コンパクトな文字発生方式について*

——ひらがなの発生——

鈴木隆一** 池田克夫** 清野　武**

Abstract

This paper discusses a technique for compact generation of "Hiragana", the Japanese cursive syllabaric charactesrs. "Hiragana" similar to those used in ordinary printing are generated, requiring 64 bits per symbol, by using five types of arcs and a frame for "Hiragana" which consists of 24 basic constructional points. These points are selected out of the biased distribution of characteristic points which are determined by dividing "Hiragana" into arcs.

"Hiragana" are generated by connecting the points with the arcs.

1. まえがき

最近，各種の図形出力装置が，新しい計算機の出力として脚光をあびている．これらの出力装置により，日常用いられている"ひらがな"を表示することが，今後，必要になるものと思われる．そこで，文字の品位に重点をおいたコンパクトな"ひらがな"発生について考えてみた．

"ひらがな"を発生する場合，文字を逐次線分で近似し，その線分列を文字データとして記憶しておく方式は，必要となるデータ量が増大するので，あまり有効でない．この理由は，"ひらがな"が基本的に曲線から構成されており，その曲線を近似した各線分を，おのおの文字データとして取り扱うためである．したがって，"ひらがな"の品位を保ったまま，文字データを減らすためにデータの圧縮を行なう必要がある．

この一方法として，"ひらがな"の"弧"による近似を行ない，文字データを圧縮することを検討した．

この方法を用いると，きわめて品位のよい"ひらがな"が，一文字あたり 64 ビットの文字データで発生できる．以下，"ひらがな"の構成法，"ひらがな"用枠の設定，"ひらがな"を構成する基本パターン，データ構造について述べる．

2. ひらがなの構成法

"ひらがな"を構成する方法として，まず，部分パターンを用いることが考えられる．これは，"は"と"ほ"にみられるような類似のパターンに着目し，幾種かの部分パターンの組合せで文字を構成する方法である．しかし，"ひらがな"の場合は，全体の個数に対する部分パターンの個数が比較的多く，しかも，おのおのを表現するためのデータ量が大きいので，あまり有効な方法ではない．そこで，以下の方法を考える．

"ひらがな"を，適当に選ばれた基礎構成点Nに属する点を，適当に選ばれた基本パターンPに属するパターンで，順次結び，構成する．

いま，"ひらがな" C_i を構成するのに必要となる点 $X_{ij}(j=1,2\cdots m_i)$ を構成点と呼び，各構成点を結ぶパターンを $R_{ij'}(j'=1,2,\cdots m_i-1)$ で表わすことにする．これらの記号を用いて，"ひらがな" C_i は，構成点 X_{i1} と構成点 X_{i2} をパターン R_{i1} で結び，次に，構成点 X_{i2} と構成点 X_{i3} をパターン R_{i2} で結ぶ……ことにより構成される．これを

$$C_i - X_{i1}R_{i1}X_{i2}R_{i2}X_{i3}\cdots X_{i,m_i-1}R_{i,m_i-1}X_{i,m_i} \quad (1)$$

$$X_{ij} \subset N \ (j=1,2,\cdots,m_i)$$
$$R_{ij'} \subset P \ (j'=1,2,\cdots,m_i-1)$$

で表わす．

すべての"ひらがな"を構成する場合，基礎構成点 N と基本パターン P の選び方が問題となる．ここでは，まず，"ひらがな"のもつ"特徴的パターン"を用いて"ひらがな"を分割し，分割点の分布から，一定の制限内に各文字データがおさまるように，基礎構成点と基本パターンを選ぶことにする．

---

* A Technique for Compact Generation of Symbols, ——Generation of "Hiragana"——, by Ryuichi SUZUKI, Katsuo IKEDA and Takeshi KIYONO (Faculty of Engineering, KYOTO University)

** 京都大学工学部

3. ひらがなの分割

“ひらがな”を構成している“特徴的パターン”は“弧”である．“ひらがな”に含まれる曲線を近似するとき，どのような“弧”を用いるのが最も適しているかを考えるために，“ひらがな”に含まれる“独立な弧”，たとえば，“あ”であれば“⌒”と“(”，だけを取り出してみると，全体的な傾向として，長い“弧”ほど曲率は小さく，短い“弧”ほど曲率は大きい．さらに，“弧”全体について，“振れ”がほぼ一定である．

このような“独立な弧”のもつ傾向に従って，こんどは逆に“ひらがな”に含まれる曲線部を適当に分割してみる．“お”を，“弧の振れ”がほぼ一定になるように分割すると **Fig. 1** のようになる．

このようにして，すべての“ひらがな”を分割してみると，分割点の分布に，ある程度の“偏り”が認められ，**Fig. 2** に示したように 25 点が格子状に正しく並んだ形になる．これらの点を用いて，逆に，すべての“ひらがな”を良好に近似することができる．

4. 符号割当ての方針

(1)式の記号が，おのおのに割り当てられた符号を表わすものとすると，文字データとして必要となる全ビット数は

$$\sum_{j=1}^{m_i-1}(\lceil \log_2 X_{ij} \rceil + \lceil \log_2 R_{ij} \rceil) + \lceil \log_2 X_{im_i} \rceil \quad (2)$$

$\lceil \alpha \rceil$ は，α 以上の，最小の整数を示す．

となる．

したがって，実際の符号割当てにおいては，対象とする文字データの最長必要ビット数が，許容ビット数以下におさまるように工夫を行なわなければならない．さらに，このために生じるデータの処理が，あまり複雑にならないように留意しなければならない．

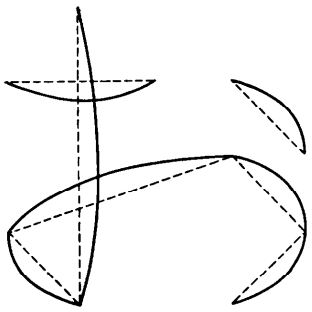


Fig. 1 Division of “お” into arcs

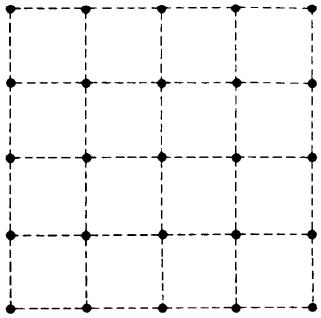


Fig. 2 The distribution of the characteristic points

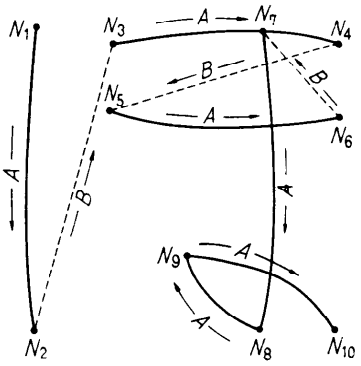


Fig. 3 The construction of “ほ”

ここでは，出現確率の最も高いパターンに対して，パターン指定を行なわないことにし，さらに，最長の文字データを必要とする文字に対して，なるべくパターン指定用の符号をはぶく方針をとる．このために，基礎構成点を示す符号と基本パターンを示す符号が，一意的に判別されるように割り当てる．

具体的に，“ひらがな”の場合を考え，文字データを 64 ビットにおさめることにする．例として“ほ”を考えると，文字データは，“弧”を A，“空線”を B として，**Fig. 3** に示されるように

$$N_1AN_2BN_3AN_4BN_5AN_6BN_7AN_8AN_9AN_{10} \quad (3)$$

となる．

いま，おのおのに等長符号を割り当てると 5 ビットの符号が必要となり，文字データは 95 ビットとなる．

ここで，出現確率の多い“弧”に，“空符号”を割り当てると，文字データは 65 ビットとなるが，まだ，許容ビット数 64 ビットにはおさまらない．

したがって，不等長符号を点指定とパターン指定に

割り当てる．**3.** の結果から，発生される“ひらがな”の品位を保つためには，基礎構成点は，20 数点必要であるので5ビットの符号を点指定に割り当て，不等長の符号処理を容易にするために，24 点とする．このように点指定符号を選ぶと，許容ビット数の制限から，パターン指定符号は最大4ビットとなる．さらに，不等長の符号処理のために，上位2ビットを使用するので，基本パターンとして，最大5パターンが指定可能である．

5. ひらがな用枠

3. で求めた分割点の分布を基にして，**4.** の制限のもとで選んだ基礎構成点によって，ひらがな用枠を構成する．分割点の分布として与えられている 25 点から適当な一点を除けばよいが，品位を保つために，全体に寄与する割合の最も少ない点を消去することが望ましい．実際に，各点の使かわれる割合を調べ，**Fig. 4** に示される 24 点をひらがな用枠とした．消去した一点を用いる文字は，“む”，“り”，“を”であり，これらの文字の形は若干悪くなるが，品位は十分に保たれる．

なお，Fig. 4 に示された値を，ひらがな用枠に属する点を示す符号とする．

6. 基本パターンの選択

ひらがなの“特徴的パターン”として，“弧”を考えた．一方，符号割り当ての方法から，パターンとして，5個までは指定できることを前に述べた．本章においては，文字の表現とパターンの処理を考慮して，基本パターンを選択する．

まず，離れた点に至るパターンとして，“空線”は無条件に必要となる．

次に，“い”に見られるように，“ノ”で示される“はね線”も，“ひらがな”に多く見られる．これをも，“弧”として表現することは可能であるが，品位および必要となる文字データを考えると，“はね線”という一つのパターンとして取り扱ったほうが有利である．

“特徴的パターン”として取り扱った“弧”も，実際の処理のためには，時計回り，反時計回りの2種が必要である．実際の筆順を調べてみると，時計回りの場合が大部分であり，反時計回りの場合は，筆順を逆にすれば時計回りとみなすことができる．しかし，その筆順変更のために，必要となる文字データは増大し，64 ビットの制限を越えることがある．したがって，“弧”として，時計回りと反時計回りの2種類のパターンが必要となる．

さらに，曲率の大きな，長い曲線を“弧”を用いて近似する場合，曲率の小さい“弧”で細かく近似するよりも，曲率の大きな“弧”であらく近似したほうが，必要となる文字データの点からも，品位の点からも有利である．実際の“ひらがな”においては，このような曲線はほとんど時計回りの筆順であるので，大きた曲率の時計回りの“弧”を，一つのパターンとして採用する．

このようにして選んだ5個のパターンを，“ひらがな”を構成する基本パターンPとし，次の記号で示すことにする．

P_1：空　線（離れた点に移ることを示す）
P_2：はね線（途中までは実線，以後は空線）
P_3：弧Ⅰ　（低曲率，時計回り）
P_4：弧Ⅱ　（低曲率，反時計回り）
P_5：弧Ⅲ　（高曲率，時計回り）

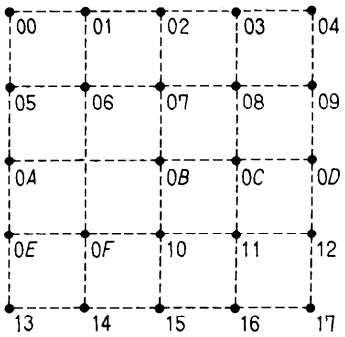
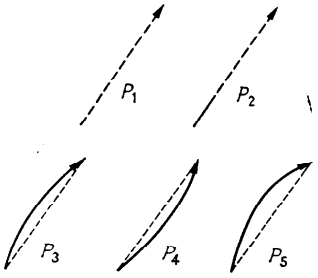


Fig. 4 The frame for “Hiragana”



P_1: blank line
P_2: sweeping line
P_3: clockwise arc of low curvature
P_4: counter-clockwise arc of low curvature
P_5: clockwise arc of high curvature

Fig. 5 Basic patterns forming “Hiragana”.

各パターンを **Fig. 5** に示す.

さらに，各パターンに次のように符号を割り当てる.

$(P_1, P_2, P_3, P_4, P_5)=((11)_2, (10)_2, \varepsilon, (01)_2, (00)_2)$

P_3 には，空符号 ε を割り当て，符号の列から次章に述べる方法で，パターンを指定する.

7. データ構造

ひらがな用枠の点を示す符号と，基本パターンを示す符号と区別するために

$k=(11)_2$,

なる前置符 (prefix code) を考え，パターンを示す符号の前に置く．この方法により，点を示す符号は（0 0＊＊＊)$_2$，（0 1＊＊＊)$_2$，（1 0＊＊＊)$_2$，パターンを示す符号は（1 1＊＊)$_2$となり，前置符 $k=(11)_2$ の有無で，直ちに符号の判別ができる．ひらがな用枠の点の個数を 24 点に制限したのは，前置符を用いて符号の判別を容易に行なうためである.

文字データの縮少をはかるために，“ひらがな”において最も出現率の多のパターン P_3（低曲率，時計回り）に，“空符号”を割り当てることを **6.** で述べた．また，パターン P_5（高曲率，時計回り）は，“の”の後半の部分等で用いるので，連続して現われることが多い．このことも考慮に入れて，文字データ作成におけるパターン指定に，次の規則を定める.

ⅰ）P_1, P_2, P_4 は，直後に現われる点のみを修飾する.

ⅱ）P_5 は，次に P_1, P_2, P_4 が現われるまで，連続して現われる点をすべて修飾し，後はⅰ)に従う.

ⅲ）ⅱ)以外における点の連続は，P_3 による修飾とする.

また，データの処理を有効に行なうために，データの終了を次の規則で示す.

ⅰ）点を示す符号以外で，データが始まるとき.

ⅱ）P_5 が，2度続いたとき(必ずしも連続でなくてよい).

ⅲ）データが 64 ビットを越えたとき.

ここで，ⅱ）の規則は，文字データとして考えられない符号列で，データの終了を示すものである．“ひらがな”では，P_5 のパターンで文字が終わることが多いので，データ終了の符号を最少にするために P_5 を用いた.

例として，**Fig. 6** に“は”の文字データをあげ，**Fig. 7** にデータ処理の流れ図を示しておく.

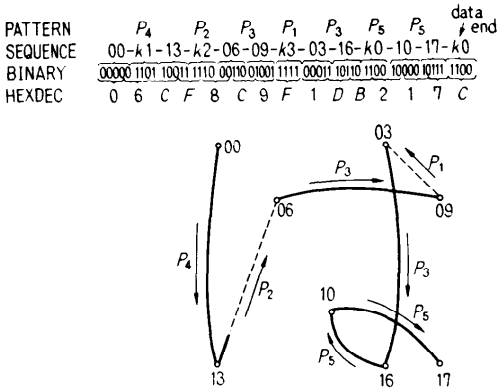


Fig. 6 Construction of “は”

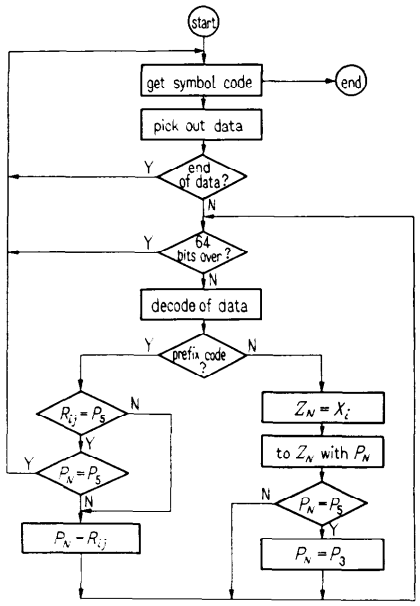


Fig. 7 Flow chart of data processing routine

8. 弧の処理

“弧の振れ”がほぼ一定という特徴を利用し，前もって与えられた振幅値と，与えられた方向から，任意の2点を結ぶ“弧”を作成する.

“弧”は適当な線分近似で表わす．近似の精度は，理論上制限はないが，実用的な面からは，なるべく少ない線分で近似することが望ましい．実際に“弧”を描き，それを適当な線分で近似してみると，四線分程度の折線近似で，十分に近似できることがわかる.

いま，(X_c, Y_c) から (X_N, Y_N) に至る弧を考える.

処理を簡単にするために，(X_c, Y_c) と (X_N, Y_N) を結ぶ弦の四等分点 (x_i, y_i) $(i=1, 2, 3,$ 以下同じ$)$ を基準にして弧を近似する．

弧を近似するのに，四等分点 (x_i, y_i) より弦に垂線を引き，近似点 (X_i, Y_i) を求める方法をとると，割算，平方，平方根の処理が必要となり能率が悪いので，より簡単な処理による弧の近似法について考える．

$\therefore \Delta X = X_N - Y_c$, $\Delta Y = Y_N - Y_c$, 振幅値（弦と弧との距離）を Δ_i として

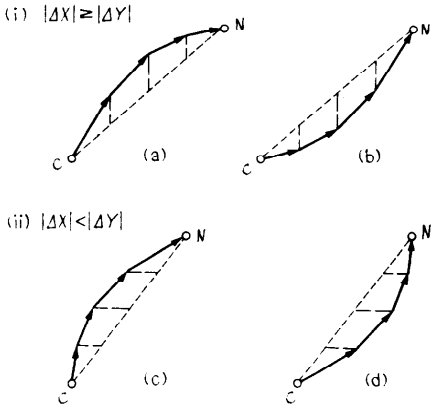
ⅰ）$|\Delta X| \geq |\Delta Y|$ のとき

$$(X_i, Y_i) = (x_i, y_i \pm \Delta_i \operatorname{sign} \Delta X).$$

ⅱ）$|\Delta X| < |\Delta Y|$ のとき

$$(X_i, Y_i) = (x_i \mp \Delta_i \operatorname{sign} \Delta Y, y_i).$$

ただし，$\operatorname{sign} \alpha = \begin{cases} 1 & \alpha \geq 0 \\ -1 & \alpha < 0 \end{cases}$



(i) arc of a gradient of less than 45 degrees, (a) clockwise, (b) counter-clockwise
(ii) arc of a gradient of more than 45 degrees (c) clockwise, (d) counter-clockwise

Fig. 8　Approximation of arcs

Fig. 9　Improving the quality of "Hiragana" through curvature of arcs

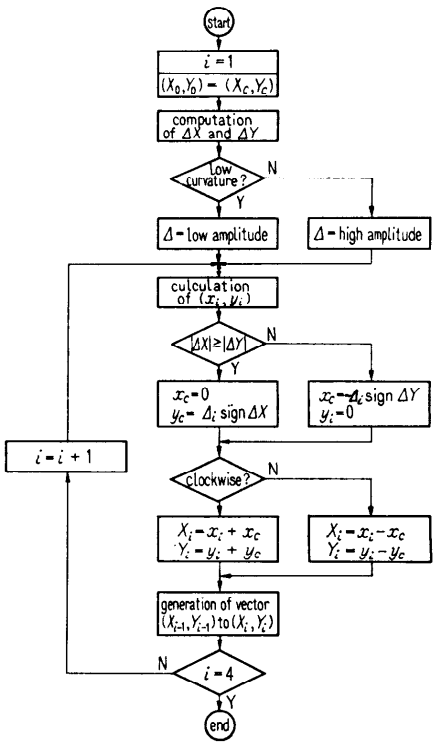


Fig. 10　Flow chart of arc processing routine

$$\text{複号} \begin{cases} \text{時計回り} \\ \text{反時計回り} \end{cases}$$

とすれば，加減算を1回行なうことにより，近似点 (X_i, Y_i) を求めることができ，しかも，**Fig. 8** に示されるように，十分，弧のもつ特徴を表わしている．

このようにして弧を発生した場合，その振幅値（Δ_1, Δ_2, Δ_3）の，単位長 U（隣り合う点間の最小距離）に対する割合をどのようにとるかが問題となる．実際に発生されたひらがな（**Fig. 9** 参照）を，品位に関して比較した結果，次の値を得た．

ⅰ）低曲率 $(U : \Delta_1 : \Delta_2 : \Delta_3) = (12 : 2.5 : 3 : 2.5)$

ⅱ）高曲率 $(U : \Delta_1 : \Delta_2 : \Delta_3) = (12 : 4 : 5 : 4)$

また，"はね線" の "実線" 対 "空線" の割合は，処理を簡単にするため1対3とした．

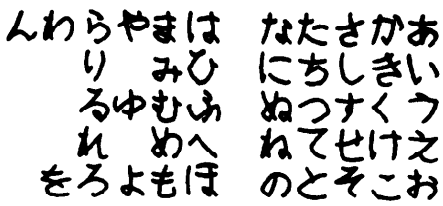


Fig. 11　Generated "Hiragana"

Table 1 Data for "Hiragana"

	DRAWING SEQUENCE	DATA
あ	09-05-k3-15-02-k3-0C-14-0E-0C-12-16-k0-k0	497E A2F6 51CC 95B3
い	00-k1-14-k2-08-12-k0-k0	06D3 912C C000
う	01-08-k3-05-09-k0-15-k0	0A3C A9CA E600
え	01-07-k3-05-k1-09-13-k3-0F-15-k1-17-k0	09FC BA99 FBEB B7CC
お	07-05-k3-01-14-0E-0C-12-16-k3-08-0D-k0	397C 3473 256F 4373
か	01-13-k3-05-08-15-k2-05-k3-03-k0-0D-k0	0CFC A8AF 17C7 8DCC
き	03-05-k3-09-0A-k3-16-14-0F-12-02-k0-k0	197D 2AFB 51F2 1660
く	02-0A-15-k0-k0	12AB 9800
け	00-k1-13-k2-05-09-k3-03-16-k0-k0	06CF 8C9F 1DB3
こ	06-k1-08-k2-0E-k1-14-k1-17-k0-k0	36A3 9DB4 DBE6
さ	09-0A-k3-16-14-0F-11-02-k0-k0	4ABE D47C 4598
し	12-15-0F-01-k0-k0	955E 1CC0
す	09-05-k3-02-10-k0-0B-14-k0	497C 50C5 D330
せ	09-0A-k3-03-0C-k3-17-14-01-k0-k0	4ABC 6CFB D039 8000
そ	03-01-k3-16-10-09-0A-k1-03-k0-k0	187E D04A B179 8000
た	07-05-k3-01-13-k3-0B-0D-k3-17-k0-0F-k0	397C 33F5 B7EF 8FCC
ち	09-05-k3-02-0F-0C-12-16-14-k0-k0	497C 4F64 AD4C C000
つ	0A-08-0D-12-15-k0-k0	521B 2AE6
て	17-10-0B-04-00-k0-k0	BC16 4066
と	01-0B-k3-17-14-0E-k0-09-k0	0AFE F476 2730
な	07-05-k3-01-13-k3-08-0D-k1-0C-16-10-17-k0-k0	397C 33F4 3759 685F
に	00-k1-13-k2-06-09-k3-17-k0-0F-k0	06CF 8C9F BE3F 3000
ぬ	15-00-k3-02-14-0E-k0-08-0D-16-11-17-k0	A83C 5476 21B6 8DF0
ね	14-01-k3-13-06-05-k3-13-k0-08-16-11-17-k0	A07E 662F CF11 68DF
の	07-14-0E-k0-07-0D-16-k0	3D1D 876D B300
は	00-k1-13-k2-06-09-k3-03-16-k0-10-17-k0	06CF 8C9F 1DB? 17C0
ひ	0D-08-11-15-14-k0-0A-02-00-k0	6A23 5A62 840C
ふ	02-08-07-11-15-k3-14-0D-17-k3-14-k0-0A-k0	120F 1AFD 1B7F A62B
へ	17-06-0A-k0-k0	B395 9800
ほ	13-00-k3-01-04-k3-09-06-k3-03-16-10-17-k0	983C 24F4 9BC7 685F
ま	05-08-k3-0C-0A-k3-02-15-k0-0E-16-k0	2A3D 8AF1 571D 7C00
み	05-k1-07-14-0E-12-k3-0C-16-k0-k0	2E9E 8E37 B2D9 8000
む	07-05-k3-01-0F-0A-0F-k3-0D-08-16-14-0F-k0	397C 2F53 FDA8 B51E
め	15-00-k3-02-14-0E-k0-08-0D-16-k0	A83C 5476 21B6 C000
も	08-05-k3-0C-0A-k3-11-16-14-02-k0-k0	417D 8AF8 DA82 CC00
や	03-08-01-k1-15-k3-0A-09-k0-0D-0B-k0	1A03 B5F5 271A BCC0
ゆ	0E-05-k3-02-15-k3-0E-k0-08-0D-0F-k0	717C 55F7 621F FC00
よ	09-07-k3-02-15-k0-0F-17-k0	49FC 55C7 DF00
ら	01-08-k2-06-0F-k0-0C-12-15-k0	0A38 CFC6 4AB8
り	01-k1-0F-k2-03-0C-15-k0-k0	0EBF 86CA E600
る	01-k1-03-0E-k0-0C-12-16-10-16-k0	0E8D D8C9 5A16 C000
れ	14-01-k3-17-16-08-13-06-05-k0-k0	A07E F644 CC5C
ろ	01-k1-03-0E-k0-0C-12-15-k0	0E8D D8C9 5700
わ	14-01-k3-13-06-05-k3-13-08-k0-0D-16-k0	A07E 662F CD18 DB60
を	08-05-k3-02-0A-11-k3-17-14-0F-0D-k0-k0	417C 4A8F DE8F 6E60
ん	02-13-10-16-k1-12-k0-k0	14E1 6D96 6000

なお，**Fig. 10** に弧処理のための手順を流れ図で示しておく．

さらに，**Fig. 11** に発生された"ひらがな"，**Table 1** に各文字の文字データを示す．

9. むすび

ひらがなを，各文字について，その形の美しさ，すなわち，品位を中心にデータを作成し，ひらがな用枠の選択，弧による近似および不等長符号の採用によって，一文字あたりに必要なデータを 64 ビットにおさめることができた．このように圧縮されたデータを処理して文字を発生するのも，"弧"の処理を工夫することにより，能率よく行なうことができた．その結果，個々の文字に関しては，ほぼ期待どおりの成果を得ることができた．

また，前に報告した ISO コードの文字発生[1]においても，曲線部を"弧"に用いて表現することにより，同一の制限内で，より品位のよい文字が発生できる．

さらに，このような方式による漢字の発生も，基本的には，本文および文献に述べた方法で，十分に可能であり，漢字の分類を工夫することにより，必要となるデータもかなり減らすことができるものと思われる．

本研究は，文部省科学研究費補助金の交付を受けている研究の一環として行なったものである．

参考文献

1) 鈴木，池田，清野：コンパクトな文字発生方式について，情報処理，Vol. 11, 1970, No. 9, pp. 517～522.

2) 鈴木，池田，清野：コンパクトな文字発生方式について—ひらがなの発生—，情報処理学会，昭和 45 年度第 11 回大会講演予稿集，pp. 345～346.

3) 池田，鈴木，清野：タブレット式図形入力装置と蓄積管表示装置を持つ図形入出力端末の構成と制御について，情報処理学会関西支部，プログラミング言語研究会，1970. 12. 23.

(昭和 46 年 2 月 8 日受付)