以降の1310 (ADD R3 R1) 命令での {udest, dest}, {usrc, src} は、図より 33, 31 となっている。

以上の動作で rfpr と udest, usrc の関係を見ると、図5.8よりグローバル・レジスタが使用される D122 や D311 命令以外で rfpr=udest=usrc であることが分かる。rfpr は F ステージの rfp なので、rfp (rfpr) は BPC ファイルとレジスタ・ファイル・バンクの両方に与えられ、割込や分岐が起きたときに BPC ファイルの BPC 保存番地とレジスタ・ファイル・バンクのバンク番号を増加させることで BPC やレジスタの「退避」を不要にしているのである。

他方復帰動作は、図の pcoutr=021 番地の instr=F000 (表5.1: RTE) 命令で実現される。RTE 命令が発行されると前5.1.2項で述べたように、rfp (rfpr) は、発行と同時にデクリメントされるのではなく、次の clk=↑でデクリメントされる。図では pcoutr=01B 番地に復帰すると、rfpr=udest=usrc=2 となり、バンク 3 からバンク 2 に移る。01B 番地の instr=1210 (ADD R2 R1) の R2 と R1 の 2 や 1 は、{udest, dest} や {usrc, src} なので、図より 22, 21 となりバンク 2 に移って計算が行われる。

以上の動作から、RTE による復帰では BPC ファイルの BPC 保存番地とレジスタ・ファイル・バンクのバンク番号を減少させることで BPC やレジスタの「復帰」を不要にしているのである。

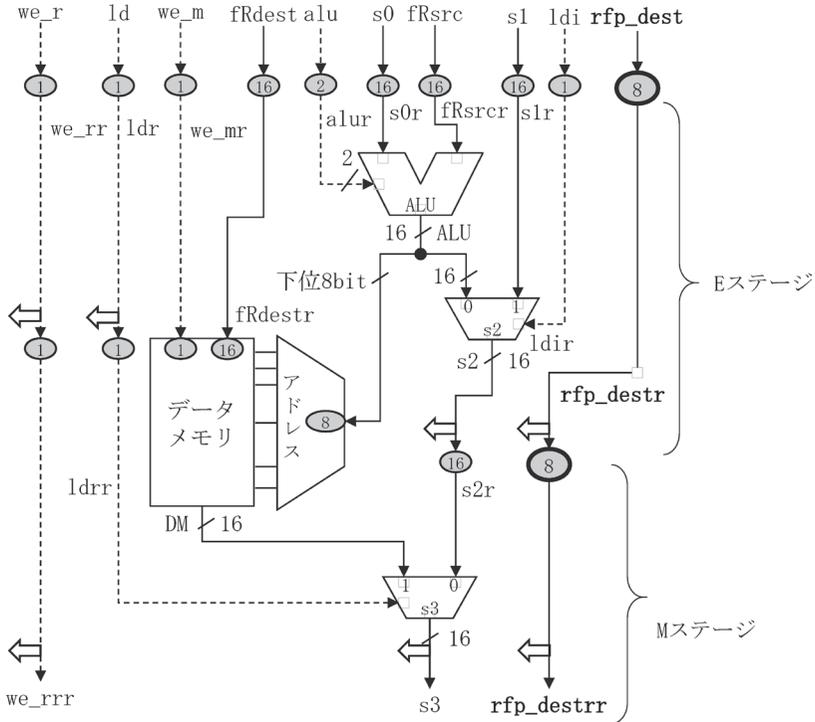


図5.9 新方式 CPU の E ステージと M ステージの回路図

5.3 E ステージと M ステージ回路

図5.9に新型 CPU の E ステージと M ステージの回路図を示す。図において上側が E ステージ、下側が M ステージである。新方式 CPU に関してこれらのステージに新奇性はないが、CPU として評価するため概要を説明する。

5.3.1 E ステージ回路

スタック方式 CPU の E ステージでは PUSH や POP 命令に必要となる SP の計算回路が必要であったが、新方式では不要である。

図5.9において、E ステージの全ての入力にはパイプライン・レジスタが配置されている。表4.1に示したように演算命令は4つだけなので、E ステージの回路は ALU とセレクタ s2 だけである。

ALU では表4.1の4つの演算命令を実行する。演算を指示するのは2ビットの alu 信号で、alu=00のとき加算、01のとき減算、10のとき論理積、11のとき論理和を指示する。また、LD 命令や ST 命令におけるデータメモリのアドレス計算（加算、減算）も行う。データメモリのアドレスは8ビットなので、ALU の16ビット出力の下位8ビットだけを M ステージに送る。上位8ビットは放置される。

E ステージの出口にある左横向き矢印は、D ステージのフォワーディング回路へ向かう

信号である。

5.3.2 M ステージ回路

図5.9において、M ステージの全ての入力にもパイプライン・レジスタが配置されている。このステージではデータメモリへの書き込みと読み出しが行われるので、回路はデータメモリとセレクタ s3だけである。

データメモリに書き込む場合、we_mr=1とすると fRdstr が ALU[7:0] 番地に書き込まれる。読み出す場合、we_mr=0とすれば ALU[7:0] 番地にあるデータが DM から出力される。即ち、書き込みか読み出しかは we_mr が 1 か 0 かで決まる。

ST 命令の場合、書き込みが終わると命令が終了する。

LD 命令の場合、読み出したデータ DM を s3 で ldr=1 として選択し、s3 から D ステージに送ってレジスタ・ファイルに書き込むまで終了しない。レジスタ・ファイルへの書き込みアドレスは、D ステージで命令が解読されてから一緒にパイプライン・レジスタを通過してきた rfp_dest である。これは5.2.2項で述べた {u dest, dest} そのもので、D ステージから rfp_dest として出力されていた。即ち、命令では dest は 4 ビットであったが、レジスタ・ファイル・バンク以降は 8 ビットとして扱っている。書込許可は we_rrr であり、D ステージの命令解読器出力の we_r が一緒にパイプライン・レジスタを通過してきている。

M ステージの出口にある左横向き矢印は、D ステージのフォワーディング回路へ向かう信号であり、且つレジスタ・ファイル・バンクへも転送されて書き込まれる。

6. 新方式 CPU の設計検証

6.1 HDL 記述と論理シミュレーション

新方式 CPU の各 RTL 回路は Verilog HDL により記述されている。表6.1に新方式 CPU の Verilog HDL ファイルの階層構造、モジュール名および HDL 行数を示す。なお行数にコメント文は含まれていない。

表6.1に示すように、0 階層は CPU 全体回路、1 階層は CPU 全体回路にインクルードされる各パイプライン・ステージそして 2 階層は各ステージにインクルードされるモジュールである。2 階層の一部のモジュールは 1 階層に重複してインクルードされるので、重複を削除した 2 階層のモジュールを右側に再掲した。新方式 CPU の HDL 記述行数の総計は 566 行である。

比較のために表6.2にスタック方式 CPU の Verilog HDL ファイルの階層構造、モジュール名および HDL 行数を示す。スタック方式 CPU の HDL 記述行数の総計は 591 行である。

表6.1と表6.2の違いは 2 階層である。新方式 CPU に存在するファイルは、bpc_file.v, control_fetch.v, file_pointer.v, reg_file_bank.v であり、スタック方式に存在するファイルは、

表6.1 新方式CPUのVerilog HDLファイル：階層別 module 名（行数）

| 0階層 (行数) | 1階層 (行数) | 2階層 | 重複のない2階層 |
|------------------|--------------------|--|---|
| cpu_novel.v (62) | fetch2.v (19) | adder_12.v bpc_file.v control_fetch.v file_pointer.v PC_12.v rom_cpu.v selector_12.v | adder_12.v (8) alu.v (15) bpc_file.v (13) control.v (33) control_fetch.v (29) EXOR_NOR_8.v (7) extend_3to12.v (11) extend_4to12.v (11) extend_4to16.v (11) extend_8to16.v (11) file_pointer.v (16) forwarding2.v (27) memory.v (13) PC_12.v (10) reg_file_bank.v (18) register_1.v (9) register_12.v (10) register_16.v (10) register_2.v (10) register_4.v (10) register_8.v (10) rom_cpu.v (44) selector_12.v (13) selector_16.v (13) selector_4.v (13) |
| | decode2.v (55) | adder_12.v control.v EXOR_NOR_8.v extend_3to12.v extend_4to12.v extend_4to16.v extend_8to16.v forwarding2.v reg_file_bank.v register_12.v register_16.v register_4.v selector_12.v selector_16.v selector_4.v | |
| | execution2.v (36) | alu.v register_1.v register_16.v register_2.v register_8.v selector_16.v | |
| | mem_access2.v (23) | memory.v register_1.v register_16.v register_8.v selector_16.v | |
| 62行 | 133行 | — | 371行 |

bpc_12.v と stackpointer2.v である。

表6.1の0階層である cpu_novel.v の HDL に対する論理シミュレーションの結果、前項 5.1.1で説明した図5.3および図5.5に示すように BPC ファイルとファイル・ポインタの正常動作を確認し、前項5.2.2で説明した図5.8に示すようにレジスタ・ファイル・バンクの正常動作を確認した。

表6.2 スタック方式 CPU の Verilog HDL ファイル：階層別 module file 名 (行数)

| 0階層 (行数) | 1階層 (行数) | 2階層 | 重複のない2階層 |
|------------------|--------------------|--|--|
| cpu_stack.v (71) | fetch2.v (22) | adder_12.v bpc_12.v extend_12to16.v PC_12.v register_1.v rom.v selector_12.v selector_16.v | adder_12.v (8) alu.v (24) bpc_12.v (19) control.v (28) EXOR_NOR_4.v (7) extend_12to16.v (8) extend_3to12.v (8) extend_4to12.v (8) extend_4to16.v (8) extend_8to16.v (8) forwarding2.v (26) memory.v (13) PC_12.v (10) reg_file.v (18) register_1.v (9) register_12.v (10) register_16.v (10) register_2.v (10) register_4.v (10) register_8.v (10) rom.v (93) selector_12.v (13) selector_16.v (13) selector_8.v (13) stackpointer2.v (20) |
| | decode2.v (45) | adder_12.v control.v EXOR_NOR_4.v extend_3to12.v extend_4to12.v extend_4to16.v extend_8to16.v forwarding2.v reg_file.v register_12.v register_16.v selector_12.v selector_16.v | |
| | execution2.v (31) | alu.v register_1.v register_16.v register_2.v register_4.v register_8.v selector_16.v selector_8.v stackpointer2.v | |
| | mem_access2.v (18) | memory.v register_1.v register_16.v register_4.v selector_16.v | |
| 71行 | 116行 | — | 404行 |

6.2 FPGA への実装

新方式 CPU を評価するために、2項で述べたスタック方式 CPU も FPGA 向けに QuartusII で論理合成し、ゲート規模や遅延時間等を比較した。

表6.3にFPGAの仕様を示す。FPGAはAltera社のCycloneIIシリーズのEP2C5で、4608個のLogic Elements (LE)、26個の4kbits+512parity bitsのRAM(計119808bits)、26個の組込用9bits x 9bits乗算器と2個のPLLを搭載している。最大I/Oピン数は158である。

図6.1にLogic Elements (LE)の回路図[12]を示す。図で組合せ回路はLook-Up Table (LUT)で構成され、レジスタ等はDフリップフロップであるProgrammable Registerで実現される。

表6.3 FPGA 仕様

| | |
|--------------------------------------|-------------------|
| Family | CycloneII/Altera |
| Device(TSMC 90nm) | EP2C5 QFP208 8bit |
| Logic Elements(LE) | 4608 |
| RAM (4kbits+512parity bits=4608bits) | 26 |
| Total RAM bits (4608bitsx26=119808) | 119808 |
| Embedded 18bitsx18bits multipliers | 13 |
| Embedded 9bitsx9bits multipliers | 26 |
| PLL | 2 |
| Maximum user I/O pins | 158 |

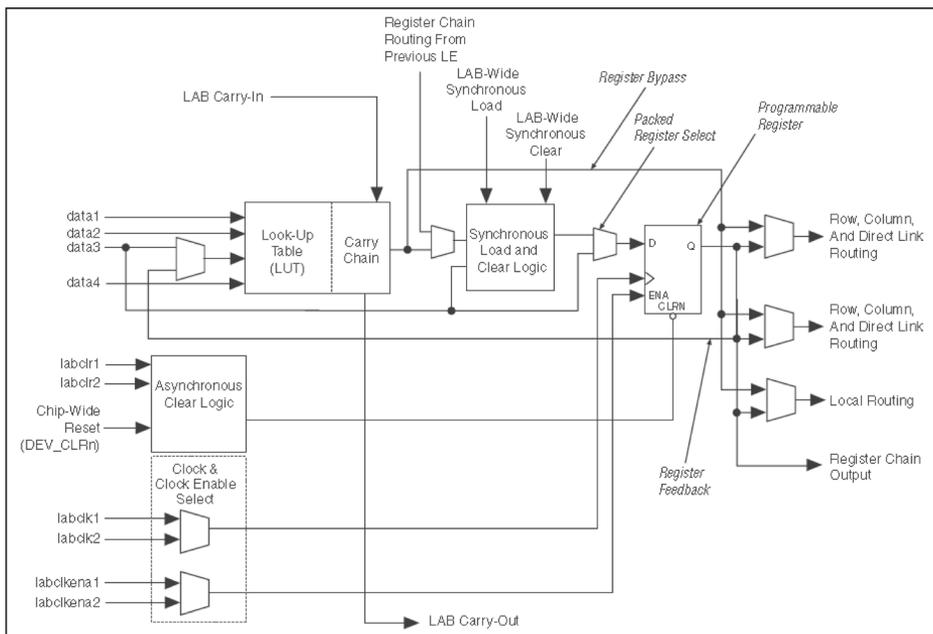


図6.1 QuartusII Logic Element

表6.4に QuartusII による新方式とスタック方式の論理合成結果を示す. QuartusII のデータ参照箇所は, Compiler tool/ Report/ Flow Summary と Compiler tool/ Report/ Timing Analyzer である.

表6.4より Total Logic Elements (LE) は新方式の方が48.8% 増加した. LE のうち組合せ回路である Total Combinational functions では, 新方式の方が47.4% 増加し, LE から構成されるフリップフロップである Dedicated Logic Registers は86.7% 増加した.

Total Combinational functions の増加は, 新方式での BPC ファイル, ファイル・ポイント回路とスタック方式でのスタックポイント回路の差であると思われる.

表6.4 論理合成結果

| 評価項目 | 新方式CPU | スタックCPU | 増分(%) | QuartusII参照箇所 |
|---|-----------|-----------|-------|--|
| Total Logic Elements (LE) | 1558 | 1047 | 48.8 | Compiler tool/ Report/Flow Summary |
| -Total Combinational functions | 1359 | 922 | 47.4 | |
| -Dedicated Logic Registers | 870 | 466 | 86.7 | |
| Total Registers | 870 | 466 | - | |
| Total memory bits (RAM bits) | 4096 | 4096 | 0 | |
| Embedded multipliers | 0 | 0 | - | |
| PLL | 0 | 0 | - | |
| Maximum user I/O pins | 22 | 22 | 0 | |
| Worst case tco (clock to out) ALU[7:0] → memory → s3 | 23.313 ns | 22.313 ns | 4.5 | Compiler tool/ Report/Timing Analyzer |
| 参考: Worst case frequency | 42.9 MHz | 44.8 MHz | -0.04 | 1/(Worst case tco) |

Dedicated Logic Registers の増加は、レジスタ・ファイルをバンク化して9倍にしたためであるが、その割に増えていないのはパイプライン・レジスタの量が多く、元々レジスタ数のベースが多かったためである。逆にいえばバンク化しても2倍以下で収まるならば、バンク化による clk 数の削減効果を考慮すると新方式を採用すべきである。

Total memory bits は新方式、スタック方式ともに4096bitsであった。これは、表6.1の cpu_novel.v/ mem_access2.v/ memory.v が両方式で同じで、HDL 記述のメモリセル部を 16bits x 256words (=4096bits) としたためである。memory.v が LE ではなく RAM で生成されたのは、memory.v の記述が表6.3の4kbits の RAM と一致したためと思われる。

使用した I/O ピンは22ピンで、RESET, CLK, 8個の LED, 8個のディップスイッチそして4個のプッシュスイッチである。

Worst case tco では、新方式 CPU とスタック方式 CPU で、ほぼ同じであった。この遅延パスは、両方式共に図5.9のデータメモリのアドレス入力である ALU からデータメモリの出力 s3までであった。メモリ内はアナログ回路でパイプライン分割できないので、メモリを含むパスは最長遅延になる傾向があるので、この結果は妥当である。従って tco 遅延時間のわずかな差は FPGA 内でのレイアウトの違いから生じていると断定できる。

以上のことから、新方式 CPU はスタック方式 CPU に比べ LE 数が48.8%増加したが、最長遅延時間は新奇回路ではなくデータメモリで律速されていることが分かった。

むすび

エンジン制御やセンサによる姿勢制御等、割込処理や分岐処理が多く、且つリアルタイム性を要求される16ビットクラスの CPU の高速化を図るため、割込命令や分岐命令に伴

う戻り先アドレスやレジスタの退避，元のプログラムへの復帰命令に伴う戻り先アドレスやレジスタの復帰操作に関して，データメモリ上のスタック領域を使用せず，その結果としてスタックへの退避命令やスタックからの復帰命令も必要としない新方式のハードウェア・アーキテクチャを開発した。

開発した方式では，割込処理や分岐処理プログラムへの分岐を実現する割込命令や分岐命令が戻り先アドレスやレジスタの退避操作も行えるようにし，元のプログラムへの復帰を実現する復帰命令が，戻り先アドレスやレジスタの復帰操作も行って元のプログラムに分岐できるようにした。

このアーキテクチャを実現するために，複数の戻り先アドレスを記憶する BPC ファイルとレジスタ・ファイル 9 個から成るレジスタ・ファイル・バンク，および割込・分岐・復帰に対応して BPC ファイルとレジスタ・ファイル・バンクを制御するファイル・ポインタを考案した。

またプログラム作成の便宜のために分岐前プログラムと分岐後プログラムとの間でデータをやり取りするためグローバル・レジスタも導入した。

HDL シミュレーションの結果，割込命令，分岐命令，復帰命令は，ファイル・ポインタを制御することで，命令発行時または次の clk で戻り先アドレスやレジスタの退避・復帰を実現していることを確認した。これにより新方式 CPU において 1 つの割込（分岐）当たり 16 レジスタで 34clk，32 レジスタで 66clk の削減が可能となった。またスタック CPU ではサポートしていなかったグローバル・レジスタの動作も確認した。

新方式およびスタック方式を論理合成し，実回路で回路規模や遅延時間を比較した結果，バンク化によるレジスタ数の増加は想定を下回り且つ遅延時間は変わらないことが判明し，新方式 CPU が有用であることが実証された。

参考文献

- [1] 三菱 32ビット RISC シングルチップマイクロコンピュータ M32R ファミリ CPU 命令セットユーザズマニュアル：1996-7-31 Ver1.00.
- [2] パターソン&ヘネシー，コンピュータの構成と設計 第2版上，日経 BP 社，東京，2002.
- [3] 黒岩将平，荒堀喜貴，権藤克彦，“実行可能コードを対象とするスケーラブルかつ部分的パス依存なバッファ・オーバフロー静的検知，”情報処理学会論文誌プログラミング (PRO) 12(3), 13-13, 2019-07-17.
- [4] 折笠雄太郎，千葉 滋，“メモリ効率の良いスレッド生成のためのスタック領域のリンクリスト化，”日本ソフトウェア科学会大会論文集 34, 331-336, 2017-09-18.
- [5] 中嶋健一郎，山田真大，長尾卓哉[他]，山崎二三雄，武井千春，本田晋也，高田広章，“ARMv6アーキテクチャを用いたメモリ保護 RTOS のユーザスタック保護の設計と評

- 価,” 情報処理学会研究報告. EMB, 組込みシステム 14, H1-H11, 2009-07-24.
- [6] 高井康浩, “高速メモリアンタフェース: DDR/GDDR-DRAM,” 電子情報通信学会技術研究報告. ICD, 集積回路 110(9), 81-82, 2010-04-15.
- [7] 井上弘士, “バッファ・オーバーフロー検出を目的としたセキュア・キャッシュの性能 / 消費電力解析,” 電子情報通信学会技術研究報告. ICD, 集積回路 105(475), 43-48, 2005-12-15.
- [8] 森若和雄, 中西恒夫, 福田 晃, “スタックのオンチップメモリへの割り当てによるキャッシュ消費電力の削減,” 情報処理学会研究報告. OS, [システムソフトウェアとオペレーティング・システム] 95, 33-40, 2004-02-26.
- [9] 佐藤弘紹, 門田暁人, 松本健一, “データの符号化と演算の変換によるプログラムの難読化手法,” 電子情報通信学会技術研究報告. ISEC, 情報セキュリティ 102(743), 13-18, 2003-03-19.
- [10] 片山清和, 安藤秀樹, 島田俊夫, “関数呼び出し時のレジスタの退避 / 復元に着目したメモリリネーミング手法,” 情報処理学会研究報告. ARC, 計算機アーキテクチャ研究会報告 150, 107-112, 2002-11-27.
- [11] 上田陽平, 山本泰宇, 関口龍郎[他], 米澤明憲, “アセンブリ言語レベルでの異種計算機間のヒープとスタックの共有機構,” 情報処理学会論文誌プログラミング (PRO) 42 (SIG03 (PRO10)), 27-39, 2001-03-15.
- [12] ALTERA CYCLONE II DEVICE HANDBOOK, VOLUME 1 CycloneII Architecture

根菜の画像解析のための形状記述についての研究ノート － フーリエ記述子による形状記述 －

淡 誠一郎

A Study Note on Shape Descriptions for Image Analysis of Root Vegetables － Fourier Descriptors －

Seiichiro Dan

概 要

果物や野菜の形状は収穫・加工の効率に大きくかわる。形状は生育環境や土壌にも左右されるが、品種独自の形質はやはり遺伝的要因に依るところが大きいと考えられる。例えば、トマトは複数の形質遺伝子の組み合わせでその果実の形が決まることがわかっている。ダイコンなどの根菜も同様に遺伝子が根形にかかわると予想されており、その解明が期待される。この目的のため、筆者を含む研究グループでは、ダイコンの根形を特徴づける複数の形質とその量的な測定方法に取り組んできた^[1,2]。

遺伝子と根形の関係を明らかにするには、まず現実の個体の形状を的確に、また解析に適した方法で記述する必要がある。形状記述方法の候補は複数あるが、本稿では、動植物の形態学的解析^[3,4]、病理診断^[5,6]、工業製品のデザイン^[7,8]など幅広く応用例の見られるフーリエ記述子を取り上げ、根菜の形状記述を試みた結果について報告する。

1. 対象画像と前処理

図1に本研究で対象とする実画像の一例と記述の対象となるシルエット画像を示す。ダイコンの形状は実際には3次元的であるが、おおむねその軸に関して回転対称であることが多い。本研究でいうダイコンの形状とは、図1の左の写真のように、ダイコンを平面に接地させて撮影した写真を入力とした場合の輪郭形状である。実画像には図1の例のように個体識別のためのプレートやサイズ指標が含まれることもあるし、葉の一部が含まれて



図1 入力画像と記述対象のシルエット画像

いたりもする。

実画像から記述対象となるシルエット輪郭データをえるための前処理について簡単に説明する。前処理の手順はつぎの通りである。

- (1) ダイコン主根部のセグメンテーションと2値画像化
- (2) 膨張収縮・平滑化によるシルエット形状整形
- (3) 輪郭点列データの抽出

以下に手順の概要を説明する。

(1) ダイコン主根部のセグメンテーションと2値画像化

領域抽出アルゴリズム GrabCut を用いて、自動または少しのヒント情報を与えることで先の実画像からシルエット画像が得られるツールを作成した。プログラム開発に用いた言語は python である。GrabCut の抽出結果は濃淡画像であるので、これをしきい値処理により2値化してシルエット画像をえる。一般に濃淡画像から2値画像を得る場合、そのしきい値設定が問題となるが、GrabCut 自体がすでに濃淡差をひとつの基準として前景と背景を分離しているのので、あらためてしきい値処理する必要はなく、前景を255、背景を0とすればよい。図2の左→中央は自動セグメンテーションの実例である。

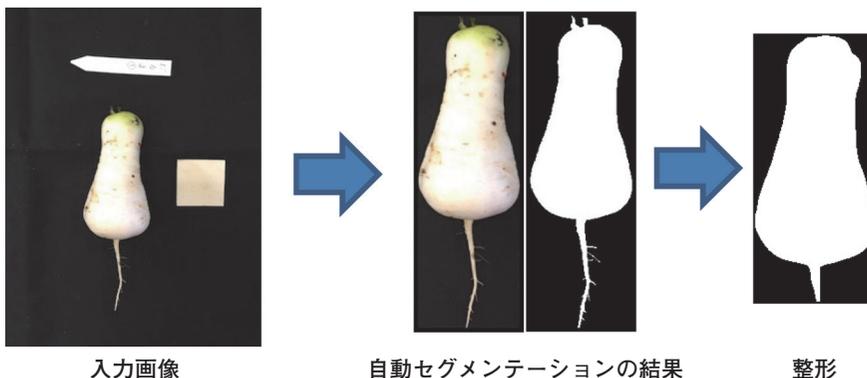


図2 前処理の流れ

(2) 膨張収縮・平滑化によるシルエット形状整形

図2でもわかるように、実画像をしきい値処理して得られるシルエット画像はヒゲ根などにより必要以上に複雑な凹凸や孔が生じるものである。さらに、図2の例には見られないが個体の色や汚れ、傷による凹凸も生じる。これらは主要部の形状記述においては雑音となるので、除去しておきたい。

まず、シルエット画像中の最大白連結成分の最外輪郭を抽出し、その内部を塗りつぶすことで内部の孔を除去する。つぎに、closing（収縮→膨張）と opening（膨張→収縮）により、ヒゲを除去し、最後にガウシアンフィルタによる平滑化で輪郭形状の複雑な凹凸を除去する。図2の右の画像が実際の整形後の画像であり、これを対象個体のシルエット、すなわち形状と考える。

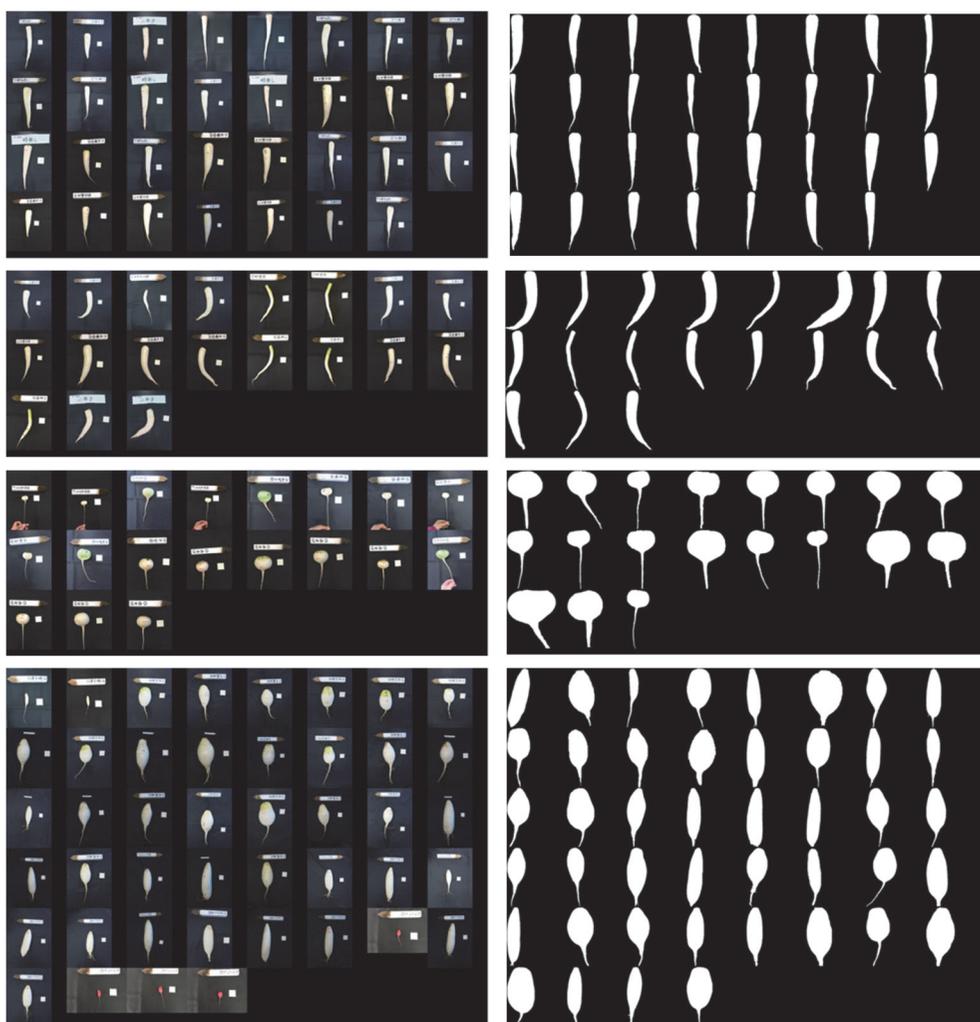


図3 入力画像とえられたシルエット画像の例
(原画像は神戸大学大学院農学研究科附属食資源教育研究センター提供)

さまざまな品種のダイコンについて、シルエット抽出を試み、いずれにおいても主要部の形状特徴が保存され、ひげ根やそれによる孔が除去された結果がえられることが確認できた。

その一部を図3に示す。一部バッチ処理によるシルエット抽出で失敗した個体について再処理をかけているため、入力画像の並び順序とシルエット画像のそれとは一致していない。

なお、この研究で用いているダイコンの画像データは、神戸大学大学院農学研究科附属食資源教育研究センターよりご提供いただいたものである。ここに記して感謝の意を表す。

(3) 輪郭点列データの抽出

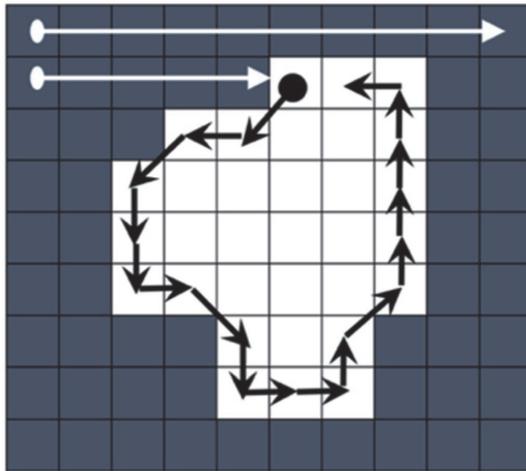


図4 輪郭追跡

ここまでの処理により、画像内には孔のない白連結成分が一つだけ含まれているはずであるので、単連結用の輪郭線追跡アルゴリズムにより輪郭線データを得ることができる。

例えば、図4の図形であれば、輪郭データは次のようなりスト構造の配列で表現される。

[[5,1], [4,2], [3,2], [2,3], ..., [7,2], [7,1], [6,1]]

この記述は他の形状記述を生成するための1次データとなるものであり、対象図形の個々のシルエットを記述するもっとも正確な記述法である。この記述は品種の特徴にノイズや個体変動が加わったものであるため、いかにしてこの記述から品種の共通性を示す特徴と個体の個性である特徴を分離抽出するかが問題となる。

2. シルエット形状の記述

上述のように、本研究では2次元的なシルエットを形状と考えている。シルエットはその輪郭線と同一視できるので、ここでは輪郭線の記述方法に絞って考えることにする。上で述べた前処理により一応輪郭線情報は記述されているわけであるが、単なる座標の羅列に過ぎないのでそのままでは解析や識別には利用できない。

輪郭線の記述方法としては、

- チェインコード (方向差分チェインコード)
- 極座標表現
- フーリエ記述子
- ベジエ曲線に代表されるパラメトリック曲線

などが考えられる。

チェインコードは線図形の一般的な表現方法である。端点もしくは閉曲線上に取った任意の点を開始点とし、隣接する次の点が8方向のうちのどの方向にあるかを記述する。この記述法はたしかにシルエットを正確に記述でき、座標そのものの記述と比べてデータ圧縮できるが、少数のパラメータで近似図形を生成したり、図形の類似性を判定したりする手がかりにはならず、本研究の目的には合わない。

極座標表現は図形内部に原点を取り、原点から見た輪郭点の距離を角度の関数とみなす表現方法である。重心を原点に設定し図形の長さや方向を正規化するなどの方法で図形同士の類似度を比較することができるが、輪郭が入り組んでいると輪郭点の距離が角度の関数とならないケースが発生する。典型的なダイコンやカブの形であれば問題ないと思われるが、湾曲の大きい個体や、ひょうたん型等で支障がでる。

フーリエ記述子は、平面上の連続曲線を、端点から曲線に沿ってはかった道のりの長さ s の関数として記述し、その関数をフーリエ級数展開した際の係数を形状の記述ととらえる。フーリエ級数展開は連続無限の周期関数を前提とした数学的操作であり、フーリエ記述子は、曲線の何をどう周期関数とみなすかによっていくつかのパリエーションが存在する。

ベジエ曲線およびそれを発展させた B スプライン、有理ベジエ曲線、NURBUS 曲線などのパラメトリック曲線は、いずれを用いても任意の精度でシルエット形状を近似できるが、データ解析や識別が目的であれば、表現能力が必要十分な範囲で、次元数は低いほど望ましい。パラメトリック曲線の中では、ベジエ曲線が最も有望であろう。

本研究では、以上の記述方法の中でフーリエ記述子を取り上げ、実際のダイコンの画像を用いて、各種フーリエ記述子のうち、どれが有用であるか、係数の次数を途中で打ち切った場合にどの程度の近似画像が得られるかについて調査する。

3. フーリエ記述子によるダイコンのシルエット記述

先にも述べたが、フーリエ記述子にはいくつかのバリエーションが存在する。本研究では、実際に

- Z型フーリエ記述子
- 楕円フーリエ記述子
- G型フーリエ記述子
- P型フーリエ記述子

を実装して曲線の記述と近似を試みた。

① Z型フーリエ記述子

閉曲線上に等間隔に N 個の標本点 P_0, P_1, \dots, P_{N-1} を取って、閉曲線を等辺多角形で近似する。各辺の方向ベクトル $\overrightarrow{P_i P_{i+1}}$ が基準辺の方向ベクトル $\overrightarrow{P_0 P_1}$ となす角 θ_i を偏角、

$$\Phi_i = \theta_i - \frac{2\pi(i \bmod N)}{N} \quad \dots(1)$$

を正規化偏角関数とよぶ。

$i > N$ について、 $P_i = P_{i \bmod N}$ であると考え、 $\Phi_i: i = 0, 1, \dots$ は周期 N ごとに 0 に帰する周期関数となり、

$$\Phi_i = \sum_{k=-\frac{N}{2}+1}^{\frac{N}{2}} c_k \exp\left(j \frac{2\pi k i}{N}\right) \quad \dots(2)$$

と離散フーリエ変換できる。ここで、

$$c_k = \frac{1}{N} \sum_{i=0}^{N-1} \Phi_i \exp\left(-j \frac{2\pi k i}{N}\right) \quad \dots(3)$$

である。

始点 (x_0, y_0) とフーリエ記述子 c_k がわかっているならば、次の計算式により、近似多角形の頂点座標 $(x_i, y_i): i = 0, 1, \dots, N-1$ を求めることができる。

$$\left. \begin{aligned} \theta_i &= \Phi_i + \frac{2\pi(i \bmod N)}{N} \\ x_i &= x_0 + \delta \sum_{k=0}^i \cos\theta_k \\ y_i &= y_0 + \delta \sum_{k=0}^i \sin\theta_k \end{aligned} \right\} \quad \dots(4)$$

ここで、 δ は 1 辺の長さである。

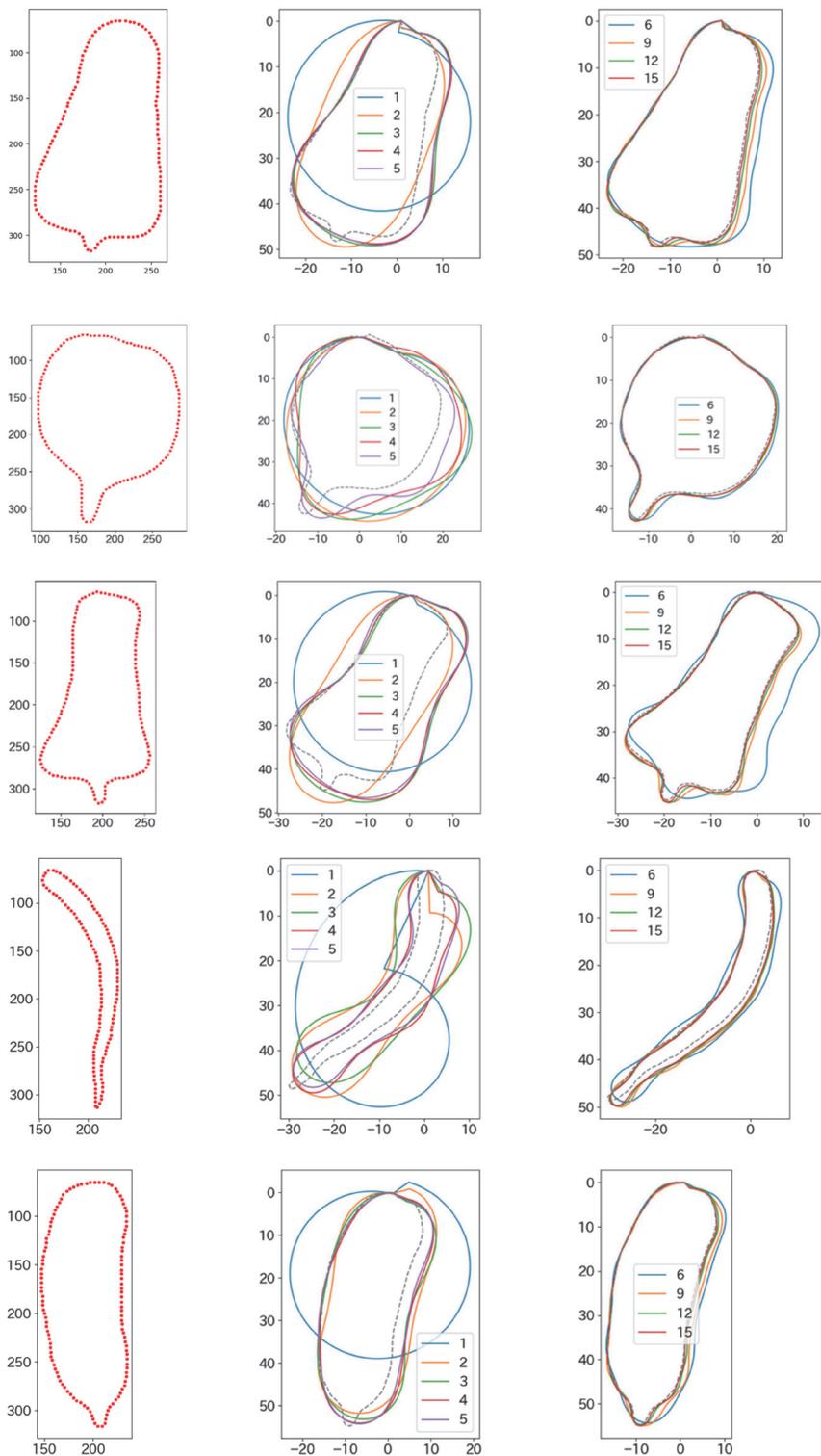


図5 Z型フーリエ記述子による復元形状

図5に実際のダイコンのシルエット画像をZ型フーリエ記述子を得て、 $0 \sim \pm n$ 次 ($n = 1 \sim 5, 6, 9, 12, 15$) の記述子から形状復元した結果を示す。なお、処理対象の入力のシルエット画像は、主要部のみが、若干のマーヅンを含めて高さが256画素となるよう正規化されるようプログラムされている。輪郭点のサンプル数は128とした。比較のため、これ以降の実験も同じ条件で行う。

復元結果からわかるように、Z型フーリエ記述子から復元される近似図形は最後の頂点と始点との間にギャップが生まれることが多い。このギャップはZ型フーリエ記述子が偏角そのものを変換対象としており、始点から計算で求められた角度と辺の長さから順次形状復元していく際に、失われた高周波成分による角度の誤差と量子化誤差の両方が蓄積していく結果であると考えられる。

② 楕円フーリエ記述子

曲線上に任意の点Sを定め、Sから一定速度で曲線上を移動しながら周回する動点Pを

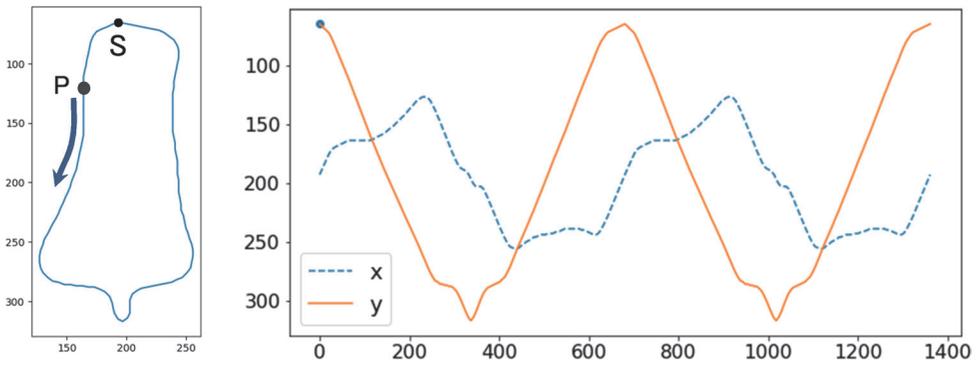


図6 閉曲線上のx座標とy座標の変化

考えると、点Pのx座標、y座標はそれぞれ独立した周期関数とみなせる(図6)。

このそれぞれを次式のように離散フーリエ変換することで得られる三角関数の係数が楕円フーリエ記述子である。

$$\begin{aligned}
 x_i &= \frac{a_0}{2} + \sum_{k=1}^{N-1} \left(a_k \cos\left(\frac{2\pi ki}{N}\right) + b_k \sin\left(\frac{2\pi ki}{N}\right) \right) \\
 y_i &= \frac{c_0}{2} + \sum_{k=1}^{N-1} \left(c_k \cos\left(\frac{2\pi ki}{N}\right) + d_k \sin\left(\frac{2\pi ki}{N}\right) \right)
 \end{aligned} \quad \dots (5)$$

ここで、

$$\begin{aligned}
 a_k &= \frac{1}{N} \sum_{i=0}^{N-1} x_i \cos\left(\frac{2\pi ki}{N}\right), & b_k &= \frac{1}{N} \sum_{i=0}^{N-1} x_i \sin\left(\frac{2\pi ki}{N}\right) \\
 c_k &= \frac{1}{N} \sum_{i=0}^{N-1} y_i \cos\left(\frac{2\pi ki}{N}\right), & d_k &= \frac{1}{N} \sum_{i=0}^{N-1} y_i \sin\left(\frac{2\pi ki}{N}\right)
 \end{aligned}
 \quad \dots (6)$$

であり, $a_k, b_k, c_k, d_k: k = 0, 1, \dots, N-1$ が楕円フーリエ記述子である.

楕円フーリエ記述子は理論に複素数概念が含まれないので直感的に理解しやすいが, 回転によって関数の波形が x_i, y_i とともに変形するので, 素のままの記述子は回転不変性を持たない. このため, 主成分分析などに利用する際には a_1, b_1, c_1, d_1 の値をもとに他の次数の記述子を標準化してやらねばならないという手間がある. a_1, b_1, c_1, d_1 の値は近似楕円を規定する量であり, この標準化操作は, 近似楕円の軸を座標軸とするような座標変換に相当する.

図7に図5と同じ対象画像に対応する, 楕円フーリエ記述子からの形状復元結果を示す.

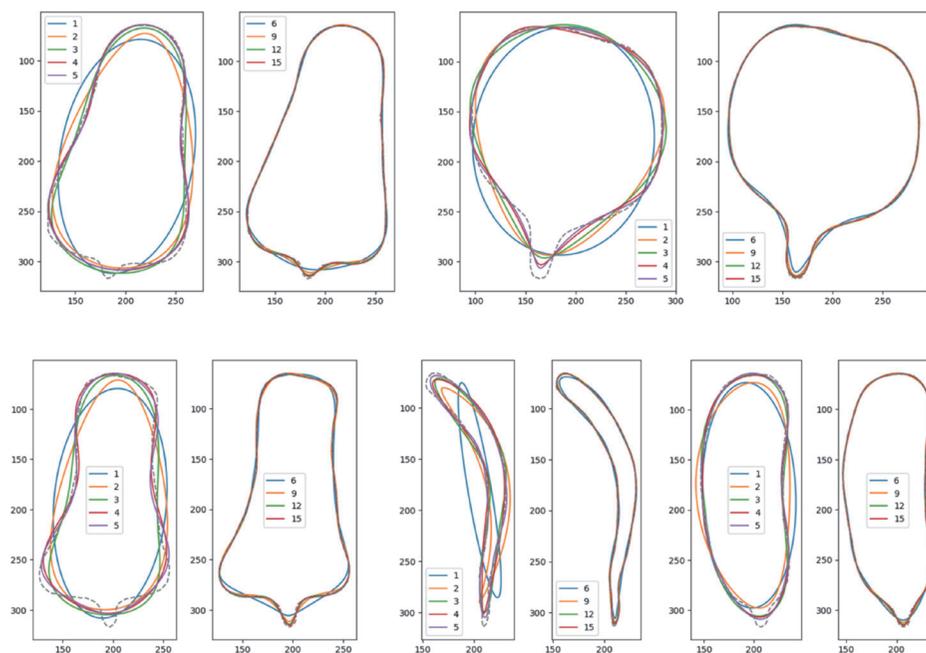


図7 楕円フーリエ記述子による復元形状

③ G型フーリエ記述子

平面座標 (x, y) と複素数 $z = x + jy$ を同一視し, 輪郭線上に等間隔に取った N 個の標本点を N 個の複素数 $z_i: i = 0, 1, \dots, N-1$ に対応させる. ただし, $z_i: i = 0, 1, \dots, N-1$ の並びは輪郭線をたどった時の標本点の出現順であり, 先頭の標本点と末尾の標本点も, 他の標本点間と同じ間隔だけ離れて隣接してなければならない.

いま, $z_i; i = 0, 1, \dots$ を周期 N で 1 周する周期関数とみなし, N は閉曲線を近似するのに十分な数であるものとするれば,

$$z_i = \sum_{k=-\frac{N}{2}+1}^{\frac{N}{2}} c_k \exp\left(\frac{j2\pi ki}{N}\right) \quad \dots(7)$$

と離散フーリエ変換できる. ここで,

$$c_k = \frac{1}{N} \sum_{i=0}^{N-1} z_i \exp\left(-j\frac{2\pi ki}{N}\right) \quad \dots(8)$$

であり, この係数 c_k を **G 型フーリエ記述子** とよぶ.

図 8 に G 型フーリエ記述子からの形状復元結果を示す. 先の楕円フーリエ記述子は G 型フーリエ記述子の実部と虚部を別々に扱ったものと考えられ, 両者は数学的に等価である. 復元結果もその通りになっており, 図 7 と図 8 の結果はきれいに一致している.

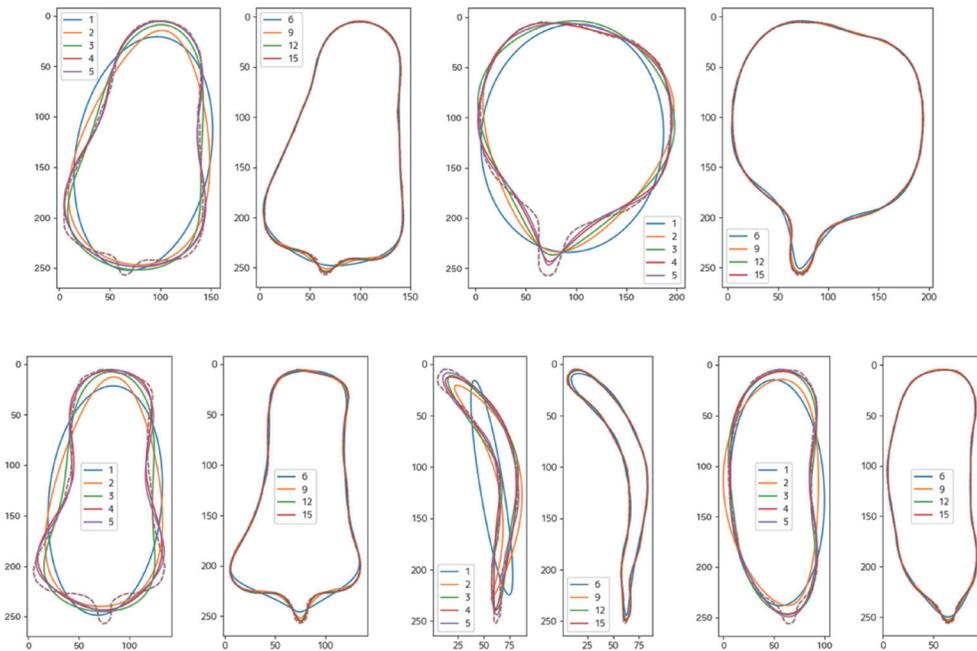


図 8 G 型フーリエ記述子による復元形状

楕円フーリエ記述子が回転に対して不変性を持たないという点は, G 型フーリエ記述子の実部と虚部を別々に見た場合には不変性を持たないということに相当するわけであるが, G 型フーリエ記述子の絶対値は回転不変であり, 回転分だけ記述子の位相が回転することになる.

④ P型フーリエ記述子

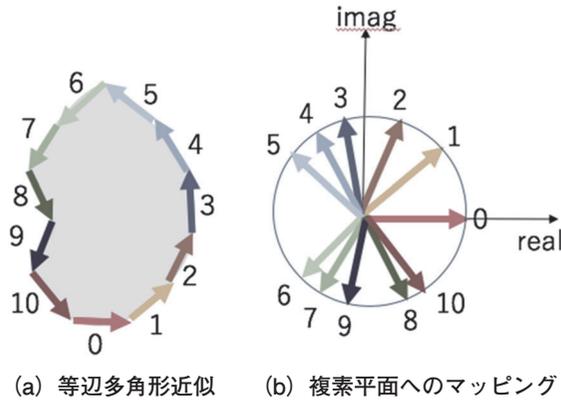


図9 P型フーリエ記述子

G型で行なったのと同様に、輪郭線上に等間隔に取ったN個の標本点 $P_i = (x_i, y_i): i = 0, 1, \dots, N-1$ に複素平面上の点 $z_i = x_i + jy_i$ を対応づける(図9)。G型では z_i を周期関数ととらえたが、P型では、ベクトル $\overrightarrow{P_i P_{i+1}}$ とx軸(実数軸)のなす角 θ_i の列を考える。この角をベクトル $\overrightarrow{P_i P_{i+1}}$ の偏角とよぶ。いま、 $w_i = \cos\theta_i + jsin\theta_i$ と定義すると、偏角 θ_i は複素平面上の複素数 w_i の偏角に一致する。

$w_i: i = 0, 1, \dots$ は周期Nの複素周期関数とみなすことができるので、

$$w_i = \sum_{k=-\frac{N}{2}+1}^{\frac{N}{2}} W_k \exp\left(j \frac{2\pi ki}{N}\right) \quad \dots(9)$$

と離散フーリエ変換できる。ここで、

$$W_k = \frac{1}{N} \sum_{i=0}^{N-1} w_i \exp\left(-j \frac{2\pi ki}{N}\right) \quad \dots(10)$$

である。この、 W_k をP型フーリエ記述子と呼ぶ。

等辺N角形の一つの辺の長さを δ とすれば、

$$w_i = \cos\theta_i + jsin\theta_i = \frac{x_{i+1} - x_i}{\delta} + j \frac{y_{i+1} - y_i}{\delta} = \frac{z_{i+1} - z_i}{\delta} \quad \dots(11)$$

であり、 $w_i: i = 0, 1, \dots, n$ と始点 z_0 が与えられたならば、 $z_n = z_0 + \delta \sum_{i=0}^n w_i$ という計算式で順次図形の頂点を求めていくことができ、形状復元ができる。

図10にP型フーリエ記述子からの形状復元結果を示す。

P型フーリエ記述子は基準方向に対する偏角をフーリエ変換するという点ではZ型と同

じであるが、偏角そのものではなく、偏角に対応する複素平面の単位円上の点の座標を変換対象にする。このため、Z型のような正規化の操作が不要であり、Z型とは違って低い次数であっても復元形状はきちんと閉じる。さらに、本稿では取り上げないが、P型フーリエ記述子は閉曲線だけでなく、開曲線にも適用可能である。

以上、各種フーリエ記述子を実装してダイコンのシルエットに適用してみたわけであるが、どのフーリエ記述子でも12~15次程度までの記述子を用いれば十分個体の形状は表現できることがわかった。品種の特定はそれ以下の次数でおそらく十分であり、大まかな品種識別には6~7次、詳細な識別でも高々10次程度で可能ではないかと推測される。

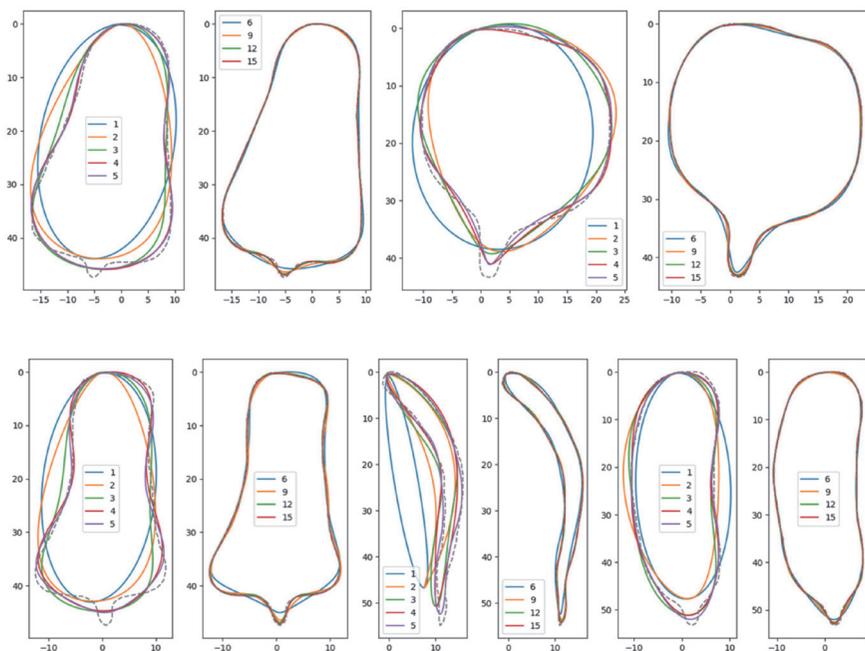


図10 P型フーリエ記述子による復元形状

参考文献

- [1] 淡 裕美子, 淡 誠一郎, 吉田康子: “画像解析を用いたダイコンの根形を特徴づける形質の選定とその評価方法の確立”, 育種学研究 19, 別2, p.70 (2017).
- [2] 淡 裕美子, 淡 誠一郎, 吉田康子: “画像解析を用いたダイコンの根形を特徴付ける形質による多様性評価”, 育種学研究 21, 別1, p.101 (2019).
- [3] 鄭澤宇, 岩田洋佳, 二宮正士, 田村義保: “P形フーリエ記述子に基づくハナス花弁の部分形状特徴の定量的評価”, 育種学研究 7,3, pp.133-142 (2005).
- [4] 古田尚也, 二宮正士, 高橋信夫, 大森 宏, 鷺飼保雄: “ダイズ小葉形の楕円フーリエ記述子に基づく主成分スコアによる定量的評価”, Breeding science 45, 3, pp.315-320

(1995).

- [5] 川村陽子：“フーリエ記述子を用いた脳梁形状の性差の解析”，岐阜大学工学部 博士学位論文 (2006).
- [6] 中嶋典子，石井晃太郎，岩田洋佳，大鹿哲郎：“楕円フーリエ記述子に基づく水晶体形状発達の定量的解析”，臨床眼科 65, 7, pp.137-1142 (2011).
- [7] 小林智美，篠田一馬，石川智治，長谷川まどか，加藤茂夫：“楕円フーリエ記述子を用いた衣服の輪郭形状の分類に関する一検討”，映像情報メディア学会技術報告39, 31, pp.5-8 (2015).
- [8] 小森政嗣，川村 智，横山卓未，森下佳昌：“楕円フーリエ記述子を用いたボディソープボトル形状の分析と評価”，日本包装学会誌 21, 6, pp.479-491 (2012).

市民ランナーに対するマラソン前の30km走の効果検証のための研究プロトコール
- ランダム化比較試験デザインを用いた研究 -

山 内 武¹・高 橋 秀 人²・得 居 雅 人³
足 立 哲 司⁴・杉 山 喜 一⁵・串 間 敦 郎⁶
中 嶋 南 紀⁷・前 田 和 良⁸・元 根 朋 美⁹
松 尾 信之介¹

Study protocol for the effect of the 30-kilometer
training run before marathon race in recreational runners
- Randomized controlled trial

Takeshi Yamauchi¹・Hideto Takahashi²・Masato Tokui³
Tetsuji Adachi⁴・Kiichi Sugiyama⁵・Atsuro Kushima⁶
Nanki Nakashima⁷・Kazuyoshi Maeda⁸・Tomomi Motone⁹
Shinnosuke Matsuo¹

研究代表者：山内 武（大阪学院大学健康スポーツ科学）
〒617-0002 大阪府吹田市岸部南2-36-1 TEL:06-6381-8434（内線 5327）
E-mail：yamauchi@ogu.ac.jp
Correspond Author: Takeshi Yamauchi
Address: 2-36-1 Kishibe-Minami, Suita-shi, Osaka, Japan, 617-0002
E-mail：yamauchi@ogu.ac.jp

Keywords: Study protocol, RCT, 30-kilometer training run

-
- 1 大阪学院大学 Osaka Gakuin University
 - 2 国立保健医療科学院 National Institute of Public Health
 - 3 九州共立大学 Kyushu Kyoritsu University
 - 4 大阪体育大学 Osaka University of Health and Sports Sciences
 - 5 北海道教育大学 Hokkaido University of Education
 - 6 宮崎看護大学 Miyazaki Prefectural Nursing University
 - 7 ミズノ・ランニング・ステーション Mizuno Corporation
 - 8 大阪経済大学 Osaka University of Economics
 - 9 帝塚山大学 Tezukayama University

I 序論

スポーツ科学において、様々なトレーニング法に対する研究が進展し、エビデンスに基づくトレーニング法が一般化しつつあるが、エビデンスレベルが高いランダム化比較試験(RCT)デザインを用いた研究は、この分野でほとんど実施されてこなかった¹⁾。また、「体育科学研究における最大のピットフォール(落とし穴)は、メカニズム(トレーニング法の仕組み)を研究しなければならないと思いついでいることかもしれない」との指摘もある²⁾。

現在、スポーツ科学の多くの研究者は、トレーニングを実施すればどう生体は変化するかなどといった研究課題に取り組んでいる。あるいは、細胞レベルに遡って研究を進めている研究者も多い。多くの研究者は、メカニズム(トレーニング法の仕組み)の解明を通じて、トレーニング法の改善に重要な役割を果たしてきた。しかし、エビデンスに基づくトレーニング法を確立していくためには、トレーニング法自体の妥当性をRCTなどエビデンスレベルの高い研究デザインを用いて検証することが必要である。

介入研究における比較では、様々なバイアス(選択バイアス、情報バイアス等)の排除および交絡因子の制御が重要である。理論的にはRCTは未知のバイアスや交絡因子を含めて制御することができる研究デザインであり³⁾、RCTの実施にあたっては、関係者で内容を共有するためや、研究開始前の情報で研究を立案した証のために、研究プロトコル(手順)を作成することが重要となる。特に「主要評価項目(primary endpoint)」に差が出やすくなるように研究を誘導することや、結果を見て差が出そうな項目を解析することは、研究者が行いがちな陥りやすい課題であるが、これでは恣意的に結果を歪めてしまうことになる。また、様々な検定を行えばよく見つけた「有意差」を提示するようなことを行うと、統計学的には「本来は差がないにも関わらず誤って差があると結論する」という第一種の過誤による結果を提示してしまう可能性が高くなる(信頼性が下がる)し、また「検定の多重性(検定を繰り返すことにより、帰無仮説の棄却の判断の確からしさが下がる)⁴⁾」という問題が生じてしまう。

このような課題や問題が起こらないようにするためには、研究を始める前にどのような項目をどのように解析するかなどを、研究プロトコルに定める等、客観性を高くするような工夫を行うことが重要である。また論文誌には、有意差が生じた結果が多く掲載されるという傾向がある(出版バイアス)ことから、現在臨床試験ではRCTに関して、研究の事前登録制が実施されているし、疫学分野でも研究プロトコル論文を掲載するなど行われている⁵⁾。スポーツ科学の分野においても、RCTを用いた研究を実施する場合には、研究プロトコルを事前に公開するなど上記の課題等への対処を考える必要があると考える。われわれは研究の客観性を高めるために、研究グループが現在実施している研究に関し、そのプロトコルを公開することとした。なお、本研究では、詳細な研究プロトコル

ルのみを公開し、研究結果は今後、別論文として公開する予定である。

さて、我々の研究グループは、ランニングにおけるトレーニング法の効果を検証するため、「市民ランナーに対するマラソン前の30km走の効果」の検証について取り組んできた。本研究は、市民マラソン関係者（コーチ、市民ランナー等）において「効果がある」と考えられている「マラソンレース前の30km走の効果」について、十分な効果があるかどうかを、エビデンスレベルの高いRCTデザインで検証することを目的とし、本稿ではその研究プロトコルをプロトコル論文として報告する。なお、このプロトコルは、JCOG（日本臨床腫瘍グループ）のプロトコルマニュアル⁶を参考に作成している。本研究のプロトコルに基づき、実際の介入研究で得られた結果は、今後国際的な医学系のRCT報告のためのガイドラインであるCONSORT 2010 Statement⁷に基づき、通常の研究論文として投稿する予定である。

II 研究プロトコル

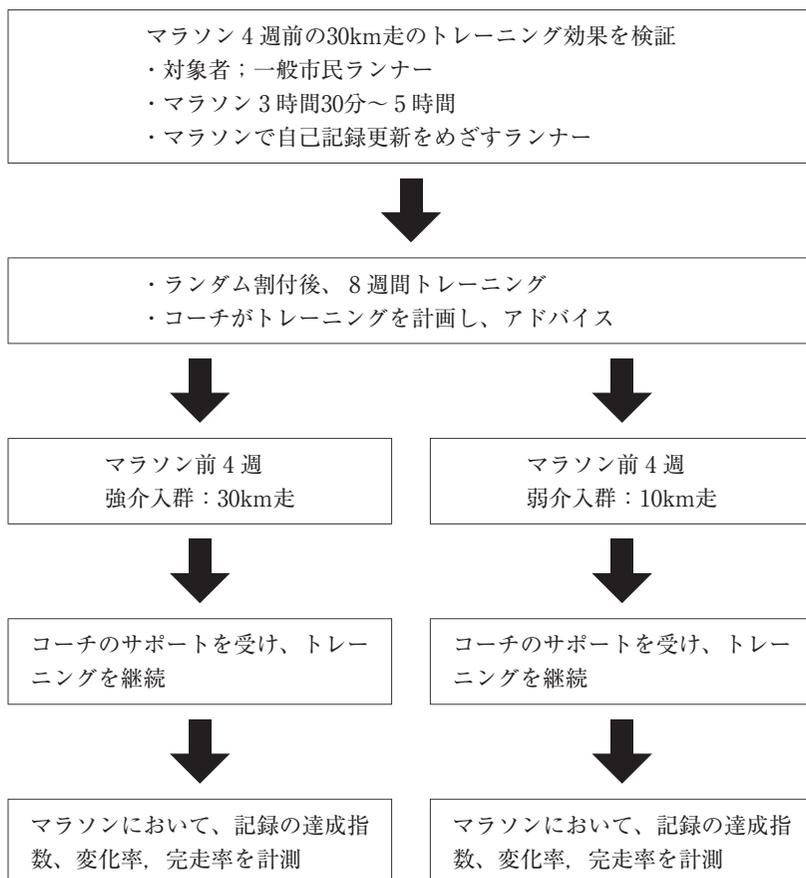


図1 研究の概要（研究シェーマ）

1. 研究の目的

本研究の目的は、マラソンのパフォーマンスに関し、トレーニング法「マラソン4週前の30km走」の有効性を明らかにすることである。対象者を3時間30分～5時間程度の自己記録を持ち、自己記録更新をめざす一般市民ランナーとする。記録を更新するためにトレーニングしている対象者への道義的倫理性に配慮し、トレーニング非実施群を設けず、トレーニングに関する強介入群（マラソン4週前の30km走）と弱介入群（マラソン4週前の10km走）の2群について比較する。比較として、主要評価項目（Primary endpoint）「ゴール記録の達成指数」と4つの副次的評価項目（Secondary endpoints）「ランニングスピードの変化比」、「前半ランニングスピードと後半ランニングスピードの変化比」、「完走割合」、「マラソン後の心身のコンディショニング状態」として、RCT デザインを設定した。なお、本研究は疫学・統計モデルを基盤としており、スポーツ科学で通常用いられる用語と異なることがある。

2. 背景と研究計画の根拠

本研究の実施根拠は以下のとおりである。

マラソン前の30km走は、競技スポーツとして高い達成を目指すアスリートでは標準的なトレーニングであり、特にマラソン直前の30km走がレースに向けての仕上がり状況を判断する上で重要とされている。これに関し、指導書やアスリートのトレーニングの影響から、マラソンに挑む多くの市民ランナーにとっても、30km走は効果的で有効なトレーニング法であり、マラソン4週前（1ヶ月前）までに最低1回だけでも取り組みたいトレーニング法だと考えられている⁸⁾。しかしその一方で、市民ランナーにとっては、30km走にこだわらずハーフマラソン（21km）程度で十分ではないかとの意見も指導書の中には存在する⁹⁾。現状は市民ランナーにおけるマラソン前の30km走の有効性について十分なエビデンスがないという状況である。昨今、市民がマラソンに参加中に健康問題が生じるなどの多くの報道があり、現在、有効と考えられている「マラソン前の30km」走についてもエビデンスに基づいた評価が必要と考える。

1) 研究対象者

本研究の対象者はマラソン3時間30分～5時間程度の自己記録を持ち、自己記録更新をめざすランナーとする。この対象者は、近年の大規模なマラソン参加者の最も人数が多い層であり、「マラソン4週前の30km走」への関心が高く、研究結果の与える影響が大きい層と想定される。

2) トレーニングスケジュール設定の根拠とその実施法

スポーツ科学に基づくマラソン前の標準的トレーニング法（トレーニングスケジュール）は確立していないので、通常は十分な指導経験を持ったランニング指導者が提供する

トレーニングスケジュールが、ランニング指導には用いられてきている。本研究においても、十分な指導経験を持ったランニング指導者（研究代表者）が、研究対象者に対してマラソンでの自己記録更新をめざすためのトレーニングスケジュールを提供することとする。

研究対象者のトレーニングについて、以下の方法で実施していく。

マラソンレース 8 週間前（2 か月前）からレースに向けてトレーニング介入を開始する。事前に4つのサブグループ（～3時間40分、3時間40分～4時間00分、4時間00分～4時間30分、4時間30分～）に分け、それぞれごとにトレーニングスケジュールを提供する。4つのグループごとに4名の担当コーチ（共同研究者）がつき、それぞれのグループのアドバイスと管理を行う。

3) 研究デザイン、研究対象者の登録

トレーニング法「マラソン 4 週前の30km走」を強介入群、通常のトレーニング法である「マラソン 4 週前の10km走」を弱介入群とし、介入効果以外を公平とするため、ランダム化比較試験（RCT）デザインを採択する。この際、研究対象者がどちらの介入群に属しているかを事前に知っていると、「その要因によりトレーニング状況に差が生じる」などのバイアスが生じる可能性があるため、このバイアスを排除するためにブラインド化（ダブルブラインド、二重盲検）として¹⁰⁾、担当コーチ（共同研究者）および研究対象者には、どちらの群（強介入群（30km走）、弱介入群（10km走））に割り当てられたかの情報を伏せる。

しかし研究対象者の体調管理の面から、主介入（30km走、もしくは10km走）実施前にどちらの群に割り付けられたかの情報提供が必要となるため、主介入実施 7 日前に、その情報を研究対象者のみに通知する。しかし、トレーニング管理を行う担当コーチには、個人情報と割り付け結果は伏せる。本研究は研究対象者にのみ強介入群・弱介入群のどちらが割り付けられたかの情報提供を行ったオープン試験である。

本研究では、レースタイム（10kmごとのラップタイムを含む）が公開されている大規模な都市型マラソンレースを3つ（大阪マラソン、泉州国際マラソン、京都マラソン）選定し、参加予定者から研究対象者を募る。研究対象者をマラソン大会、記録レベルにより層別ランダム化し、30km走を実施する強介入群と、通常のトレーニングである10kmを実施する弱介入群の二群について、実際のマラソンレース 8 週間前からトレーニングを実施する。

研究対象者には、募集の段階でトレーニングにおいて30km走を実施する可能性があることを明示しておく。強介入群には、マラソン 4 週前に、ペースセッターと給水が準備されている30km走を実施する。弱介入群には、同時期に通常のトレーニングである10kmペース走を実施する。30km走、10km走においては、実際のマラソンの目標タイムからペースを割り出し、複数のグループに分かれて、ペースセッターの先導のもと、5 kmごとのラップタイムを計測して実施する。

30km走、10km走以外のトレーニングを強介入群、弱介入群ともに同様なものにするため、研究対象者に対して、マラソンでの自己記録更新をめざすための効率的なトレーニングスケジュールを提供し、それにしたがってトレーニングを実施してもらう。このトレーニングスケジュールは、20km走程度を上限とするトレーニングで自己記録更新を狙った効率的で安全性の高いものとする。トレーニングスケジュールの作成は、すべて研究代表者が行い安全で効果的なものを提供する。原則として、週に3回トレーニング実施を目安として、トレーニングスケジュールを作成し、担当コーチへ連絡する。研究対象者への提供は、担当グループのコーチがメールで行い、スケジュールの調整や研究対象者からの相談に担当コーチが応じる。

トレーニングスケジュールにしたがって、トレーニングを研究対象者が各自実施する。期間中に合同練習会4回実施するが、参加は任意とする。合同練習会はトレーニングスケジュールの中に組み込んでおき、合同練習会に参加できない研究対象者も個人で実施可能な内容にしておく。

トレーニングスケジュールの実施率を高いものにするため、担当コーチが研究対象者からメールで相談を受け、またトレーニングの実施状況の把握を「Google form」を活用しトレーニングログを記録させ、トレーニング実施状況を把握しトレーニング実施率を算出する。なお、トレーニングログは走行距離と運動強度の情報を含むものとする。

マラソン4週前に強介入群では30km走を実施し、弱介入群は10km走（通常のトレーニングと同等）を実施する。これ以外は、強介入群と弱介入群では、同様なトレーニングスケジュールとする。マラソンの直前までトレーニングスケジュールを提供し、研究対象者がレースに向けて、体調が整うようにテーパリングしていく。

最終的に、マラソンの記録を用いてパフォーマンスを測定し評価項目（endpoint）とする。インターネット上で公開されているレース記録（10km毎）を、研究対象者のナンバーカード等から取得する。事前に研究対象者に記載させた過去3年間の自己記録とマラソンのゴール記録から記録の達成率を算出し、主要評価項目（primary endpoint）とする。また、ランニングスピードの変化比、前半ランニングスピードと後半ランニングスピードの変化比、完走割合を副次的評価項目（secondary endpoint）とする。そして、マラソンレース後の心身のコンディショニング状態も副次的評価項目（secondary endpoint）とする。

4) 評価項目（endpoint）の設定

主要評価項目（primary endpoint）は以下に示すように定義する。

「記録の達成指数」（記録の達成指数＝マラソン・ゴール記録／マラソン・自己記録（過去3年間）×100

副次的評価項目（secondary endpoint）は以下に示すように定義する。

① 「ランニングスピードの変化比（低下比）」

計測区間 5 区間 (0-10km、10-20km、20-30km、30-40km、40-42.195km) の各区間の平均ランニング速度 (m/min) における最高平均速度と最低平均速度の比 (%) とし、ランニングスピードの変化比 = $100 - (\text{最低平均速度} / \text{最高平均速度} \times 100)$ と定義

② 「前半・後半ランニングスピードの変化比」

計測区間 2 区間 (前半0-20km、後半20km-42.195km) の前半平均ランニング速度 (m/min) と後半平均ランニング速度 (m/min) の変化比とし、前半・後半ランニングスピードの変化比 = $\text{前半平均ランニング速度} / \text{後半平均ランニング速度} \times 100$ と定義

③ マラソンの完走割合 (%)

各グループでマラソンを完走した対象者の割合 (%) と定義

なお、追加の評価項目として、マラソン後の心身のコンディショニング状態に関し、マラソン終了後に、心身のコンディショニング状態を知る質問紙調査を行う。

5) 研究対象者数設定の根拠

本研究の主たる研究仮説は、「マラソン 4 週前の30km走は有効であること」ことである。マラソン 4 週前の30km走について、科学的に検証した研究はこれまで見あたらない。本研究の主要評価項目 (primary endpoint) である記録の達成指数は、2017年のパイロット研究の結果¹¹⁾をもとに試算したところ、 5.9 ± 12.8 (SD) となった。この試算をもとに、帰無仮説「マラソン 4 週前の30km走は有効でない」を有意水準 5%、検出力80%で棄却するための必要サイズは、片群74例となる。脱落20%を見込み、片群93例、両群で196例必要となる¹²⁾。

$$N = 2 \times (1.96 + 0.84)^2 / (5.9 / 12.8)^2 / 0.8 = 92.3$$

これから必要とされる研究対象者は、強介入群、弱介入群ともに100名程度となる。本研究では、研究対象者に2ヶ月間トレーニングを継続させ、マラソンの結果で検証するため、予想よりも高い脱落が生じる可能性は否定できない。研究対象者に関する研究実施に関する不利益は小さいことから、強介入群、弱介入群あわせて研究対象者を200名以上と設定する。

6) 研究対象者登録見込み

共同研究者1名がマラソン4週前に30km走をペースメーカー付き、給水サポートありの有料ランニングイベント(2,000円)を実施する。このランニングイベントと絡めて、研究対象者を募集する。3つのマラソン参加予定者の中で、3時間30分~5時間のレベルで、自己記録更新を目指し、30km走に参加できるランナーが研究対象者となる。また、研究代表者が継続的に指導しているランニングクラブの会員からも募集を行う。研究対象者の募集はマラソンレース2ヶ月前で締め切り、研究対象者に対してインフォームド・コンセントを行い、研究への同意を得る。

7) 研究参加に伴って予想される利益と不利益の要約

①予想される利益

本研究の研究対象者の利益として、予想されることは以下の事項である。

- ・ マラソン自己記録更新に向けて8週間のトレーニングスケジュール提供（無料）
- ・ コーチからのアドバイスを提供（無料）
- ・ 30km走または10km走（ペースセッターと給水が準備されている）を提供（無料）
- ・ 4回の合同ランニング練習会を提供（無料）

②予想される危険と不利益

本研究の研究対象者は、マラソンへの参加が認められた健康なランナーである。提供されるトレーニングスケジュールも、無理のない適切なものであり、体調が悪い場合には中止ができる。日常の練習は、各対象者の責任において実施し、各対象者が適切なスポーツ保険あるいは医療保険に加入するものとする。4回実施する合同練習会並びに30km走においては、研究対象者全員にマラソン保険を掛け、不慮の事態にも備える。そして給水対策、救護体制を準備し十分に安全に配慮し実施する。トレーニングコース上には、多くのサポート要員を配置し、体調が悪くなった場合には中止し休養できる状態にする。

3. 研究対象者の登録・割り付け

1) 研究対象者の募集・登録

研究対象者の募集については、以下の方針で実施する。

研究代表者が継続的に指導しているランニングクラブの会員やランニング学会会員を中心として研究対象者を募集する。また、共同研究者が実施するランニングイベント（30km走）と絡めて、研究対象者を募集する。そのほかにFacebook、ランニング雑誌なども通じて、研究についての情報を流し、研究対象者を募集する。

応募の受付はHPを通じて行い、参加を希望する者に対して、研究についての詳細な説明とトレーニング内容の説明を封書で送る。また、研究参加同意書も同封し、研究参加に対して十分な理解を求め、同意書を記載の上、同封の封筒で郵送してもらう。同意書を取得できた参加希望者を、研究対象者として登録する。

2) 研究対象者の割り付け

共同研究所の1名である統計担当者は、研究代表者が管理するデータセンターより、匿名化された登録後の研究対象者のデータを受領する。統計担当者は、統計処理の公平性や信頼性を確保するため、統計処理以外の作業には携わらない。研究対象者の割り付けは、マラソン大会、記録レベルを層としてランダムに二群に割り付ける。二群は30km走実施群（強介入群）と、通常のトレーニングである10km走を実施する群（弱介入群）である。調整因子は、大会（大阪マラソン、京都マラソン、泉州国際マラソン）、およびマラソンの自己記録である。記録に関しては～3時間40分、3時間40分～4時間00分、4時間

00分～4時間30分、4時間30分以上4水準とする。割り付け後統計担当者は割り付け結果を、速やかに研究代表者に送付する。

4. 効果判定と統計的事項

1) 解析集団の設定

本研究は初めに割り付けた集団を基に解析する (Intention to treat) ため、解析集団は最大解析対象集団 (Full analysis set : FAS) とする。また真に強介入、弱介入を完了した適合象集団 (Per protocol set : PPS) での比較も実施する。

2) 評価項目の解析

各評価項目について、「強介入群」「弱介入群」の2群のそれぞれの平均値を \bar{X}_1 , \bar{X}_2 とするとき、 $\bar{X}_1 - \bar{X}_2$ の95%信頼区間を推定する。このとき差 = 0 を意味する「0がこの信頼区間に含まれない」とき、「(有意水準5%で) 有意に差がある」と結論する。

主要評価項目は「記録の達成指数」とする。副次的評価項目はランニングスピードの変化比、前半ランニングスピードと後半ランニングスピードの変化比、マラソンレースの完走割合とする。

3) 探索的解析

主要評価項目、副次的評価項目の他に、解析時まで積み上げた知見を元に、その他の解析を実施する。

4) 統計担当者は下記の事項を実施する

- ・対象者の2群へのランダム割り付け
- ・解析計画案、解析報告書の作成
- ・統計解析の実施

5) データセンターは下記の事項を実施する

- ・データセンターは、取得したデータを匿名化し、対応表を保管する。
- ・データセンターは、匿名化したデータをパスワードかけて機密保護を施し CD-ROM 1枚のみに焼き、簡易書留で統計担当者に送付する (受け取り記録を確認する)。
- ・データのバックアップは、データセンター内の PC に匿名化したデータをパスワードかけて機密保護を施し保存する。
- ・データセンターは研究終了後5年後、統計担当者からデータ消去 (CDの物理的破壊等) の確認文書を受け取り、また同時にデータセンター内で保管している本研究のデータを消去・削除する。
- ・データセンターは、鍵のかかる部屋で入出者を管理する。
- ・1台の PC で管理する。
- ・インターネット等を接続しないなど、外部とは遮断する。
- ・データが外部へ流失しないように十分に配慮する。

5. 倫理的事項

本研究は「人を対象とする医学系研究に関する倫理指針」(平成29年2月28日一部改正版：以下「指針」)の対象外ではあるが、人を研究対象とすることから、この指針に準拠して行う¹³⁾。指針上、本研究は、「(1)新たに試料・情報を取得して実施する研究」(軽微な侵襲を伴う研究)、統計担当者にデータの提供があるため、「(2)他の研究機関に既存資料・情報を提供する研究」、統計担当者側では、「(3)既存資料・情報の提供を受けて実施する研究」として整理される。

1) 研究対象者の保護

本研究に関係するすべての研究者は「ヘルシンキ宣言」の精神に則り、本研究を実施する。

2) インフォームド・コンセント

研究対象者への説明は、以下の手順で実施する。研究倫理委員会 (IRB : Institutional Review Board) から承認を受けた研究計画書に基づき、研究代表者から説明文書を研究対象者に郵送し、以下の内容を文書で詳しく説明し、対象者から同意を受ける (インフォームド・コンセント)。

【説明する内容】

- ① 研究の名称及び当該研究の実施について研究機関の長の許可
- ② 研究機関の名称及び研究代表者
- ③ 研究の目的及び意義
- ④ 研究の方法及び期間
- ⑤ 研究対象者として選定された理由
- ⑥ 研究対象者に生じる負担並びに予測されるリスク及び利益
- ⑦ 研究が実施又は継続されることに同意した場合であっても随時これを撤回できる旨
- ⑧ 研究が実施又は継続されることに同意しないこと又は同意を撤回することによって研究対象者等が不利益な取扱いを受けない旨
- ⑨ 研究に関する情報公開の方法
- ⑩ 研究対象者等の求めに応じて、他の研究対象者等の個人情報等の保護及び当該研究の独創性の確保に支障がない範囲内で研究計画書及び研究の方法に関する資料を入手又は閲覧できる旨並びにその入手又は閲覧の方法
- ⑪ 個人情報等の取扱い
- ⑫ 試料・情報の保管及び廃棄の方法
- ⑬ 研究の資金源等、研究機関の研究に係る利益相反及び個人の収益等、研究者等の研究に係る利益相反に関する状況
- ⑭ 研究対象者等及びその関係者からの相談等への対応
- ⑮ 研究対象者等に経済的負担又は謝礼がある場合には、その旨及びその内容

- ⑩ 侵襲を伴う研究の場合には、当該研究によって生じた健康被害に対する補償の有無及びその内容
- ⑪ 研究対象者から取得された情報について、研究対象者等から同意を受ける時点では特定されない将来の研究のために用いられる可能性又は他の研究機関に提供する可能性がある場合には、その旨と同意を受ける時点において想定される内容

3) 研究への参加と撤回について (オプトアウト)

研究代表者から、研究への参加希望者に研究内容、トレーニングの進め方に関する説明書、同意書を郵送する。研究参加について同意できる希望者は、同意書に署名し、同封の封筒で返送する。この手続きを持って研究対象者に登録する。内容に関して同意できない希望者は、同意書を返送しない。同意書の返送がない希望者は、研究参加を撤回されたとみなす。

6. 研究組織

1) 本研究の主たる研究グループと資金源

本研究の主たる研究資金は、研究代表者・山内武の大阪学院大学個人研究費である。また、本研究の中心となる介入である30km走、10km走は、共同研究者の1名が所属するミズノ・ランニング・ステーションの協力を得て実施するが、ミズノ・ランニング・ステーションは研究の結果に対して中立な立場であり、その結果に影響を与えない。また、ミズノ・ランニング・ステーションとの間に金銭の授受は行われぬ。本研究の研究グループは、ランニング科学、スポーツ科学、健康科学、医療統計、疫学の専門知識を有する9名の研究者で構成される。

2) 研究代表者：山内 武 大阪学院大学健康スポーツ科学

3) 研究事務局：大阪学院大学健康スポーツ科学・山内研究室

4) 研究倫理委員会

本研究は、研究グループの一員である九州共立大学の得居雅人の属する九州共立大学研究倫理委員会に倫理審査を要請し、同委員会で承認された（承認番号2017-08）。

7. 予定研究対象者登録数と研究期間

予定研究対象者数：200人

登録期間：2か月。追跡期間：登録終了後3か月。解析期間9か月。総研究期間：2年

8. 研究結果の発表

研究成果の外部への発信については、スポーツ科学関連学会（体育学会、ランニング学会）で論文・研究発表を行う。スポーツトレーニング関連の国際学会（NSCA等）でも成果を発表する。そして、マラソンをめざした一般のランナーに向けて、一般向けのラン

ニング雑誌において研究成果を紹介していく。また、研究方法、成果は、HPにおいても公開していく¹⁴⁾。

文 献

- 1) 澤田 亨：信頼できるトレーニング指導方法の見つけ方・作られ方、JATI EXPRESS、50：24-25、2015
- 2) 澤田 亨：疫学的研究手法の活用、体育の科学、65 No11：815-818、2015
- 3) 新谷 歩：今日から使える医療統計、医学書院：115-123、2015
- 4) 浜田知久馬：学会・論文発表のための統計学、真興交易・医書出版部：149-151、2013
- 5) Yasumura S, Hosoya M, Yamashita S, Kamiya K, Abe M, Akashi M, Kadoma K, and Ozasa K: Study protocol for the Fukushima health management survey, J Epidemiol; 22 (5) : 375-383, 2012
- 6) JCOG (日本臨床腫瘍グループ)：プロトコルマニュアル、
<http://www.jcog.jp/doctor/tool/manual.html> (2020年2月17日アクセス可能)
- 7) Kenneth S , Douglas A, David M, for the CONSORT Group: CONSORT 2010 Statement: Updated guideline for parallel group randomized trials, In 中山健夫、津谷喜一郎 編著：臨床研究と疫学研究のための国際ルール集 Part2、ライフサイエンス研究出版：42-48、2016
- 8) ランナーズ：30km走の効果、<https://runners.30k-series.com/> (2020年2月17日アクセス可能)
- 9) 鍋倉賢治：1時間走ればフルマラソンは完走できる、学習研究社：86-87、2007
- 10) 対馬栄輝：医療系研究論文の読み方・まとめ方、東京図書：59-60、2013
- 11) 山内 武、高橋秀人、串間敦郎、得居雅人、杉山喜一、足立哲司、中嶋南紀、前田和良、元根朋美、山内悠介：市民ランナーではマラソン4週前の30km走は効果的か？－ランダム化比較試験(RCT)を用いた検証－、ランニング学研究 Vol.29 No.1：82、2017
- 12) 加納克己、高橋秀人：医学統計学、南江堂：153-157、2012
- 13) 文部科学省、厚生労働省：人を対象とする医学系研究に関する倫理指針、
<https://www.lifescience.mext.go.jp/bioethics/ekigaku.html> (2020年2月17日アクセス可能)
- 14) 北大阪ランニングクラブ HP：2017マラソントレーニング実験、<https://www.kita-osaka-rc.com/2017%E3%83%9E%E3%83%A9%E3%82%BD%E3%83%B3%E3%83%88%E3%83%AC%E3%83%BC%E3%83%8B%E3%83%B3%E3%82%B0%E5%AE%9F%E9%A8%93/> (2020年2月17日アクセス可能)

2019年度 人文自然学会研究会
2019年6月10日(月) 午後3時30分～

2019年度前期の人文自然学会研究会が、2019年6月10日(月)午後3時30分より、木村正徳講師を司会者として2号館4階02教室において開催された。発表者と発表概要は以下のとおりである。

岩岡 眞弘

生徒指導から見た発達障がい

筆者は、高等学校に20年、特別支援学校に14年間勤務してきた。その間に様々な困難を抱える生徒たちと出会い、関わってきた。そして、そのような彼らを臨床心理、教育、生徒指導の面からどのように理解し、関わるべきかが筆者にとってのライフワークであった。今回は、教育現場で関わってきた様々な生徒の中で、近年、大きく取り上げられている発達障がい、自閉症との関わりについての経験を紹介することとした。

近年、発達障がいとその行動特性について、書籍やインターネットなど、様々なところで見聞きすることができる。しかし実際に、そのような人たちを目の前にして、どのように関わり、どのように配慮したら良いか、実践の場からの発信はまだあまり多くないように思う。

まず、自閉症を理解するにあたって、その発見そのものが、第2次世界大戦中の交戦国である国交のない国での同時発見という、不思議な偶然によるものであることを紹介した。そのような経緯で発見された自閉症は、その後も診断や理解について、常に変更されてきた。それは今も変わりがない。その捉え難く、定義も何も実は確立されていないことを出発点として、実践者として彼らをどのように理解し、関わるかについて、筆者の経験から述べてみた。

具体的には、自閉症である彼らは、私たちからは容易に考えられないような恐怖に満ちた世界を体験している事を報告した。様々なメディアなどで取り沙汰される、発達障がいや自閉症の問題行動の多くは、その恐怖に起因することが多い事を指摘し、筆者はその恐怖を理解することが、まず重要であることを提案した。彼らが恐怖に苛まれず、「安心」することが、問題行動の改善のために必要なのである。

その彼らが「安心」を経験する具体的な支援の方法として、筆者の体験からも、視覚を利用したコミュニケーションなどが有効であることを報告した。視覚支援などにより、彼らが周囲を理解し、「安心」を得ることにより、教師や支援者、周囲の人たちと安定した関係をつくることが可能となる。それが、発達障がいと呼ばれている彼らと共に生きやすくするために重要であることを訴えた。

2019年度 人文自然学会総会

2019年度の人文自然学会定期総会が、2020年2月25日(火)午後1時より、山下博志准教授を議長として開催され、以下の案件について審議、報告があった。

(1) 2019年度事業報告

研究発表会：2019年6月10日(月)

発表者 岩岡 真弘

テーマ 「生徒指導から見た発達障がい」

司会者 木村正徳

定期総会：2020年2月25日(火)

出版事業：『大阪学院大学人文自然論叢』(第79-80号) 合併号

(2) 2020年度事業計画

研究発表会、定期総会

出版事業『大阪学院大学人文自然論叢』(第81号)・(第82号)

(3) 新役員の承認

庶務委員：岩岡 真弘 講師 (経済学部)

編集委員：野口 隆 准教授 (経済学部) (任期1年)

編集委員：山口 雅之 教授 (経営実務科)

(4) 新入会員の報告

山口 雅之 (経営実務科)

(5) 会則の変更

会長および庶務委員より会則の改正案について説明があり、審議の結果承認された。

| 改正後 | 改正前 |
|---|---|
| <p>第7条 本会には次の役員をおく。任期は原則2年とする。<u>ただし、連続2期を超えて選出されてはならない。</u></p> <p>1. 会長 1名 2. 副会長 1名 3. 庶務委員 2名 4. 編集委員 2名</p> <p>第8条 会長は会員（教授に限る）の中から投票により選出し、総長がこれを委嘱する。 副会長および委員は、会長が会員の中から委嘱する。 会長の選出において、得票数が同数の場合は、年長者を選出する。</p> <p>附 則 <u>2020年4月1日改正。</u></p> | <p>第7条 本会には次の役員をおく。任期は原則2年とする。</p> <p>1. 会長 1名 2. 副会長 1名 3. 庶務委員 2名 4. 編集委員 2名</p> <p>第8条 会長は会員（教授に限る）の中から投票により選出し、総長がこれを委嘱する。<u>ただし、連続2期を超えて選出されてはならない。</u> 副会長および委員は、会長が会員の中から委嘱する。<u>任期は会長に準ずる。</u> 会長の選出において、得票数が同数の場合は、年長者を選出する。</p> |

淡 誠一郎

デジタル画像におけるフーリエ記述子の回転不変性の検証

物体の識別やクラスタリングにおいて、シルエットはもっとも重要な特徴の一つであり、多くの場合にスケールや回転に対する不変性が求められる。フーリエ記述子は、シルエットを構成する点列の座標、あるいは点列の偏角を周期関数とみなしてフーリエ級数展開することによって、スケールや回転に対して不変な形状記述を得ようとする手法であり、いくつかの形式化のバリエーションが提案されてきた。それらは動植物の形態学的解析、病理診断、工業製品のデザインなどに利用されている。

理論的には、フーリエ記述子がスケール不変性や回転不変性をもつことに疑いの余地はない。また十分短い標準化間隔と十分な量子化レベルをもって離散化されているならば、デジタル画像（ここではラスタ表現を前提としている）であっても、近似的にそれらの性質を持つであろうことも予想できる。しかし逆に言えば、ラスタ表現されたデジタル画像に対しては、フーリエ記述子の回転不変性は多少なりとも崩れるということも意味する。

本研究は実際の画像をもちいて、フーリエ記述の回転不変性がどの程度成り立っているのかを調査するものである。結果として、デジタル画像では回転不変性が大きく崩れることが分かった。

岩出 秀平

組込 CPU における割込・分岐処理の高速化に関する研究

本研究はリアルタイムシステムにおける割込処理に対応する組込用 CPU の高速化に関するものである。高速化の狙いは割込処理時に必要となるレジスタ等の退避や復帰処理に係る命令を削減できる新奇回路の開発である。そのため割込処理プログラムからの戻り先アドレスを保存する BPC (Backup Program Counter) レジスタを複数個保存可能な BPC ファイルを開発し、レジスタの集合体であるレジスタ・ファイルを 1 バンクとして 9 バンクに多バンク化し、これらのハードウェアをポインタで制御可能にした。そして割込や復帰に対応してポインタを発生するファイル・ポインタを開発した。この結果、割込や復帰時の BPC やレジスタの退避・復帰のための命令数を 0 に削減することに成功した。また分岐前プログラムと分岐後プログラムとの間でデータをやり取りする機能を付加するため、このポインタを利用したグローバル・レジスタも考案した。以上のハードウェアを搭載した新方式 CPU に対して HDL シミュレーションを実施し、全ての命令の正常動作を確認した。しかし、命令数を削減できても新奇回路により回路遅延が増大すると高速化の効果は相殺されるので、新奇回路による回路遅延を増大させないことが前提条件となる。そこで新奇回路による回路規模や回路遅延を評価するため、QuartusII で新方式 CPU と旧方式 CPU の論理合成をおこない両回路を比較した。その結果、BPC のファイル化やレジスタ・ファイルのバンク化に伴い増加した総レジスタ数は旧方式 CPU に比べて 87% の増加に留まり、組合せ回路も含めた CPU 全体の回路増加分は 49% であった。また回路遅延時間を律速する回路パスは新奇回路ではなく新方式 CPU にも旧方式 CPU にも搭載されているデータメモリであることが判明し遅延時間の差は 0.04% でほとんど変化がなかった。従って、新奇回路導入による回路遅延の増大はなく、且つ回路規模の増加を 50% 以内に留め、割込・分岐処理における退避・復帰命令を 0 に削減できたことから、新方式 CPU が割込・分岐処理を多用するリアルタイムシステムに貢献できることが分かった。

淡 誠一郎

根菜の画像解析のための形状記述についての研究ノート —フーリエ記述子による形状記述—

果物や野菜の形状は収穫・加工の効率に大きくかわる。形状は生育環境や土壌にも左右されるが、品種独自の形質はやはり遺伝的要因に依るところが大きいと考えられる。例えば、トマトは複数の形質遺伝子の組み合わせでその果実の形が決まることがわかっている。ダイコンなどの根菜も同様に遺伝子が根形にかかわると予想されており、その解明が期待される。この目的のため、筆者を含む研究グループでは、ダイコンの根形を特徴づける複数の形質とその量的な測定方法に取り組んできた。

遺伝子と根形の関係を解明するには、まず現実の個体の形状を的確に、また解析に適した方法で記述する必要がある。そのような形状記述の候補は複数考えられるが、本稿では、動植物の形態学的解析、病理診断、工業製品のデザインなどに幅広く応用例の見られるフーリエ記述子を取り上げ、さまざまな品種さまざまな形状のダイコンについて根形の記述を試みた。その結果、複数種のフーリエ記述子のいずれを用いても、12～15次程度までの記述子によって個体の形状はかなり詳細に記述でき、大まかな品種識別には6～7次程度、詳細な識別でも高々10次程度までで可能であろうと推察されることが分かった。

山内 武・高橋 秀人・得居 雅人・足立 哲司・杉山 喜一
串間 敦郎・中嶋 南紀・前田 和良・元根 朋美・松尾信之介

市民ランナーに対するマラソン前の30km走の効果検証のための 研究プロトコール ーランダム化比較試験デザインを用いた研究ー

ランダム化比較試験（RCT）に関して、臨床医学研究分野では事前登録制が実施されており、疫学分野でも研究プロトコール論文を掲載するなど行われている。スポーツ科学の分野においても、RCTを用いた研究を実施する場合には、研究プロトコールを事前に公開が求められると考え、われわれは、そのプロトコールを論文として公開することとした。

本研究は、市民マラソン関係者（コーチ、市民ランナー等）において「効果がある」と考えられている「マラソンレース前の30km走の効果」について、十分な効果があるのかどうかを、エビデンスレベルの高いRCTデザインで検証することを目的とし、本稿ではその研究プロトコールをプロトコール論文として報告する。なお、このプロトコールは、JCOG（日本臨床腫瘍グループ）のプロトコールマニュアルを参考に作成している。本研究では、対象者を3時間30分～5時間程度の自己記録を持ち、自己記録更新をめざす一般市民ランナーとする。記録を更新するためにトレーニングしている対象者への道義的倫理性に配慮し、トレーニング非実施群を設けず、トレーニングに関する強介入群（マラソン4週前の30km走）と弱介入群（マラソン4週前の10km走）の2群について比較する。比較として、主要評価項目（Primary endpoint）「ゴール記録の達成指数」と4つの副次的評価項目（Secondary endpoints）「ランニングスピードの変化比」、「前半ランニングスピードと後半ランニングスピードの変化比」、「完走割合」、「マラソン後の心身のコンディショニング状態」として、RCTデザインを設定した。なお、本研究は疫学・統計モデルを基盤としている。

大阪学院大学人文自然学会会則

- 第1条 本会は「大阪学院大学人文自然学会」と称する。
- 第2条 本会の事務所は大阪学院大学図書館内におく。
- 第3条 本会は本学の設立の趣旨にもとづいて人文および自然科学の発展に寄与し、各分野にわたる会員相互の学術研究の交流を目的とする。
- 第4条 本会は次の事業を行う。
1. 機関誌『大阪学院大学人文自然論叢』の発行
 2. 研究会、講演会および討論会の開催
 3. その他、本会の目的を達成するために必要な事業
- 第5条 本会の会員は、大阪学院大学・大阪学院大学短期大学部の専任教員で、人文もしくは自然科学を専門とする者および本会の趣旨に賛同し、役員会の承認を得た者とする。
- 第6条 会員は本会の機関誌に投稿し、かつ機関誌その他の本会の刊行物の配布を受けることができる。
- 第7条 本会には次の役員をおく。任期は原則2年とする。
1. 会 長 1名
 2. 副 会 長 1名
 3. 庶務委員 2名
 4. 編集委員 2名
- 第8条 会長は会員（教授に限る）の中から投票により選出し、総長がこれを委嘱する。ただし、連続2期を超えて選出されてはならない。
- 副会長および委員は、会長が会員の中から委嘱する。任期は会長に準ずる。
- 会長の選出において、得票数が同数の場合は、年長者を選出する。
- 第9条 会長は本会を代表し、会務を統轄する。副会長は会長を補佐する。
- 第10条 会長は役員会を招集して、その議長となる。
- 第11条 会長は会務執行に必要なとき、会員の中から実行委員を委嘱することがある。
- 第12条 総会は年1回これを開く。ただし、必要あるときは、会長が臨時に招集することができる。
- 第13条 本会の経費は大阪学院大学からの交付金のほかに、有志からの寄付金、その他の収入をもってあてる。
- 第14条 各学会の相互の連絡調整をはかるため「大阪学院大学学会連合」をおく。
- 本連合に関する規程は別に定める。
- 第15条 会計は毎年4月1日に始まり、翌年3月31日に終る。
- 第16条 本会会則の改正は総会の議を経て総長の承認をうるものとする。

附 則

- この会則は、昭和49年10月1日から施行する。
- 昭和51年9月9日改正
- 昭和56年1月22日改正
- 昭和61年6月5日改正
- 昭和62年6月4日改正

平成15年6月12日改正

平成23年4月18日改正

平成25年4月1日改正

平成30年4月1日改正（ただし、第7条の編集委員の人数は、平成31年3月31日まで4名とする）。

以 上

大阪学院大学人文自然論叢投稿規程

1. 投稿論文は、未発表のものに限る。
2. 投稿論文は、編集委員会が依頼した査読者の査読を経た後、編集委員会で掲載を決定する。
3. 投稿資格
 - イ. 投稿者は、原則として本会の会員に限る。
 - ロ. 会員外の投稿は役員会の承認を必要とする。
4. 提出原稿の分量の限度は、原則として本誌刷り上がりページ数12ページとする。

これは和文原稿の場合、200字詰原稿用紙80枚（16,000字）に相当し、欧文原稿の場合、A 4判・12ポイント・ダブルスペースで20枚に相当する。（提出原稿がデータの場合は、Wordファイル形式とする。または、その他のファイル形式の場合テキストファイル形式とする。なお、提出原稿は、正本とコピーを2部提出することとする。）

ただし、制限分量を超えるものについては編集委員会の承認を必要とする。
5. 本誌はB 5判とする。発行は、前期・後期の2回とし、年間総ページ数は220ページを基準とする。
6. 原稿は、論説、研究ノート、資料、その他（レビュー、紹介、書評、学内消息など）に区分し、この順序にしたがって編集する。原稿の掲載順序については、編集委員会で検討のうえ決定する。
7. 別刷（抜刷）40部 無償とする。
8. 要旨集発行のため、要旨を添えること。（200字詰原稿用紙3～4枚。邦文ヨコ書き）

なお、この要旨は独立行政法人 科学技術振興機構（JST）の「科学技術文献データベース」にも掲載される。ただし、和文1,000字、英文2,000字を超える場合は（...）と表示される。
9. 投稿され掲載された成果物の著作権は、著作者が保持する。

なお、出版権、頒布権は大学が保持するため、論文転載を希望する場合は、学会宛に転載許可願を提出願うこととする。
10. 投稿された論文の著作者は、当該論文を電子化により公開することについて、複製権および公衆送信権を大学に許諾したものとみなす。大学が、複製権および公衆送信権を第三者に委託した場合も同様とする。

この規程は、平成26年4月1日から適用する。

以 上

大阪学院大学人文自然論叢執筆要領

原稿について

1. 原稿は確定稿であることを原則とする。校正段階での加筆訂正は極力さけること。
2. 原稿は正本を提出し、かならず原稿提出票をつけること。原稿提出票は学会連合事務局（図書館内）にある。
3. 原稿は黒または青のペン書き、またはワープロ原稿とすること。（欧文原稿はタイプライト）。
4. 原則として目次は使用しないこと。
5. 原則として古典・地名または学術用語などの特殊な例を除き現代かなづかい、および当用漢字を使用し、難字は欄外に朱筆大書すること。
6. 原稿中の数字は原則としてアラビア数字を使用すること。
7. 和文原稿中の欧語はタイプライトまたは活字体とすること。

注と文献表について

8. 注は原則として脚注とする。その他の場合は、注の付け方を明示すること。
9. 原稿での注は一括して本文の文末にまとめること。印刷に際して脚注（あるいは指定した書式）となる。
10. 文献表を必要とする方は、原稿執筆に際して注の後に一括して書くこと。論文末に一括して掲載する。
11. 文献表の配列及び形式は各自が所属する学会で行われているものとする。

図・表について

12. 図・表は1表・1図ごとに別紙に書き、図表ごとに通し番号・図表名・説明文を記入して、本文原稿の右欄外に挿入箇所を指定すること。
13. 凸版印刷にかける図・表は原則としてトレーシングペーパーに墨入れし、厚手の台紙に貼付すること。
ただし、図表中に入れる文字は写植をするため鉛筆書きすること。
14. 写真は鮮明なものに限り、特別な場合を除いて白黒のものを使用し、取り扱い図・表に準ずること。
図・表・写真などのカラー印刷については、役員会の承認を必要とする。
15. 図・表・写真などはかならず縮尺または仕上り寸法を明示すること。

印刷・校正

16. 原則として、執筆者校正は2校までとし、2校までに校正の必要がなくなれば校了または責了と書くこと。
17. 執筆者校正は、校正刷到着の日より3日以内とする。
18. 執筆者校正の段階で加筆・訂正を加え、印刷組替えを生じた場合、組替料の実費を必要とする。また特殊な印刷によって通常の印刷費を大幅に上回る場合も同様とする。
19. 特別に指定される活字の大きさや字体については、執筆者が直接原稿に明示すること。それ以外の印刷上のスタイル、活字号数などは編集委員会において決定する。

以 上

執筆者紹介 (原稿受理順)

淡 誠一郎 (情報学部) 教授
山内 武 (経済学部) 教授
岩出 秀平 (情報学部) 教授

大阪学院大学人文自然学会会員 (2020年3月31日現在)

| | | | | | |
|-------|------|-------|-------|-------|--------|
| 秋田 亨 | 情報学部 | 岩岡 眞弘 | 経済学部 | 岩出 秀平 | 情報学部 |
| 上地 宏 | 商学部 | 梅田 享 | 外国語学部 | 大谷 朗 | 情報学部 |
| 川本 芳久 | 情報学部 | 菊野 亨 | 情報学部 | 木村 正徳 | 商学部 |
| 佐野 学 | 情報学部 | 杉山 伸一 | 商学部 | 角田 聡 | 経済学部 |
| 高田 裕文 | 商学部 | 竹井恵美子 | 国際学部 | 竹松 良明 | 経営実務科 |
| 田中 豊 | 情報学部 | 谷口 高士 | 情報学部 | 淡 誠一郎 | 情報学部 |
| 寺井 正幸 | 情報学部 | 寺脇 久人 | 経済学部 | 長澤 宣親 | 経済学部 |
| 西田 知博 | 情報学部 | 野口 隆 | 経済学部 | 廣藤千代子 | 経営実務科 |
| 藤原 哲郎 | 経営学部 | 藤原 秀雄 | 情報学部 | 松尾信之介 | 経済学部 |
| 松田 潤 | 情報学部 | 村上 孝三 | 情報学部 | 森井 紳 | 商学部 |
| 山内 武 | 経済学部 | 山口 雅之 | 経営実務科 | 山下 博志 | 情報学部 |
| 吉山 昭 | 情報学部 | 渡辺千香子 | 国際学部 | | (50音順) |

大阪学院大学人文自然学会役員

会 長 谷 口 高 士 (情報学部)
副 会 長 藤 原 哲 郎 (経営学部)
庶務委員 西 田 知 博 (情報学部) 吉 山 昭 (情報学部)
編集委員 淡 誠一郎 (情報学部) 渡 辺 千香子 (国際学部)

2020年3月25日 印刷
2020年3月31日 発行

編集兼 大阪学院大学
発行者 人文自然学会
〒564-8511 大阪府吹田市岸部南二丁目36番1号
TEL 06 (6381) 8434(代)

印刷所 大枝印刷株式会社
〒564-0031 大阪府吹田市元町28番7号
TEL 06 (6381) 3395(代)

**THE BULLETIN
OF
THE CULTURAL AND NATURAL SCIENCES
IN
OSAKA GAKUIN UNIVERSITY**

Nos. 79-80

March 2020

CONTENTS

Articles

Seiichiro Dan

A Study on Rotation Invariant Features of Fourier Descriptors on Digital Images 1

Shuhei Iwade

Study of High-speed Embedded CPU for EIT Processing 21

Note

Seiichiro Dan

A Study Note on Shape Descriptions for Image Analysis of Root Vegetables
– Fourier Descriptors – 55**Data**

Takeshi Yamauchi, Hideto Takahashi, Masato Tokui, Tetsuji Adachi, Kiichi Sugiyama

Atsuro Kushima, Nanki Nakashima, Kazuyoshi Maeda, Tomomi Motone, Shinnosuke Matsuo

Study protocol for the effect of the 30-kilometer training run before marathon race
in recreational runners – Randomized controlled trial 69**News** 81**Summaries** 84