

卒業論文

# 印刷文字認識の並列処理に関する基礎研究

昭和63年3月25日  
東北大学工学部情報工学科  
木村研究室  
大町真一郎

|                                |    |
|--------------------------------|----|
| 第1章 序論                         | 1  |
| 1.1 本研究の目的                     | 1  |
| 1.2 シストリックアルゴリズム               | 1  |
| 1.3 文字認識アルゴリズム                 | 2  |
| 1.4 本論文の構成                     | 2  |
| 第2章 パターン整合法による文字認識を実行するセル構造回路網 | 3  |
| 2.1 パターン整合法によるアルゴリズム           | 3  |
| 2.2 配置面積重視のセル構造回路網             | 4  |
| 2.2.1 クラスの選出                   | 4  |
| 2.2.2 得点計算及びソート                | 5  |
| 2.2.3 処理時間の評価                  | 6  |
| 2.2.4 配置面積の評価                  | 7  |
| 2.3 処理速度重視のセル構造回路網             | 7  |
| 2.3.1 クラスの選出                   | 7  |
| 2.3.2 得点計算                     | 8  |
| 2.3.3 得点のソート及び結果の出力            | 8  |
| 2.3.4 処理時間の評価                  | 9  |
| 2.3.5 配置面積の評価                  | 9  |
| 2.4 折衷型のセル構造回路網                | 10 |
| 2.4.1 クラスの選出                   | 10 |
| 2.4.2 得点計算及びソート                | 10 |
| 2.4.3 処理時間の評価                  | 11 |
| 2.4.4 配置面積の評価                  | 11 |
| 2.5 逐次処理の処理時間の評価               | 12 |
| 2.6 各方法の比較                     | 13 |

|                               |    |
|-------------------------------|----|
| 第3章 重み付きグラフの問題を解くセル構造回路網      | 32 |
| 3.1 本章の目的                     | 32 |
| 3.2 重み付きグラフの問題                | 32 |
| 3.3 逐次処理に要する演算回数              | 34 |
| 3.4 重み付きグラフの問題を解くセル構造回路網      | 35 |
| 3.4.1 シストリックアルゴリズムを適用するための変更点 | 35 |
| 3.4.2 適合関数の計算                 | 36 |
| 3.4.3 正規化関数の計算                | 37 |
| 3.4.4 確率の修正                   | 37 |
| 3.4.5 処理時間の検討                 | 37 |
| 3.5 逐次処理と並列処理の演算回数の比較         | 39 |
| 第4章 結論                        | 46 |
| 謝辞                            | 47 |
| 参考文献                          | 48 |
| 発表論文リスト                       | 49 |
| 付録 候補選出に要する比較回数の検討            | 50 |

## 第1章 序論

### 1.1 本研究の目的

近年、情報処理技術の進歩により、文書の管理が計算機によって行なわれるようになってきた。それに伴い、既存の文書を計算機に効率よく入力する光学読み取り装置 (Optical Character Reader) の開発が期待されている。

ところで、漢字認識を行なう場合、問題点の一つに字種の多いことが挙げられる。漢字の総数は5万とも言われ、日常よく使われるJIS第一水準の漢字だけでも約3000字種存在する。従って、文字認識アルゴリズムは計算量が膨大になり、逐次処理で認識を行なったのでは非常に時間がかかって実用化には向かない。そこで、並列処理を行なうことが必要になる。

本研究の目的は、並列処理アルゴリズムの一種であるシストリックアルゴリズムを用いて、文字認識を高速に実行するセル構造回路網を構成することである。

### 1.2 シストリックアルゴリズム

並列処理アルゴリズムとして、本研究ではシストリックアルゴリズムを用いる。これは、同一の機能を持つ多数のセルで構成されるセルアレイ上で、セル間の局所的なデータの通信とセル内の演算のみで実行されるアルゴリズムである。従って、演算の並列性・規則性、通信の局所性、構造の一様性・拡張性に優れており、VLSI化に向いている<sup>[1]</sup>。

シストリックアルゴリズムは、従来は個々の具体的な問題に対して設計されてきたが、最近、文献[1]等によって、シストリックアルゴリズムで解き得る問題についての検討がなされ、一般的な設計法が示されている。これは多重ループプログラムを対象とし、ループ本体から添え字式、生成時点関数を求め、計算時刻関数、計算セル位置関数を得ることにより、セル構造回路網を決定する手法である。本研究では、この考え方を基本とし、一部発見的な手法も取り入れ、セル構造回路網を構成する。

### 1.3 文字認識アルゴリズム

文字認識アルゴリズムには、大きく分けて二つの手法がある。一つはパターン整合法であり、もう一つは構造解析法である。パターン整合法は、標準パターンと未知入力文字を重ね合わせたときの類似度をもとに認識を行なう方法であり、比較的高速に処理を行える。具体的には、未知入力文字から特徴ベクトルを抽出し、標準パターンの特徴ベクトルとの距離によって候補を選出する方法が広く用いられている。一方の構造解析法は、文字の線分関係に着目し、標準パターンとの線分の対応付けを行なうことによって認識を行なう方法であり、高精度の認識が期待できる。しかし、計算量が膨大であり、線分の正確な対応付けも難しいため、現段階ではあまり用いられていない。

本研究では、文字認識アルゴリズムとしては前者の手法を用い、文字認識を高速に実行するセル構造回路網を構成する。一方で、後者の手法で用いられる確率的弛緩法についての簡単な問題を取り上げ、処理の並列化について検討する。

### 1.4 本論文の構成

第1章は序論である。第2章では、パターン整合法による文字認識を並列に実行するセル構造回路網を構成し、配置面積、処理時間などを検討する。第3章では、構造解析法への応用の準備として、確率的弛緩法を用いた簡単な問題を解くセル構造回路網を構成する。第4章では、結論を述べる。

## 第2章 パターン整合法による文字認識を実行するセル構造回路網

### 2.1 パターン整合法によるアルゴリズム

本章では、パターン整合法による文字認識アルゴリズムの一種である、部分情報を用いた連想整合アルゴリズム<sup>[2]</sup>を取りあげる。これは、文字の部分情報を利用して分類を行なうことにより、文字を高速に識別する手法である。手順の概要を述べると、次のようになる。

- ① 未知入力文字から活字特徴量を求める。
- ② 49個のマス（横7×縦7）をマスごとにあらかじめ用意された各字種（以後、文字という）の標準パターン（マスごとの部分情報を用いて20クラスにクラスターリングしてから、各クラスをそれに属する文字の特徴量を表わすベクトルの平均値で代表させたもの）の対応するマスについて、各クラスごとに未知入力との距離計算を行い、距離の小さい順にクラスを取り出す。
- ③ クラスに入っている文字に対し、得点を与える。（未知入力との距離が小さいクラスほど高い得点を、小さい方から5位までのクラスに与える。）  
49個のマスから得た得点を合計する。
- ④ 得点の高い順から出力し、認識結果とする。ここでは、得点の高い方から5位までの文字を候補として出力する。

ここでは、上記の②～④の情報処理を並列に実行するセル構造回路網を構成する。

ところで、セル構造回路網について検討する場合に問題となる要因は、処理に要する時間と素子の配置に必要な面積である。これらの要因のうちどちらを重視するかによってアルゴリズムが異なる。そこで、本章では、処理速度を重視した回路、配置面積を重視した回路、及び双方の折衷型の回路をそれぞれ構成し、比較検討する。

なお、評価する際に文字数が必要な場合には、J I S第一水準の漢字数（2965文字）を用いる。

## 2.2 配置面積重視のセル構造回路網

### 2.2.1 クラスの選出 (ステップ②)

全体の構成を図2.1に示す。マス1～マス49のアレイの機能はすべて同じであるから、以下、マス1のアレイについてのみ、その機能を示す。

距離計算セル：各クラスのマス1の標準パターンの代表ベクトルを順に読み込み、未知入力のマス1のベクトルとの距離を計算する。距離としては、ユークリッド距離を用いるものとする。その距離の値及びクラスの番号を、クラス選出セルへ送る。ただし、クラスのデータをすべて(20クラス分)送り終えた後は、距離として取り得る値の最大値を送るものとする。

セルの構成例を図2.2に示す。ベクトルの成分として取り得る値の最大値は9ビットの整数であり、距離として取り得る値の最大値は19ビットの整数で表わされる<sup>[2]</sup>。なお、最初のデータがセル内に読み込まれた時刻を1とし、1クロックで時刻が1増加するものとするれば、クラス番号と時刻の値は一致する。従って、クラス番号としては、時刻の値を出力すればよい(図2.2では省略)。

クラス選出セル：各セルは、左側のセルから送られてきた距離及びクラス番号を読み込み、セル内に格納されている距離と比較する。送られてきた距離の方が小さければ、送られてきた距離とクラス番号をセル内に格納し、セル内に格納されていた距離及びクラス番号を次のセルに送る。逆に、送られてきた距離の方が大きいか等しければ、セル内のデータは変更せず、送られてきたデータをそのまま次のセルに送る。

ただし、クラス選出セル内に格納されている距離は、時刻0では理論上取り得る最大値であるものとする。

クラス選出セルの機能を関数で表わしたものを図2.3に、セルの具体的な構成例を図2.4に示す。距離は先に述べたように19ビットで表わせ、クラス番号は20以下であるから5ビットの整数で表わせる。

以上の動作を各セルが各クロックごとに行うと、20+5クロックでクラス選出が完了し、クラス選出セル(1)～(5)に、距離の小さい順にクラス番号と距

離が格納されることになる。

また、マス1～マス49のレイは同時に動作するから、全体として要するのは20+5クロックである。

## 2.2.2 得点計算及びソート(ステップ③④)

2.2.1節で用いた49×5個のクラス選出セルと、マスの個数(49個)の得点計算セルを用いて、各文字に得点を与える。更に、得点ソートセルで得点の高い順に5文字を出力し、認識結果とする。全体の構成を図2.5に示す。

以下、各セルの機能を示す。

クラス選出セル: マス $n$ 用の5個のセルは、選出されたクラス番号をマス $n$ 用得点計算セルに出力する。

得点計算セル: マス $n$ 用のセル( $1 \leq n \leq 49$ )は、マス $n$ でその文字がどのクラスに属するかを表すクラス番号を辞書から読み込み、クラス選出セルから送られてきたクラス番号と比較する。もし、クラス選出セルからのクラス番号の中に辞書からのクラス番号と一致するものがあれば、得点を与える。得点としては、クラス選出セル(1)と一致した場合には5点、(2)と一致した場合には4点、以下、3点、2点、1点を与えるものとする。この得点は、マス( $n-1$ )用のセルから送られてきた得点に加えられ( $n=1$ の場合はそのまま)、マス( $n+1$ )用のセル( $n=49$ の場合は得点ソートセル)に送られる。

クラス選出セルからのクラス番号の中に辞書からのクラス番号と一致するものがない場合は、マス( $n-1$ )用のセルから送られてきた得点をそのままマス( $n+1$ )用のセルに送る。

得点計算セルの機能を関数で表わしたものを図2.6に、セルの具体的な構成例を図2.7に示す。セル内で与えられる得点は5点以下であるから3ビットで表わせ、総得点は245点(=5点×49マス)以下であるから8ビットで表わせる。

得点ソートセル: セルの機能はクラス選出セルとほぼ等しく、次のようになる。

時刻0では、セル内に格納されている得点は0点であるとする。

各セルは、左側のセルから送られてきた得点及び文字番号を読み込み、セル



内に格納されている得点と比較する。送られてきた得点の方が大きければ、送られてきた得点と文字番号をセル内に格納し、セル内に格納されていた得点及び文字番号を次のセルに送る。逆に、送られてきた得点の方が小さいか等しければ、セル内のデータは変更せず、送られてきたデータをそのまま次のセルに送る。

得点ソートセルの機能を関数で表わしたものを図2.8に、セルの具体的な構成例を図2.9に示す。得点は先に述べたように8ビットで表わせ、文字番号はJIS第一水準の漢字2965文字を扱うものとするれば12ビットで表わせる。

得点計算セルは49個、得点ソートセルは5個存在するので、1文字目の処理が終了するまでに49+5クロック必要である。2文字目以降は、前の文字から1クロック遅れて処理されるので、文字数をNとすれば、すべての処理が完了するには、

$$49 + 5 + N - 1 \quad \text{クロック}$$

要する。

### 2.2.3 処理時間の評価

2.2.1、2.2.2節で述べた距離計算セル、クラス選出セル、得点計算セル、得点ソートセルは、すべて機能が異なるため、1回の動作に必要な時間は厳密には四種類のセルすべてで異なる。本章では、評価を簡単にするため、一回の加算（または減算・比較）に要する時間を基本単位として処理時間を考えることにする。この基本時間を $t_0$ とする。乗数がnビットの乗算は加算をn回繰り返して行なうものとするれば、処理時間は $n t_0$ となる。

以上のように定義することにより、各セルの動作に必要な時間が求まる。距離計算セルは、図2.2からわかるように、1回の減算と1回の乗算（9ビット）、2回の加算によって処理が行なわれるので、動作に必要な時間は

$$t_0 + 9 t_0 + 2 t_0 = 12 t_0 \quad \text{時間}$$

である。また、クラス選出セル、得点計算セル、得点ソートセルは、すべて $t_0$ 時間で処理できる。

距離計算セルが実際に距離計算を実行するのは最初の20クロックのみであるか

ら、この20クロックは1クロックが $12t_0$ 時間に相当し、残りは1クロックが $t_0$ 時間に相当する。従って、文字数を $N$ とすると、2.2.1、2.2.2節で求めたクロック数より、全体の処理時間は、

$$(20 \times 12 + 5 + 49 + 5 + N - 1) t_0 \quad \text{時間}$$

すなわち

$$(N + 298) t_0 \quad \text{時間} \quad (2.1)$$

である。 $N = 2965$ とすれば、 $3263 t_0$ 時間になる。

## 2.2.4 配置面積の評価

素子の配置面積を正確に評価するのは難しいので、本章では加算器（減算器）のみに着目し、1ビットの加算器（減算器）に必要な面積を基本単位として考える。この基本面積を $a_0$ とする。

距離計算セルは、図2.2からわかるように、9ビットの減算器4個、9ビットの乗算器4個、19ビットの加算器3個から構成される。9ビットの乗算は加算を9回繰り返すことによって行なうとすれば、18ビットの加算器で実現できる。従って、配置面積は、

$$(9 \times 4 + 18 \times 4 + 19 \times 3) a_0 = 165 a_0$$

となる。また、クラス選出セルは19ビットの減算器1個、得点計算セルは8ビットの加算器1個、得点ソートセルは8ビットの減算器1個から構成されるので、配置面積は、それぞれ、 $19 a_0$ 、 $8 a_0$ 、 $8 a_0$ である。

セル構造回路網は、距離計算セル49個、クラス選出セル（ $49 \times 5$ ）個、得点計算セル49個、得点ソートセル5個から構成されるので、全体の配置面積は

$$(165 \times 49 + 19 \times 49 \times 5 + 8 \times 49 + 8 \times 5) a_0 = 13172 a_0$$

である。

## 2.3 処理速度重視のセル構造回路網

### 2.3.1 クラスの選出（ステップ②）

クラス選出を行なうセル構造回路網としては、2.2.1節で用いたものをそのまま利用する。処理に必要なクロック数は、 $20 + 5$ クロックである。

### 2.3.2 得点計算 (ステップ③)

2.3.1節で用いた $49 \times 5$ のクラス選出セルと、文字数個(N個)の得点計算セルを用いて、各文字に得点を与える。

全体の構成を図2.10に示し、以下、各セルの機能を示す。

クラス選出セル: マス1用の5個のセルは、選出されたクラス番号をバスに出力する。このバスは、得点計算セルに接続されている。一方、マスn用のセル( $2 \leq n \leq 49$ )は、マス(n-1)用のセルにクラス番号を送る。

得点計算セル: 時刻0では、セル内に格納されている得点は0点であるとする。

各文字のセルごとに、2.2.2節と同様のクラス番号を辞書から読み込み、バスに送られてきたクラス番号と比較する。もし、バスのクラス番号の中に辞書からのクラス番号と一致するものがあれば、得点を与える。得点の与え方は2.2.2節と同様であり、この得点をセル内の得点に加え、新たに格納する。

マスの数が49個であるから、以上の動作を49回行なえば、各文字ごとの総得点を得られる。従って、得点計算に要するのは49クロックである。

得点計算セルの機能を関数で表わしたものを図2.11に、セルの具体的な構成例を図2.12に示す。2.2.2節で述べたようにセル内で与えられる得点は3ビットで表わせ、総得点は8ビットで表わせる。

### 2.3.3 得点のソート及び結果の出力 (ステップ④)

得点のソートには、図2.13に示す二進木状のセル構造回路網を用いる。図で、各ノードが1個のセルを表す。リーフは、2.3.2節で用いた得点計算用のセルである。リーフ以外のセルを得点ソートセルと呼び、以下に機能を示す。

時刻0では、リーフ以外の各セルの得点は-1であるとする。

各ノードのセルは、まず格納されている得点が-1か否かを調べる。もし-1でなければ、何もしない。-1であれば、左右の子の得点のうち大きい方を自分の得点とし、その子の漢字番号を自分の漢字番号とする。また、その子へリセット信号を送り、その子の得点を-1にする。

また、ルートセルは、データが送られた次のクロックで、そのデータを出力

し、自分の得点を -1 にする。

以上の動作を各セルが各クロックごとに同時に行うと、得点の高いものから順にルートへ送られ、出力される。文字の数を  $N$  とすれば、最も得点の高いものがルートへ送られるまでに  $\lceil \log_2 N \rceil$ 、出力されるまでに  $\lceil \log_2 N \rceil + 1$  だけのクロックが必要であり、以後 1 文字出力するごとに 2 クロックが必要となる。従って、上位  $k$  個を出力するとすれば、全体で

$$\lceil \log_2 N \rceil + 1 + 2 \times (k - 1) \quad \text{クロック} \quad (2.2)$$

要する。いま、 $N = 2965$  とし、上位 5 個を出力するとすれば、

$$11 < \log_2 2965 < 12$$

より、 $12 + 1 + 2 \times (5 - 1) = 21$  クロックとなる。

得点ソートセルの機能を関数で表わしたものを図 2.14 に、セルの具体的な構成例を図 2.15 に示す。2.2.2 節で述べたように、得点は 8 ビットで表わせ、文字番号は 12 ビットで表わせる。

このアルゴリズムを用いたソートの具体例 ( $N = 8$  の場合) を図 2.16 に示す。

#### 2.3.4 処理時間の評価

2.2.3 節と同様に「時間」を定義すると、得点計算セル (図 2.12)、得点ソートセル (図 2.15) の動作に要する時間は、両方とも  $t_0$  である。また、距離計算セルとクラス選出セルの処理時間は、2.2.3 節で求めたようにそれぞれ  $12t_0$ 、 $t_0$  である。従って、文字数を  $N$  とすると、2.3.1 ~ 2.3.3 節で求めたクロック数より、全体の処理時間は、

$$(20 \times 12 + 5 + 49 + \lceil \log_2 N \rceil + 1 + 2 \times (5 - 1)) t_0 \quad \text{時間}$$

すなわち

$$(\lceil \log_2 N \rceil + 303) t_0 \quad \text{時間} \quad (2.3)$$

である。 $N = 2965$  とすれば、 $315 t_0$  時間になる。

#### 2.3.5 配置面積の評価

2.2.4 節と同様に「面積」を定義すると、得点計算セル、得点ソートセルに必要な配置面積は、共に  $8 a_0$  である。また、距離計算セルとクラス選出セルの配置面積は、2.2.4 節で求めたようにそれぞれ  $165 a_0$ 、 $19 a_0$  である。

文字数を  $N$  とすると、セル構造回路網は、距離計算セル 49 個、クラス選出セル  $(49 \times 5)$  個、得点計算セル  $N$  個、得点ソートセル  $(N - 1)$  個から構成されるので、全体の配置面積は

$$(165 \times 49 + 19 \times 49 \times 5 + 8 \times N + 8 \times (N - 1)) a_0$$

すなわち

$$(16N + 12732) a_0 \quad (2.4)$$

である。  $N = 2965$  とすれば、  $60172 a_0$  となる。

## 2.4 折衷型のセル構造回路網

### 2.4.1 クラスの選出 (ステップ②)

クラス選出を行なうセル構造回路網としては、2.2.1 節で用いたものをそのまま利用する。処理に必要なクロック数は、  $20 + 5$  クロックである。

### 2.4.2 得点計算及びソート (ステップ③④)

文字の総数を  $N$  とする。まず、この  $N$  個の文字を  $K$  個 ( $2$  のべき乗) のグループに分け (各グループ内の文字数は  $(\lceil N/K \rceil - 1)$  個または  $\lceil N/K \rceil$  個)、グループごとに図 2.5 の回路を用いて 5 個の候補を選出する。次に、この候補を図 2.13 の回路を用いてソートすれば、結果が得られる。

図 2.17 に、  $N = 2965$  とし、四つのグループに分けた場合の全体の構成を示す。得点計算セルは 2.2.2 節で示したセル (図 2.6) であり、得点ソートセル II は 2.3.3 節で示したセル (図 2.14) である。得点ソートセル I は、グループ内の 5 個の候補を選出するまでは 2.2.2 節で述べた動作をするが、その後は得点ソートセル II に順番に候補を出力する役割を果たす。

各グループ内の文字数は高々  $\lceil N/K \rceil$  個であるから、グループ内の 5 個の候補を選出するには、2.2.2 節で検討したように

$$49 + 5 + \lceil N/K \rceil - 1 \quad \text{クロック}$$

必要であり、こうして選ばれた  $(K \times 5)$  個の候補の中から上位 5 個を出力するには、2.3.3 節で検討したように

$$\log_2 K + 1 + 2 \times (5 - 1) \quad \text{クロック}$$

要する。

### 2.4.3 処理時間の評価

各セルの動作に要する時間については、2.2.3、2.3.4節で述べたので、ここでは省略する。

まず、各グループ内の5個の候補を選出するのに必要な時間を求める。これは、グループ内に含まれる文字数が高々「 $N/K$ 」個であることから、2.2節で述べた文字認識を、文字数を「 $N/K$ 」個として実行する時間に等しい。従って、処理時間は、(2.2)式で $N$ の代わりに「 $N/K$ 」を代入したものであるから、

$$(\lceil N/K \rceil + 298) t_0 \quad \text{時間} \quad (2.5)$$

となる。また、選出された( $K \times 5$ )個の候補の中から上位5個を出力するのに要する時間は、2.3.3節で述べたソートを $K$ 個の文字について実行する時間に等しいから、(2.2)式より、

$$(\lceil \log_2 K \rceil + 1 + 2 \times (5 - 1)) t_0 \quad \text{時間}$$

すなわち、

$$(\lceil \log_2 K \rceil + 9) t_0 \quad \text{時間} \quad (2.6)$$

となる。従って、全体の処理時間は、

$$(\lceil N/K \rceil + \lceil \log_2 K \rceil + 307) t_0 \quad \text{時間} \quad (2.7)$$

と求められる。

### 2.4.4 配置面積の評価

2.2.4、2.3.5節で述べたように、距離計算セル、クラス選出セルに必要な面積はそれぞれ $165 a_0$ 、 $19 a_0$ であり、得点計算セル、得点ソートセルI、得点ソートセルIIに必要な面積はすべて $8 a_0$ である。

グループ数を $K$ とすると、セル構造回路網は、距離計算セル49個、クラス選出セル( $49 \times 5$ )個、得点計算セル( $49 \times K$ )個、得点ソートセルI( $5 \times K$ )個、得点ソートセルII( $K - 1$ )個から構成されるので、全体の配置面積は、

$$(165 \times 49 + 19 \times 49 \times 5 + 8 \times 49 \times K + 8 \times 5 \times K + 8 \times (K - 1)) a_0$$

すなわち

$$(440K + 12732) a_0 \quad (2.8)$$

となる。

## 2.5 逐次処理の処理時間の評価

前節までは、セル構造回路網による並列処理を実行した場合の処理時間、配置面積について検討した。本節では、比較のため、逐次処理で認識を実行した場合の処理時間を求めてみる。ただし、「時間」の定義は2.2.3節と同様である。

### (a) 距離計算

一つのベクトルは4次元であるから、距離を一回計算するのに4回の減算、4回の乗算(9ビット)、3回の加算が必要である。従って、計算時間は、

$$4 \times t_0 + 4 \times 9 t_0 + 3 \times t_0 = 43 t_0 \quad \text{時間}$$

である。距離計算は(49マス×20クラス)分実行されるので、全体の計算時間は、

$$43 t_0 \times 49 \times 20 = 42140 t_0 \quad \text{時間}$$

である。

### (b) クラス選出

マスごとに5個ずつのクラスを選出するので、一つのマスのクラス選出に要する比較回数の期待値は、距離の分布が一様であるとするれば、付録の(a.4)式より、

$$20 - 29.25 + 14 \sum_{k=1}^N \frac{1}{k} = 41.1 \quad \text{回}$$

となる。マスは49個存在するので、全体の計算時間(期待値)は

$$41.1 t_0 \times 49 = 2014 t_0 \quad \text{時間}$$

である。

### (c) 得点計算

便宜上、各クラスに含まれている文字数は等しいとする。文字の総数をNとすれば、20クラスにクラスタリングされているので、各クラスに含まれる文字数はN/20個である。1マスにつき上位5個のクラスに含まれる文字に得点を与え、マスの数は全部で49個であるから、得点計算は((N/20)×5×49)回の加算によって実行される。従って、処理時間は、

$$\left( (N/20) \times 5 \times 49 \right) t_0 = 12.25N t_0 \quad \text{時間}$$

である。N = 2965とすれば、

$$36321 t_0 \quad \text{時間}$$

となる。

#### (d) 得点のソート（候補の選出）

最後に、N個の文字に与えられた得点を比較し、得点の高い5文字を候補として出力する。N = 2965とし、得点の分布が一様であるとするれば、付録の(a.4)式より、比較回数の期待値は

$$2965 - 29.25 + 14 \sum_{k=1}^N \frac{1}{k} = 3056 \quad \text{回}$$

となる。すなわち、処理時間（期待値）は、

$$3056 t_0 \quad \text{時間}$$

である。

以上(a)～(d)より、逐次処理で認識を実行した場合の処理時間（期待値）は、文字数が2965個のとき、

$$83531 t_0 \quad \text{時間}$$

と求められる。

## 2.6 各方法の比較

文字数を2965とし、配置面積重視の並列処理、処理速度重視の並列処理、及び逐次処理の各方法を用いたときの処理時間、配置面積などを表2.1に示す。ただし、“AT<sup>2</sup>積”とは、セル構造回路網の“良さ”を表わす指標の一つであり、処理時間の二乗と配置面積との積である。また、折衷型の（）内の数字は、グループ数を表わす。表より、AT<sup>2</sup>積のみを指標とした場合には、折衷型でグループ数を32程度にしたものが最も“良い”と言える。このときの処理時間は、逐次処理で認識を実行した場合の約1/200である。



|                  |            | 処理時間                 | 配置面積                 | 逐次処理に対する処理時間    | AT <sup>2</sup> 積              |
|------------------|------------|----------------------|----------------------|-----------------|--------------------------------|
|                  | 逐次処理       | 83531 t <sub>0</sub> | —                    | —               | —                              |
| 並<br>列<br>処<br>理 | 面積重視       | 3263 t <sub>0</sub>  | 13172 a <sub>0</sub> | $\frac{1}{26}$  | $1.4 \times 10^{11} a_0 t_0^2$ |
|                  | 速度重視       | 315 t <sub>0</sub>   | 60172 a <sub>0</sub> | $\frac{1}{265}$ | $6.0 \times 10^9 a_0 t_0^2$    |
|                  | 折衷型 ( 2 )  | 1791 t <sub>0</sub>  | 13612 a <sub>0</sub> | $\frac{1}{47}$  | $4.4 \times 10^{10} a_0 t_0^2$ |
|                  | 折衷型 ( 4 )  | 1051 t <sub>0</sub>  | 14492 a <sub>0</sub> | $\frac{1}{79}$  | $1.6 \times 10^{10} a_0 t_0^2$ |
|                  | 折衷型 ( 8 )  | 681 t <sub>0</sub>   | 16252 a <sub>0</sub> | $\frac{1}{123}$ | $7.5 \times 10^9 a_0 t_0^2$    |
|                  | 折衷型 ( 16 ) | 497 t <sub>0</sub>   | 19772 a <sub>0</sub> | $\frac{1}{168}$ | $4.9 \times 10^9 a_0 t_0^2$    |
|                  | 折衷型 ( 32 ) | 405 t <sub>0</sub>   | 26812 a <sub>0</sub> | $\frac{1}{206}$ | $4.4 \times 10^9 a_0 t_0^2$    |
|                  | 折衷型 ( 64 ) | 360 t <sub>0</sub>   | 40892 a <sub>0</sub> | $\frac{1}{232}$ | $5.3 \times 10^9 a_0 t_0^2$    |

表 2. 1 各方法の比較

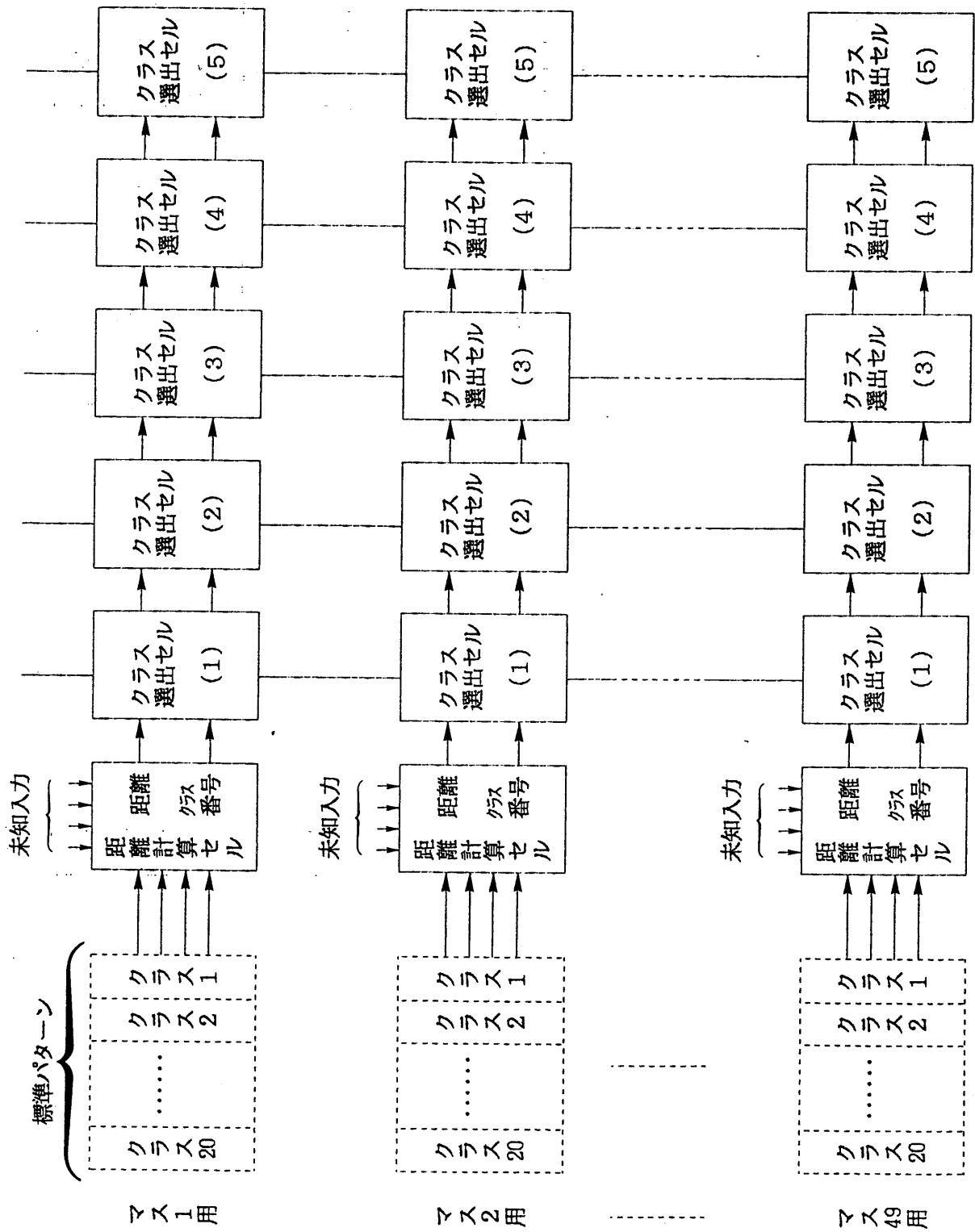


図2.1 クラス選出を行なうセル構造回路網

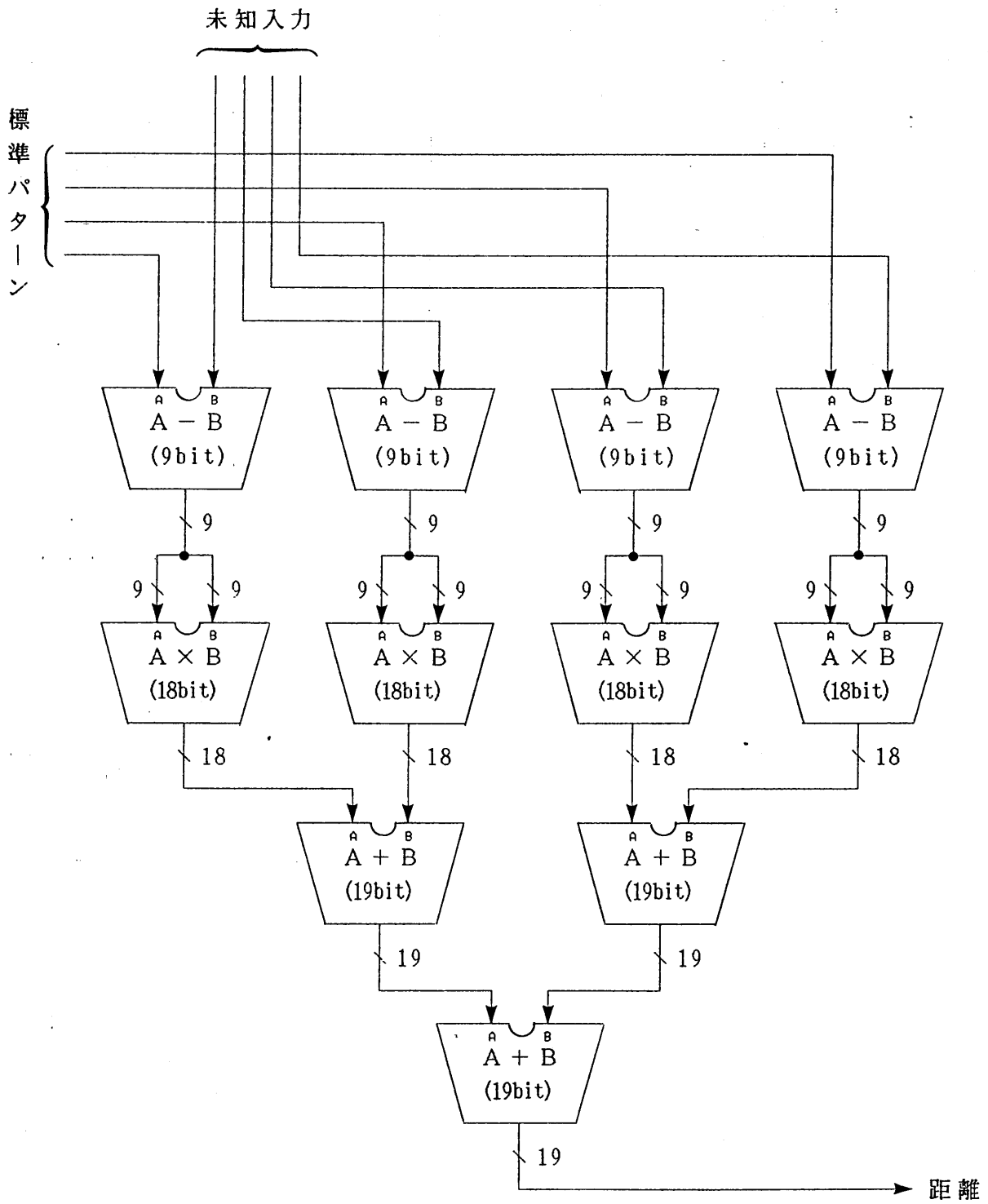
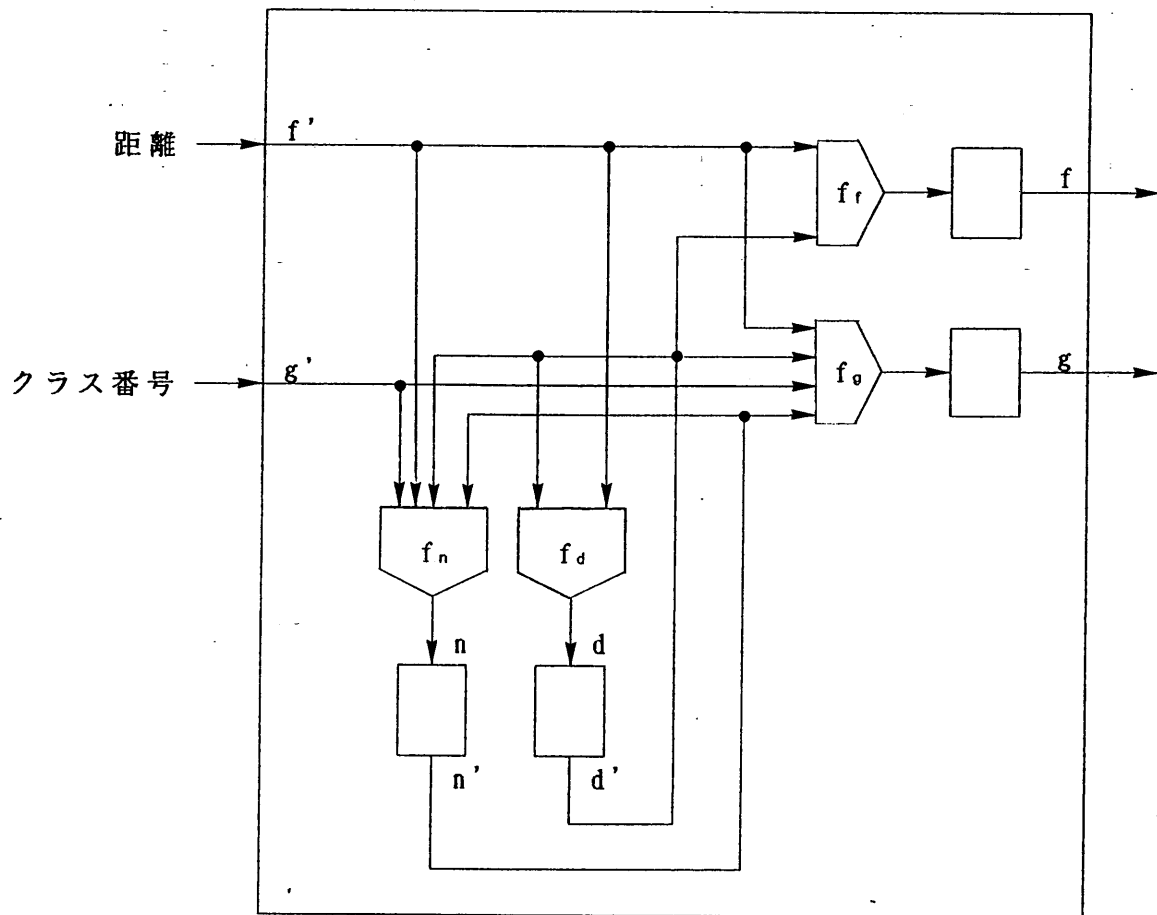


図 2.2 距離計算セル



$d$ : 格納している距離  
 $n$ : 格納しているクラス番号  
 $f$ : 次のセルに送る距離  
 $g$ : 次のセルに送るクラス番号

各関数の意味

$f_d : d = \text{if } f' < d' \text{ then } f' \text{ else } d'$   
 $f_n : n = \text{if } f' < d' \text{ then } g' \text{ else } n'$   
 $f_r : f = \text{if } f' < d' \text{ then } d' \text{ else } f'$   
 $f_g : g = \text{if } f' < d' \text{ then } n' \text{ else } g'$

図 2.3 クラス選出セル

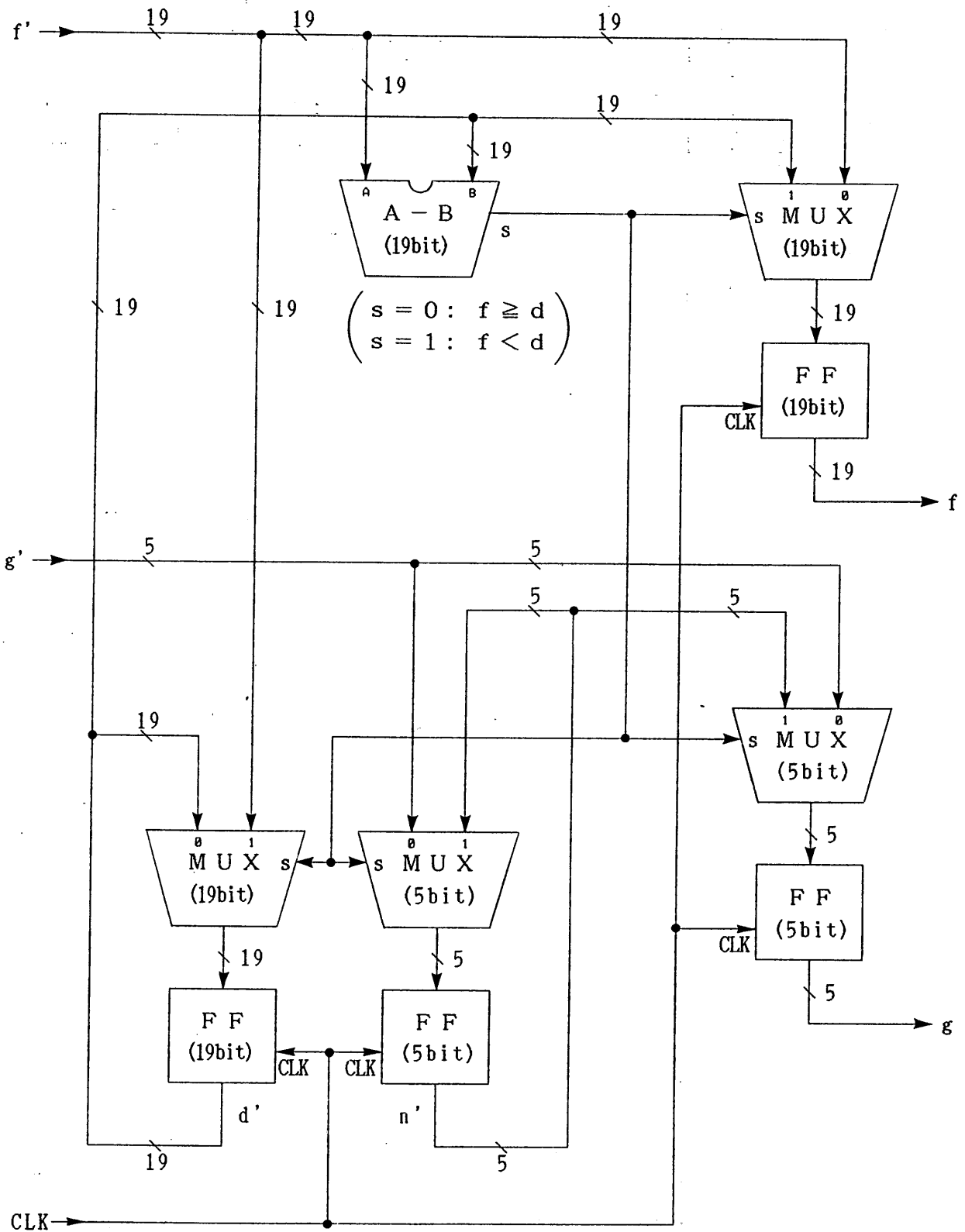


図 2.4 クラス選出セルの構成例

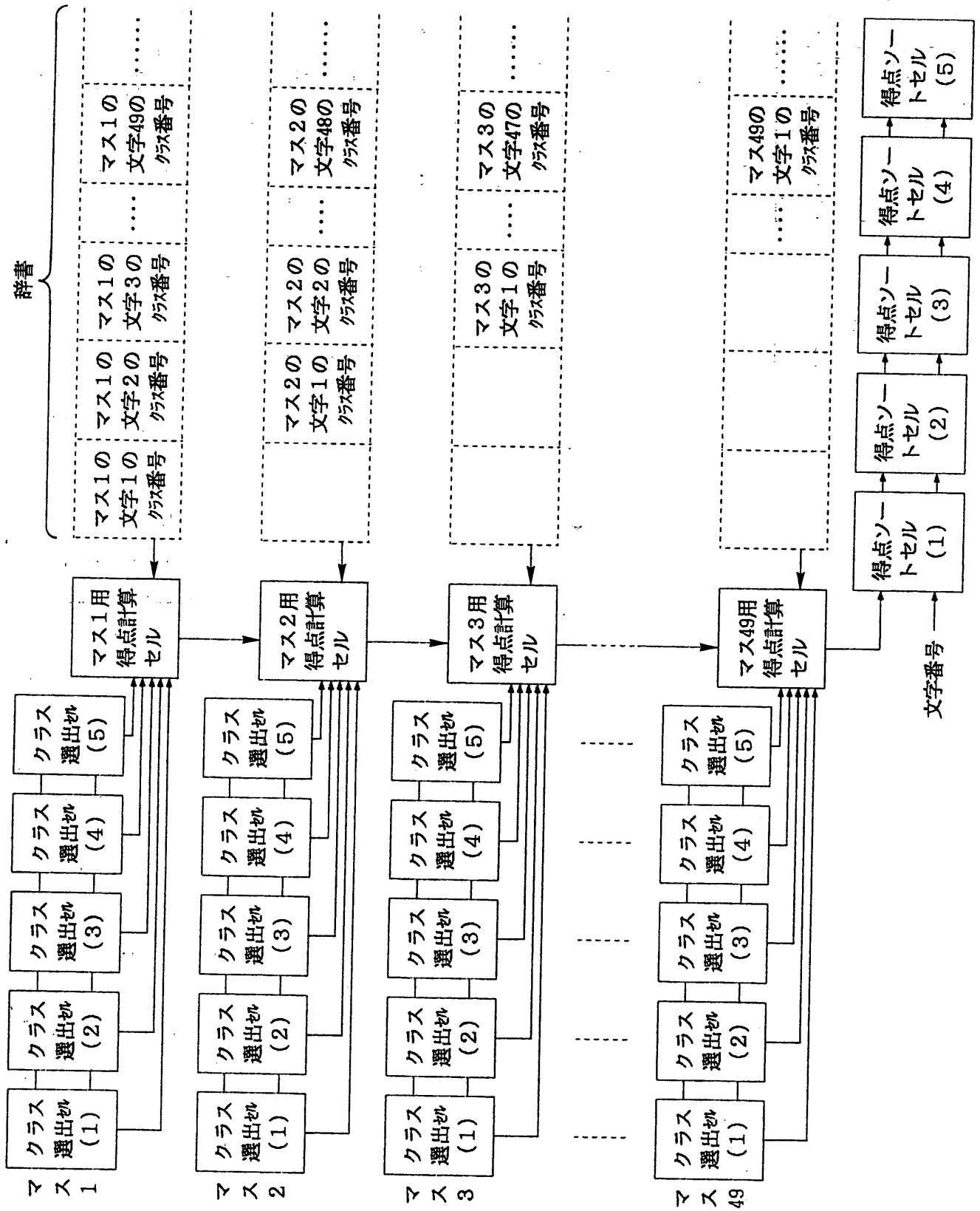
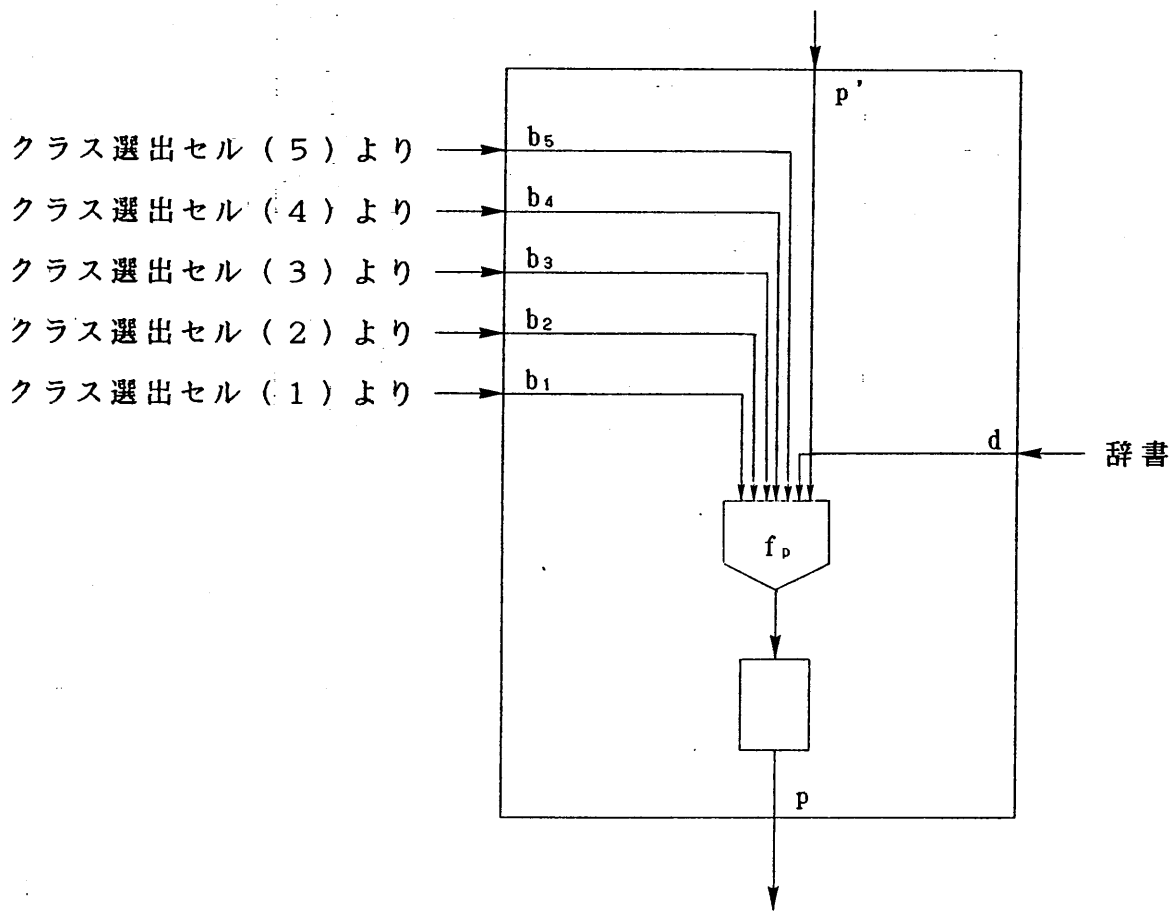


図 2.5 得点計算及びソートを行なうセル構造回路網



p: 格納している得点

関数の意味

```

fp : p = if b1==d then p'+5
        else if b2==d then p'+4
        else if b3==d then p'+3
        else if b4==d then p'+2
        else if b5==d then p'+1
        else p'

```

図 2.6 得点計算セル

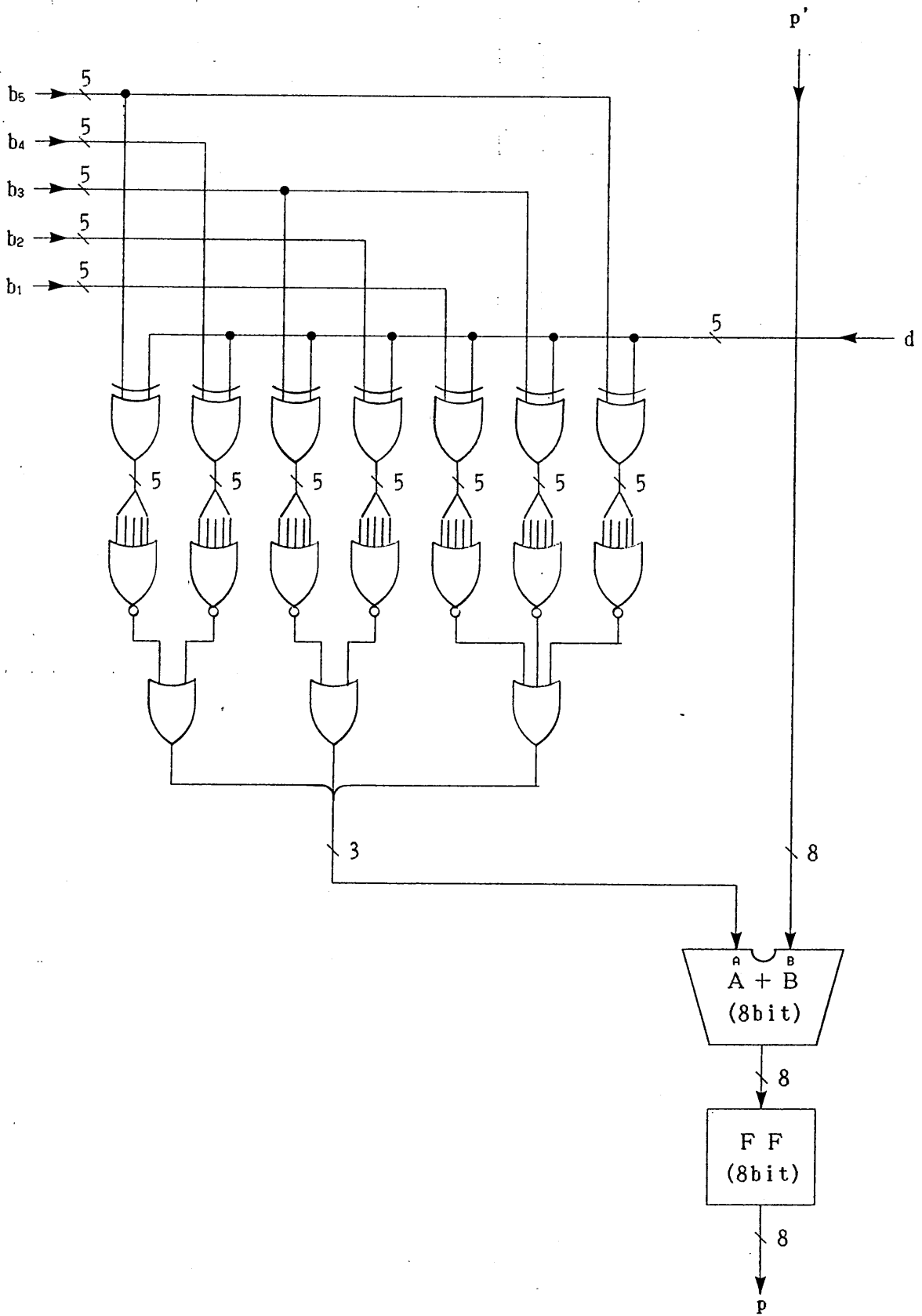
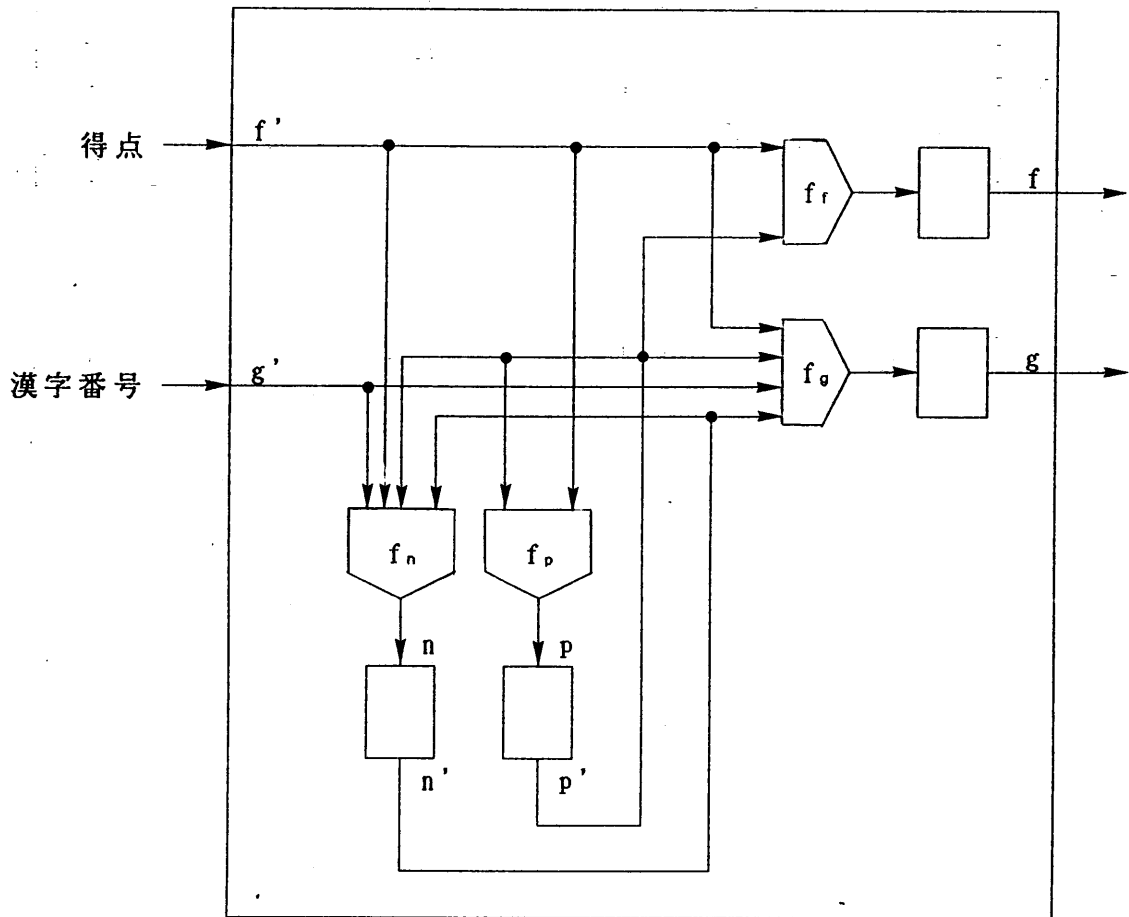


図 2.7 得点計算セルの構成例





p: 格納している得点  
 n: 格納している漢字番号  
 f: 次のセルに送る得点  
 g: 次のセルに送る漢字番号

各関数の意味

$f_p : p = \text{if } f' > p' \text{ then } f' \text{ else } p'$   
 $f_n : n = \text{if } f' > p' \text{ then } g' \text{ else } n'$   
 $f_r : f = \text{if } f' > p' \text{ then } p' \text{ else } f'$   
 $f_g : g = \text{if } f' > p' \text{ then } n' \text{ else } g'$

図 2.8 得点ソートセル

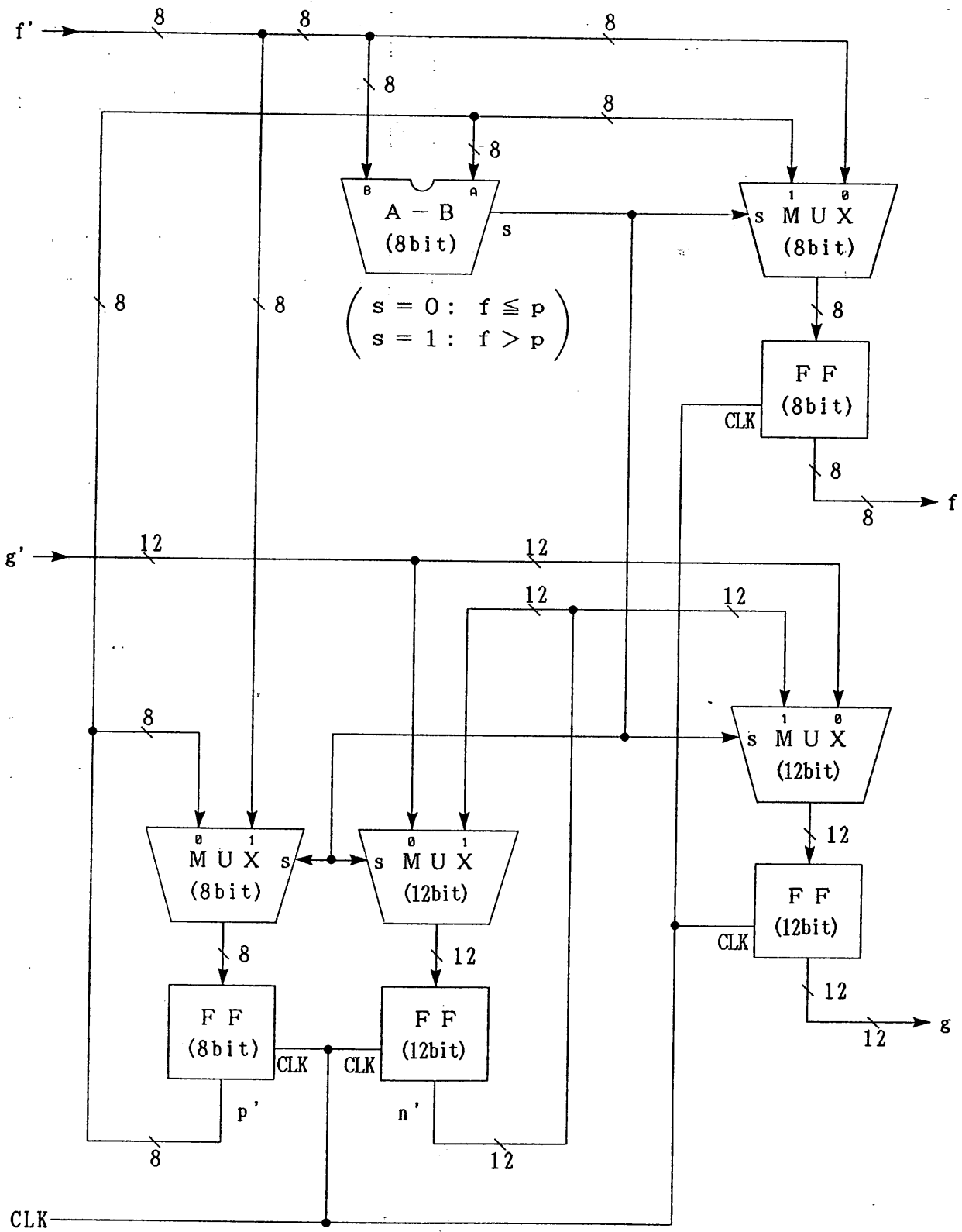


図 2.9 得点ソートセルの構成例

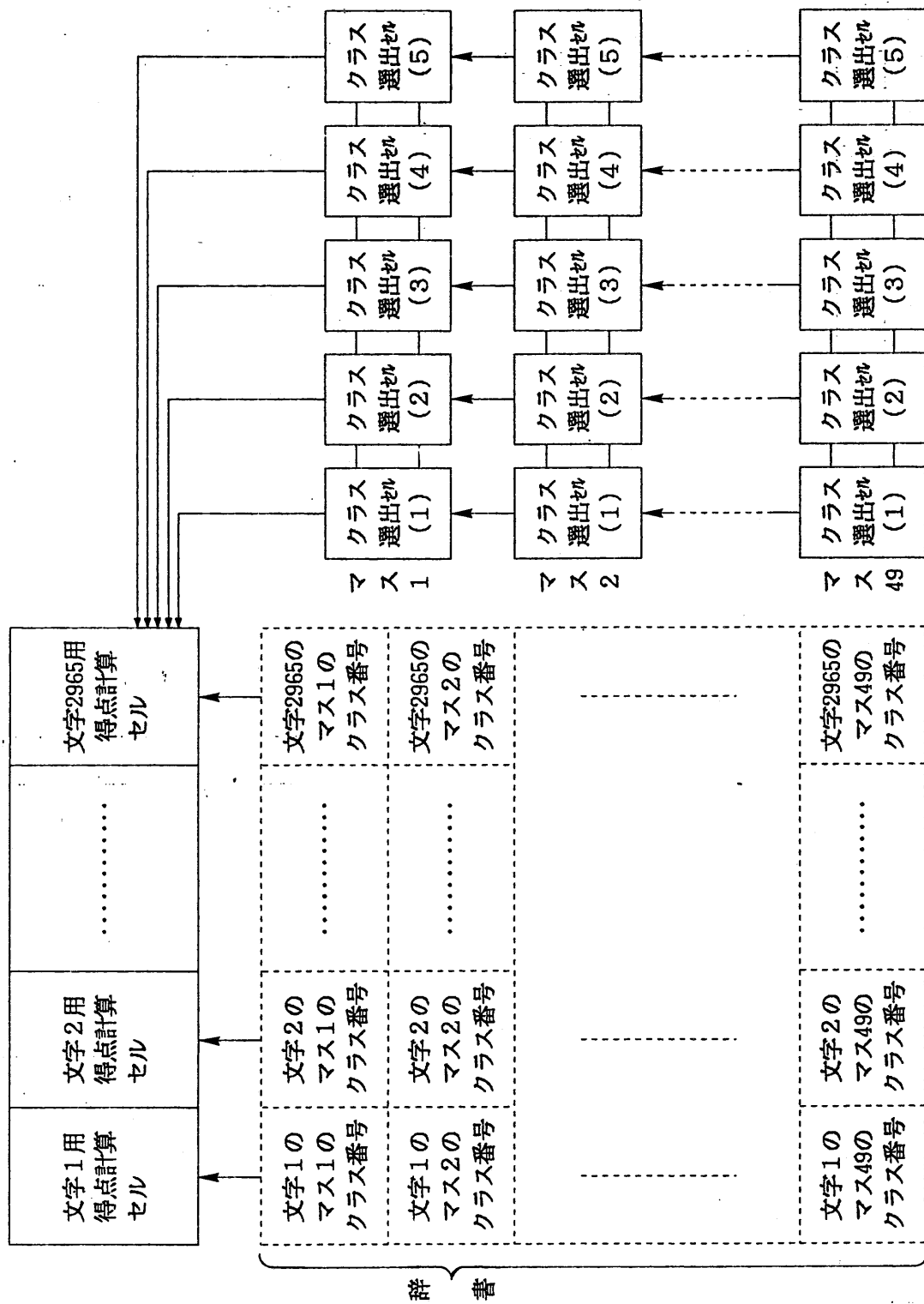
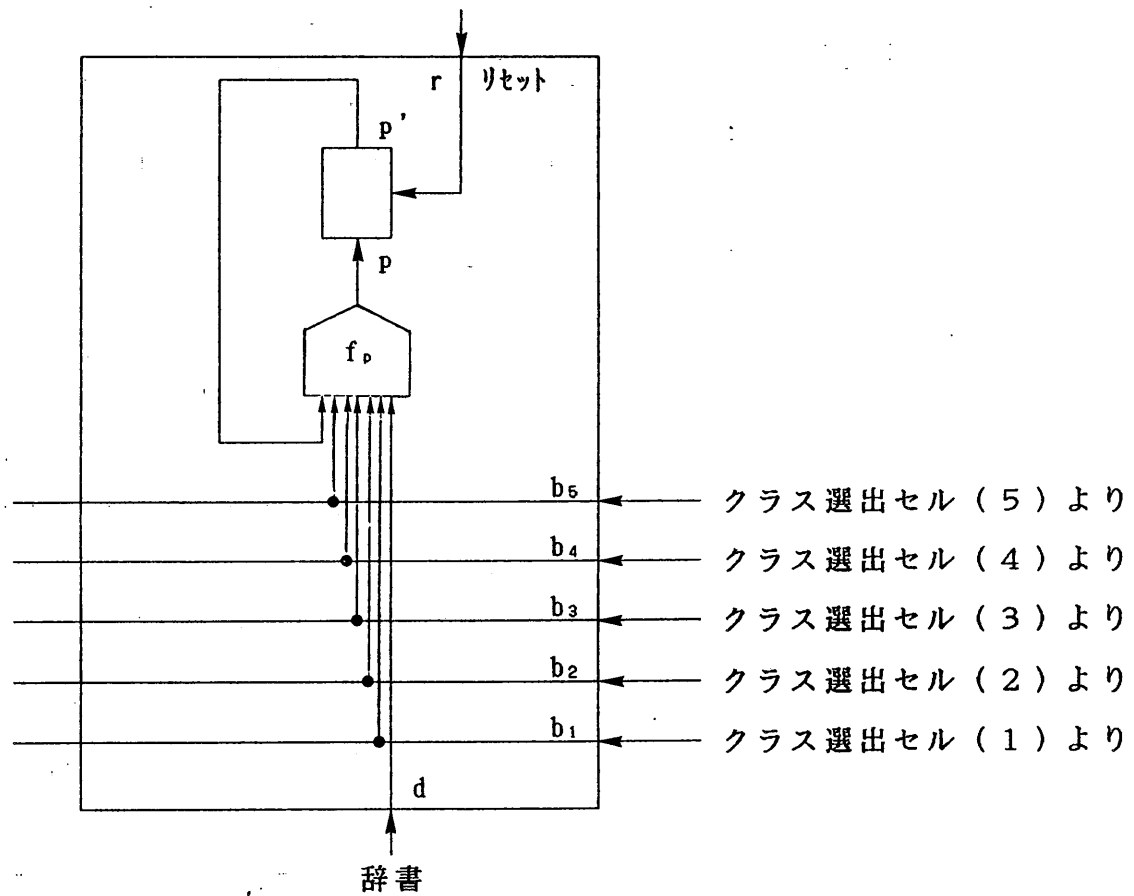


図 2.10 得点計算を行なうセル構造回路網



p: 格納している得点

関数の意味

```

fp : p = if b1==d then p'+5
        else if b2==d then p'+4
        else if b3==d then p'+3
        else if b4==d then p'+2
        else if b5==d then p'+1
        else p'

```

図 2.1 1 得点計算セル

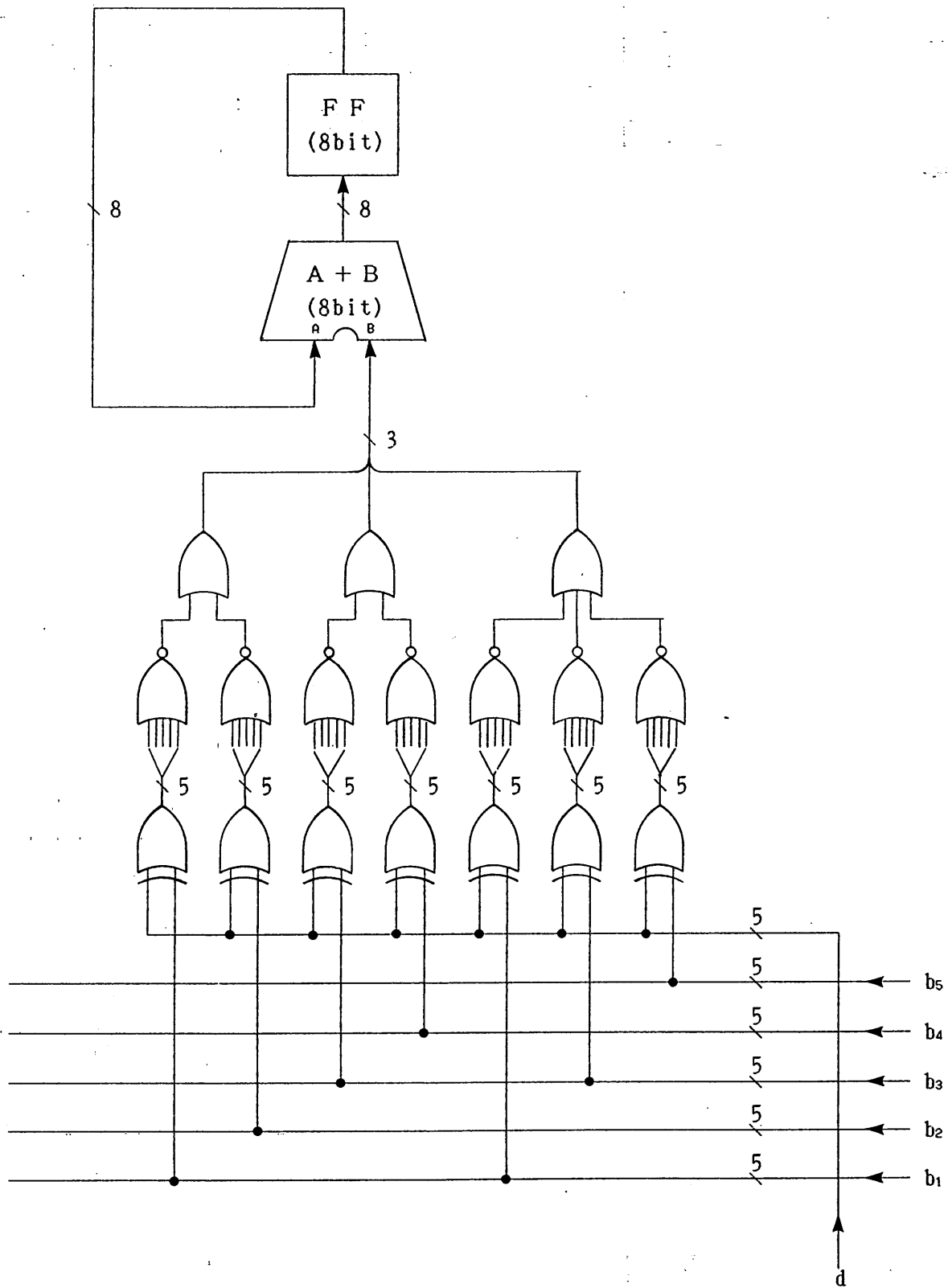
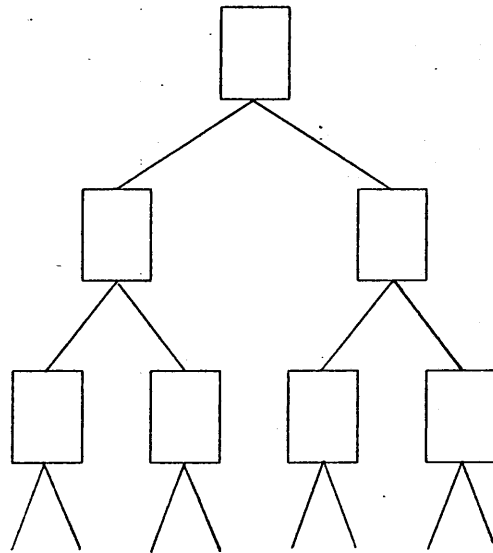
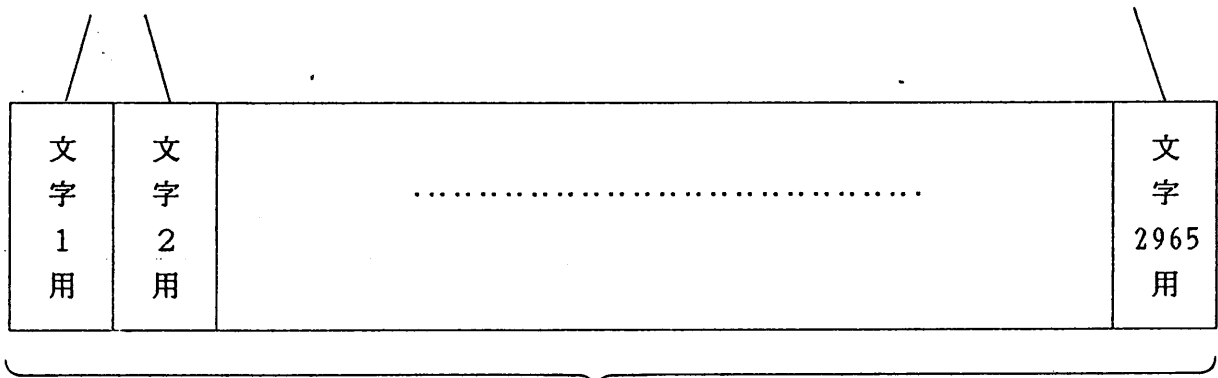


図 2.12 得点計算セルの構成例

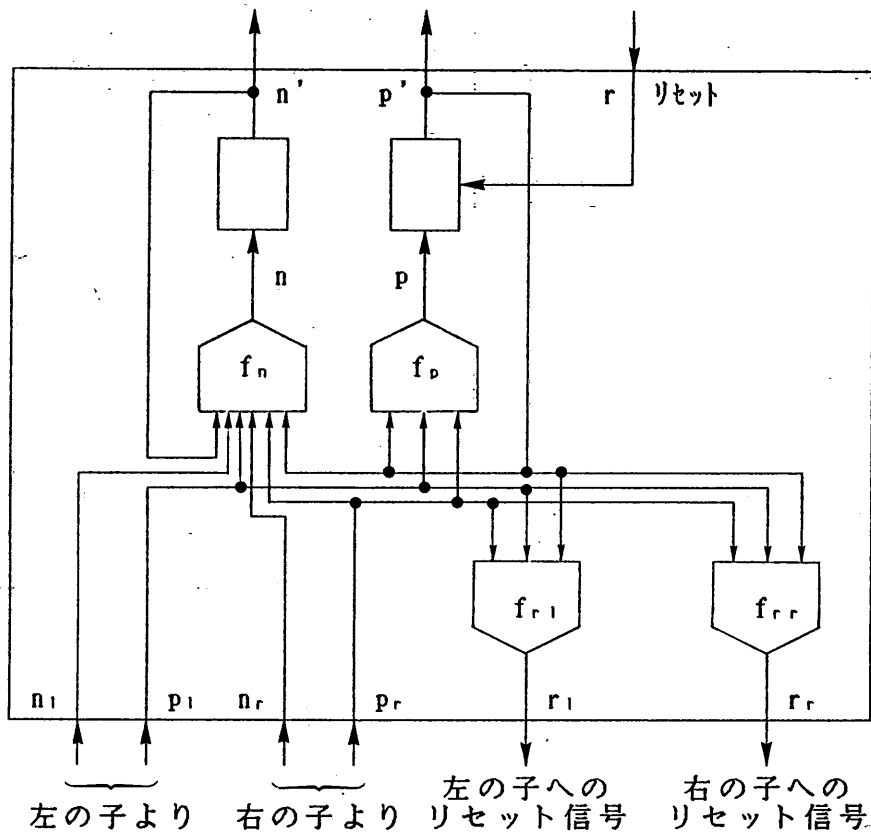


.....



得点計算セル

図 2.13 得点のソートを行なうセル構造回路網



$p$ : 格納している得点  
 $n$ : 格納している漢字番号  
 $r$ : リセット信号

各関数の意味

$f_p$ :  $p = \text{if } p' \neq -1 \text{ then } p'$   
            $\text{else if } p_l > p_r \text{ then } p_l$   
            $\text{else } p_r$   
 $f_n$ :  $n = \text{if } p' \neq -1 \text{ then } n'$   
            $\text{else if } p_l > p_r \text{ then } n_l$   
            $\text{else } n_r$   
 $f_{rl}$ :  $r_l = \text{if } p' \neq -1 \text{ then } 0$   
            $\text{else if } p_l > p_r \text{ then } 1$   
            $\text{else } 0$   
 $f_{rr}$ :  $r_r = \text{if } p' \neq -1 \text{ then } 0$   
            $\text{else if } p_l > p_r \text{ then } 0$   
            $\text{else } 1$

図 2.14 得点ソートセル

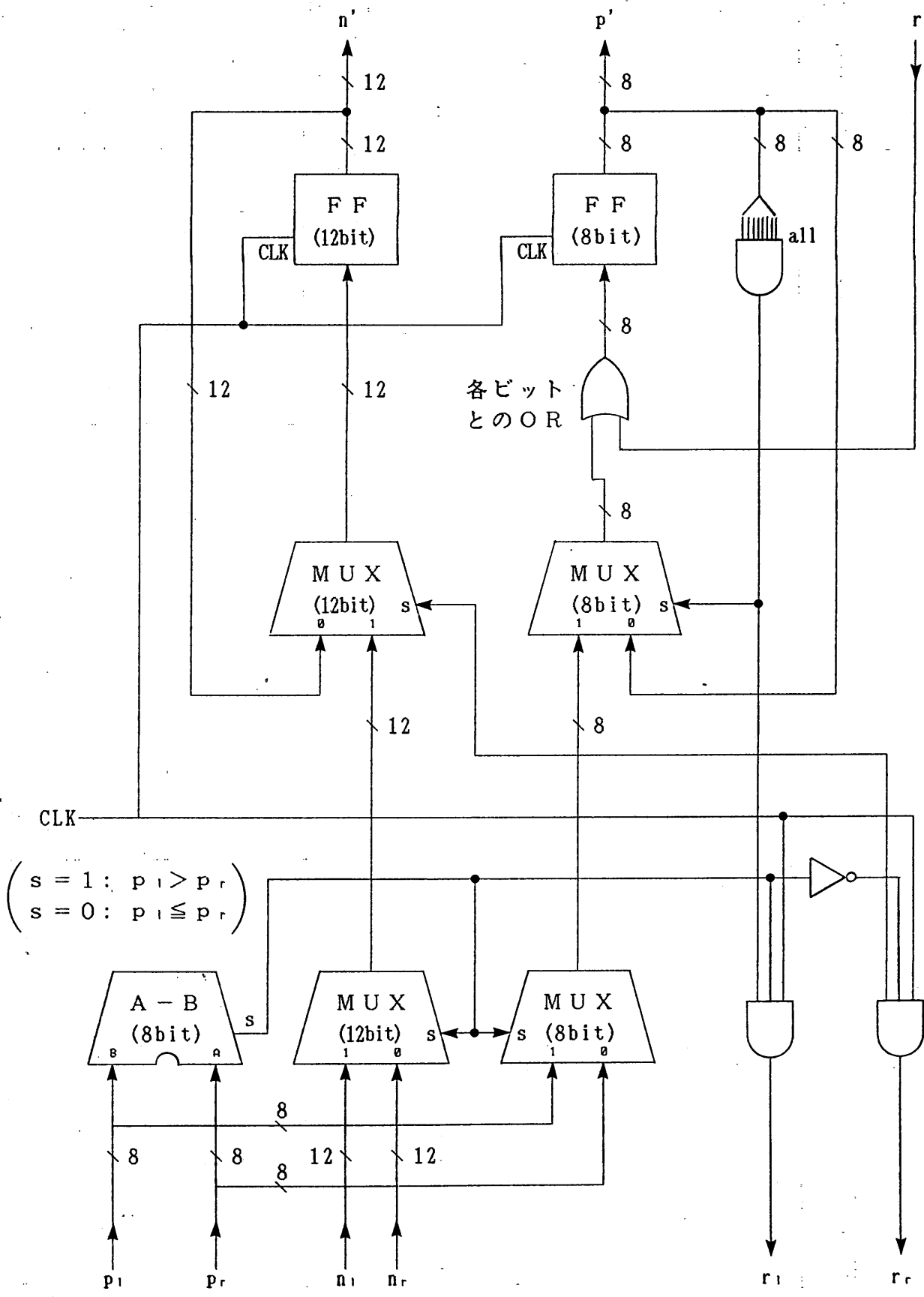
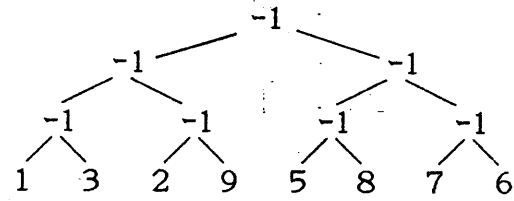


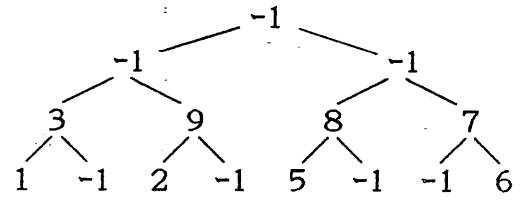
図 2.15 得点ソートセルの構成例-



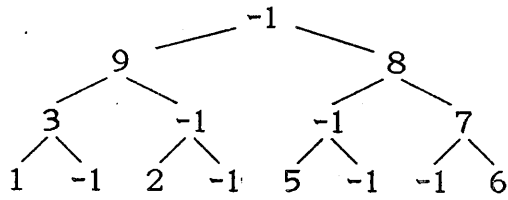
(1)



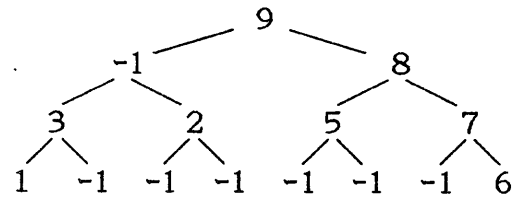
(2)



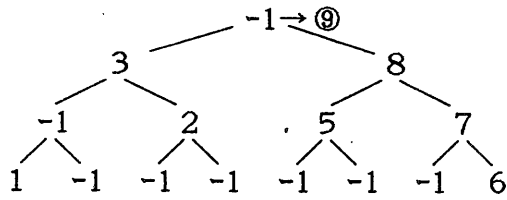
(3)



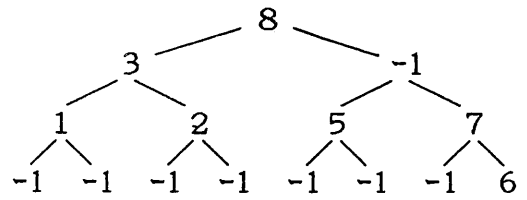
(4)



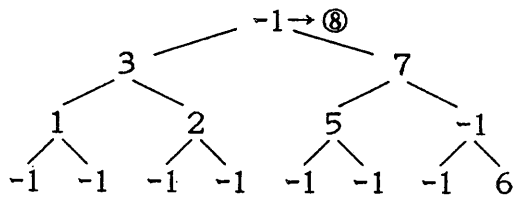
(5)



(6)



(7)



(8)

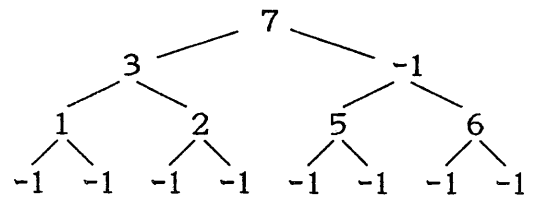


図 2.16 ソートにおけるデータの流れ

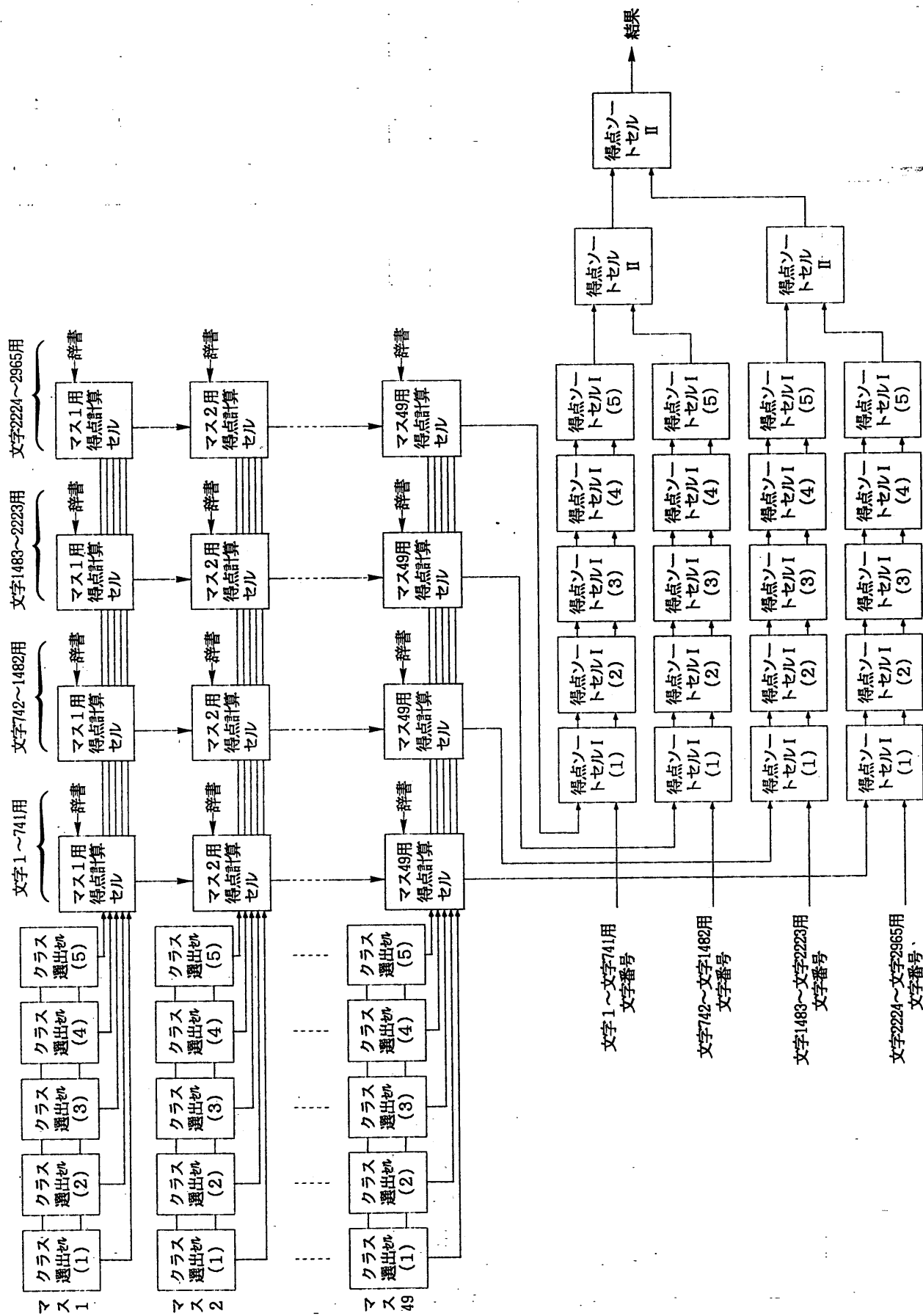


図 2.1.7 得点計算及びソートを行なうセル構造回路網

### 第3章 重み付きグラフの問題を解くセル構造回路網

#### 3.1 本章の目的

前章では、パターン整合法による文字認識アルゴリズムについて検討した。しかし、パターン整合法は、文字の類似度をもとに認識を行なうため、類似文字の識別は難しい。例えば、「王」と「玉」などは、ノイズ等の影響によって事実上認識不可能となる。このように、パターン整合法には限界があるので、類似文字識別のためには構造解析法を用いる必要がある。

ところで、文字の構造解析等を行なう際に用いられる手法の一つに、確率的弛緩法がある<sup>[3][4]</sup>。これは、対象がどのようなラベルを持つかが一意に定まらないとき、可能性のあるラベルをすべて与えておき、周囲とのつじつま合わせによって矛盾を取り除き、最適なラベルを得る手法である。これは、各ラベルに初期確率を与えておき、この確率をつじつま合わせによって何度か修正し、最終的に最も確率の高いラベルを最適であるとするものである。

本章では、確率的弛緩法を用いた簡単な問題として重み付きグラフの問題を取り上げ、この問題を解くセル構造回路網について検討する。

#### 3.2 重み付きグラフの問題<sup>[3]</sup>

図3.1のような、M個のグループからなり、各グループにN個のノードがある関係グラフを考えてみる（この場合は $M=3$ ,  $N=2$ ）。各ノード間には枝が存在し、各枝にはノード間の関係の強さを表わす重みを与えられている。重みが大きいほど関係が強いことになる。グループkのノードiを $c_i^{(k)}$ で表わし、 $c_i^{(k)}$ と $c_j^{(l)}$ の間の枝の重みを $r_{ij}^{(k,l)}$ で表わす。ただし、同じグループ内のノード間には枝は存在しないので、 $k \neq l$ である。

いま、各グループからノードを1個ずつ取ってきて得られる部分グラフの中で、枝の重みの和が最大のものを選ぶ問題を考えてみる。

自明な方法は、関係グラフに含まれるすべての部分グラフの枝の重みの和を求め、比較する方法である。しかし、グループの数がM、各グループ内のノードの数がNの時、部分グラフの数は $N^M$ 個になり、M、Nが大きいと膨大な計算量とな

るので実用的ではない。そこで、必ずしも最適な解は得られないが、準最適な解が得られる計算回数の少ない方法として、確率的弛緩法を用いる。まず、各ノードに等しい大きさの確率を与えておき、その確率を周りの状況に応じて何度か修正し、最終的に確率の大きいノードを選び出す。

ノード  $c_i^{(k)}$  の確率を  $s_i^{(k)}$  とおく。  $s_i^{(k)}$  の値を修正するには、  $c_i^{(k)}$  から出ている枝の重みと相手のノードの確率を参照する。つじつまが合えば、つまり、この場合なら枝の重みの和が大きくなるような状況であれば、  $s_i^{(k)}$  が大きくなるように修正する。

修正の際の修正作用素としては様々なものが提案され、文献[3]では、修正作用素の一般的な定義が与えられているが、ここではRosenfeldが提唱した次のような非線形の修正作用素を用いる。

$$s_i^{(k)} \leftarrow \frac{u_i^{(k)} \times s_i^{(k)}}{\sum_{j=1}^N (u_j^{(k)} \times s_j^{(k)})} \quad (3.1)$$

ここで、  $u_i^{(k)}$  は適合関数と呼ばれ、周囲とのつじつまが合えば大きい値を持ち、合わなければ小さい値を持つ関数である。また、分母は、  $s_i^{(k)}$  が確率であることから、任意の  $k$  に対して

$$\sum_{i=1}^N s_i^{(k)} = 1 \quad (3.2)$$

が成り立つように正規化するための関数である。グループ  $k$  の正規化関数として、  $\lambda^{(k)}$  を

$$\lambda^{(k)} = \frac{1}{\sum_{j=1}^N (u_j^{(k)} \times s_j^{(k)})} \quad (3.3)$$

と定義すれば、(3.1)式は次のように書ける。

$$s_i^{(k)} \leftarrow \lambda^{(k)} \times u_i^{(k)} \times s_i^{(k)} \quad (3.4)$$

適合関数は、先に述べた性質を持つ関数として、やはりRosenfeldが提唱した次のような関数を用いる。

$$u_i^{(k)} = \sum_{k \neq l} \max_j (r_{ij}^{(k,l)} \times s_j^{(l)}) \quad (3.5)$$

### 3.3 逐次処理に要する演算回数

本節では、3.2節で述べた修正を逐次処理で行なった場合の演算回数について検討する。確率的弛緩法では修正を何回か行なうが、まず、一回の修正に要する演算回数を求める。

ところで、3.2節の(3.3)～(3.5)式を見ると、すべての計算は

① 1回の乗算

② 1回の乗算と、1回の加算または比較

のいずれかを基本単位として行なわれていることがわかる。この章では、加算や比較に要する時間は乗算に要する時間に比べて無視できるものとして、①または②を1回の「演算」と考え、検討することにする。

#### (a) 適合関数の計算

適合関数  $u_i^{(k)}$  を求めるのに必要な演算回数は、(3.5)式より、ノード  $c_i^{(k)}$  から出ている枝の数に等しい。グループ  $k$  以外の  $(M-1)$  個のグループに、それぞれ  $N$  本ずつの枝が出ているので、一個の適合関数を求めるのに必要な演算回数は、

$$(M-1)N \text{ 回}$$

である。

ノードの数は全部で  $MN$  個であるから、すべての適合関数を求めるのに必要な演算回数は、

$$(M-1)N \times MN \text{ 回}$$

となる。

#### (b) 正規化関数の計算

グループ  $k$  の正規化関数の逆数

$$\frac{1}{\lambda^{(k)}} = \sum_{j=1}^N (u_j^{(k)} \times s_j^{(k)}) \quad (3.6)$$

を求めるには、グループ  $k$  内の各ノードについて適合関数  $u_i^{(k)}$  と確率  $s_i^{(k)}$  との積を求め、それらの和を計算する必要がある。従って、各グループ内での演算回数は

$N$  回

である。グループの数はM個であるから、全体の演算回数は

$$M \times N \quad \text{回}$$

となる。

(c) 確率の修正

確率を修正するには、各ノードについて、正規化関数を求める段階で計算済みの  $u_i^{(k)} \times s_i^{(k)}$  の値に正規化関数  $\lambda^{(k)}$  を掛ければよい。従って、演算回数は全体のノード数に等しく、

$$M N \quad \text{回}$$

である。

以上(a)~(c)より、全体の演算回数は

$$N^2 M (M - 1) + 2 M N \quad \text{回}$$

となる。これは、M、Nが大きいとき、

$$M^2 N^2 \quad \text{回}$$

と近似できる。

### 3.4 重み付きグラフの問題を解くセル構造回路網

#### 3.4.1 シストリックアルゴリズムを適用するための変更点

シストリックアルゴリズムを適用するために、3.2節で述べたアルゴリズムに若干の変更を加える。3.2節では、 $r_{ij}^{(k,l)}$  をノード  $c_i^{(k)}$  と  $c_j^{(l)}$  の間の枝の重みと定義したので、 $k \neq l$  であった。本節では、 $r_{ij}^{(k,l)}$  に代わる新しい値として、次の  $r_{ij}^{(k,l)}$  を用いる。

$$r_{ij}^{(k,l)} = \begin{cases} r_{ij}^{(k,l)} & (k \neq l) \\ 0 & (k = l) \end{cases}$$

以後、 $r_{ij}^{(k,l)}$  を  $r_{ij}^{(k,l)}$  と書く。

この変更によって、適合関数  $u_i^{(k)}$  は、次のように変わる。

$$u_i^{(k)} = \sum_{l=1}^M \max_j ( r_{ij}^{(k,l)} \times s_j^{(l)} ) \quad (3.7)$$

変更後の手順をまとめると、次のようになる。

- ① (3.7)式によって適合関数を求める。
- ② (3.6)式によって正規化関数の逆数を求める。
- ③ (3.4)式によって確率を修正する。

以下の節では、上記①～③の各ステップに分け、それぞれの情報処理を並列に実行するセル構造回路網を構成する。

### 3.4.2 適合関数の計算

適合関数を求めるには、 $M \times N$ 個の、クロックに同期して動作する適合関数計算セルを用いる。全体の構成及びデータの配置を図3.2に、適合関数計算セルを図3.3に示す。(3.7)式の適合関数を計算するために、適合関数計算セルは次のような機能を持つ。

- (a)  $r_{ij}^{(k,l)}$  と  $s_j^{(l)}$  の値を読み込み、積  $r_{ij}^{(k,l)} \times s_j^{(l)}$  を計算する。
- (b)  $j = 1 \sim N$  について、(a)で計算した積のうち最大のものを選ぶ。
- (c)  $l = 1 \sim M$  について、(b)で選んだ最大値の和を求める。

以下、図3.2で  $i$  行  $l$  列のセルを“セル( $i, l$ )”と表わすものとする。いま、セル( $1, 1$ )を例にとって説明する。これは、(3.7)式のうち

$$\max_j ( r_{1j}^{(k,1)} \times s_j^{(1)} ) \quad (k = 1, \dots, M) \quad (3.8)$$

を求めるセルである。 $k = 1$ の場合には、(3.8)式は、

$$\max_j ( r_{1j}^{(1,1)} \times s_j^{(1)} ) \quad (3.9)$$

となる。(3.9)式を計算するためには、 $r_{11}^{(1,1)}$  と  $s_1^{(1)}$ 、 $\dots$ 、 $r_{1N}^{(1,1)}$  と  $s_N^{(1)}$  の値を順に読み込み、積を計算し、その中の最大値を求めればよいことになる。

セル( $1, 1$ )は、まず最初のクロックで  $r_{11}^{(1,1)}$  と  $s_1^{(1)}$  の値を読み込み、積を計算する。これは最初のデータであるから、そのままセル内に格納する。次のクロックからは、 $r_{1j}^{(1,1)}$  と  $s_j^{(1)}$  ( $j = 2 \sim N$ ) の値を順に読み込んで積を計算し、セル内に格納されている値と比較し、大きい方を新たに格納する。

以上の動作が完了すると、(3.9)式の値が求まる。この値は、 $l$  についての和をとるために、次のクロックでセル( $1, 2$ )に送られる。同時にセル( $1, 1$ )

内では、 $k = 2$  の場合について  $k = 1$  の場合と同様の動作を始める。セル  $(1, \ell)$  ( $2 \leq \ell \leq M$ ) では、セル内で求めた最大値にセル  $(1, \ell - 1)$  から送られてきた値を加え、セル  $(1, \ell + 1)$  に送る。図 3.2 のようにデータを配置することにより、セル  $(i, \ell - 1)$  から最大値が送られてくる時刻とセル  $(i, \ell)$  の最大値が求まる時刻は一致する。

以上の動作を一般的に記述したものが、図 3.3 に示した関数である。

### 3.4.3 正規化関数の計算

正規化関数の逆数を (3.6) 式で求めるには、図 3.4 のセル構造回路網を用いる。正規化関数計算セルの機能に関数で表わしたものを図 3.5 に示す。 $i$  列目の正規化関数計算セルは、図 3.2 の回路で求めた適合関数  $u_i^{(k)}$  と、確率  $s_i^{(k)}$  を読み込み、積を求める。この積は、確率修正セルに送られると同時に、 $(i - 1)$  列目のセルから送られてきた値に加えられ、 $(i + 1)$  列目のセルに送られる。 $N$  列目のセルの動作が完了すると、結果として  $N$  列目のセルから正規化関数の逆数が出力される。なお、3.4.2 節で述べたことからわかるように、適合関数  $u_i^{(k)}$  は  $N$  クロックごとに出力されるので、確率  $s_i^{(k)}$  も  $N$  クロックごとに取り込む。

### 3.4.4 確率の修正

(3.4) 式に従って確率を修正するには、図 3.6 の確率修正セルを用いる。セル構造回路網は図 3.4 に示されている。

$i$  列目の確率修正セルは、 $i$  列目の正規化関数修正セルで計算されて送られてきた  $u_i^{(k)} \times s_i^{(k)}$  の値に正規化関数  $\lambda^{(k)}$  を掛け、修正された新しい確率として出力する。正規化関数は、そのまま  $(i + 1)$  列目のセルに送る。

### 3.4.5 処理時間の検討

本節では、3.4.2 ~ 3.3.4 節で構成したセル構造回路網の演算回数について検討する。確率的弛緩法では修正を何回か行なうが、まず、一回の修正に要する演算回数を求める。なお、演算回数の定義は 3.3 節と同様である。また、各セルが 1 クロックで行なう処理は、3.3 節で定義した一回の「演算」であるから、クロック数と演算回数は等しい。



まず、最初のグループ ( $k = 1$ ) の確率の修正が完了するまでの演算回数 (クロック数) を求める。

(a) 適合関数の計算

3.4.2 節で検討したように、最大値を求めるのに  $N$  クロック、最大値の和を計算するのに  $M$  クロック 必要である。従って、適合関数の計算には、

$$M + N \quad \text{クロック}$$

要する。

(b) 正規化関数の計算

3.4.3 節で検討したように、正規化関数を求めるには  $N$  個の正規化関数計算セルで演算が行なわれなければならない。必要な時間は、

$$N \quad \text{クロック}$$

である。

(c) 確率の修正

3.4.3 節で検討したように、 $N$  個のノードの確率をそれぞれ一個の確率修正セルを用いて修正する。必要な時間は、

$$N \quad \text{クロック}$$

である。

以上 (a) ~ (c) より、最初のグループ ( $k = 1$ ) の確率の修正が完了するまでのクロック数は、

$$M + 3N \quad \text{クロック}$$

である。また、残りの ( $M - 1$ ) 個のグループ ( $2 \leq k \leq M$ ) の演算は、それぞれ前のグループの演算が終了してから  $N$  クロックで完了するから、全体に必要な演算回数は、

$$M + 3N + N(M - 1) \quad \text{回}$$

すなわち、

$$MN + M + 2N \quad \text{回}$$

である。これは、 $M$ 、 $N$  が大きいとき、

$$MN \quad \text{回}$$

と近似できる。

### 3.5 逐次処理と並列処理の演算回数の比較

重みつきグラフの問題を、自明な方法で解いた場合、確率的弛緩法を用いて逐次処理で解いた場合、並列処理で解いた場合のそれぞれの演算回数を表3.1に示す。ただし、 $t_0$ は弛緩法の繰返し回数である。

| 方法   | 自明な方法 | 確率的弛緩法<br>(逐次処理) | 確率的弛緩法<br>(並列処理) |
|------|-------|------------------|------------------|
| 演算回数 | $N^M$ | $t_0 (MN)^2$     | $t_0 MN$         |

表3.1 演算回数の比較

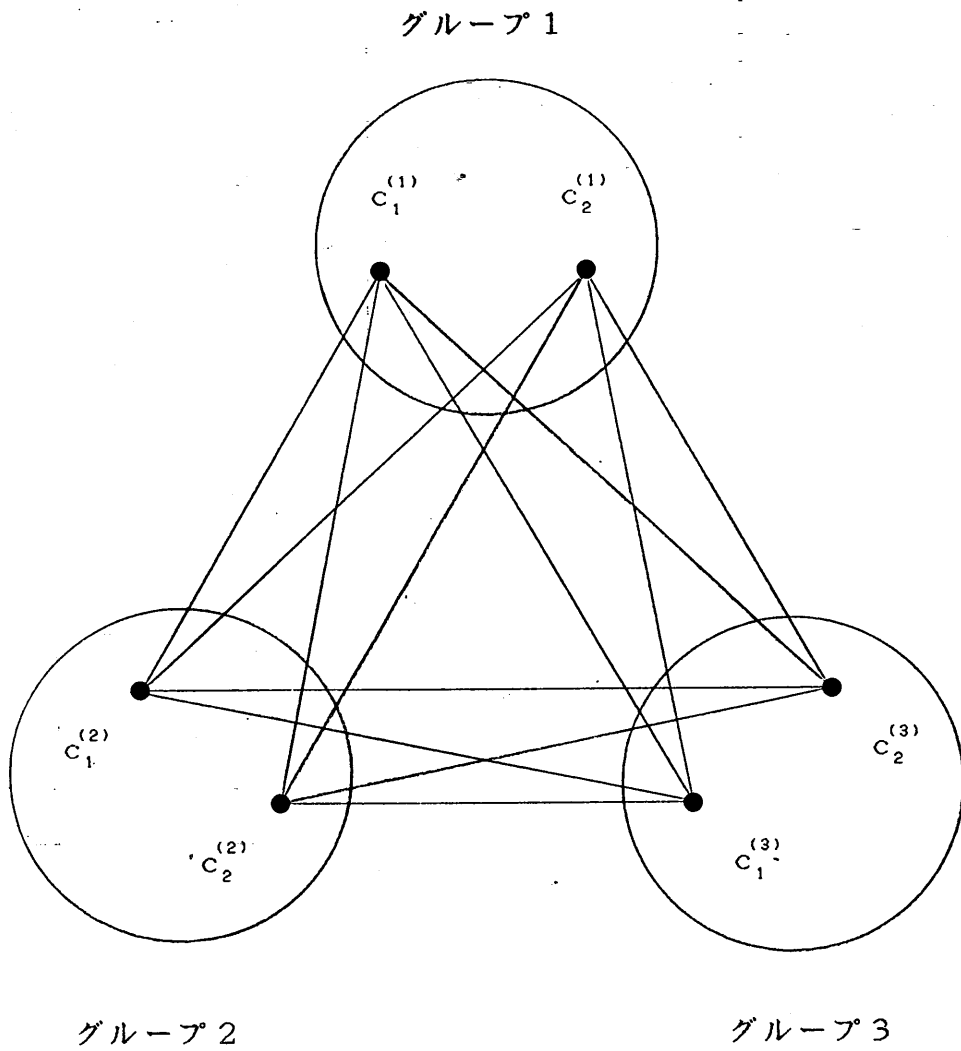
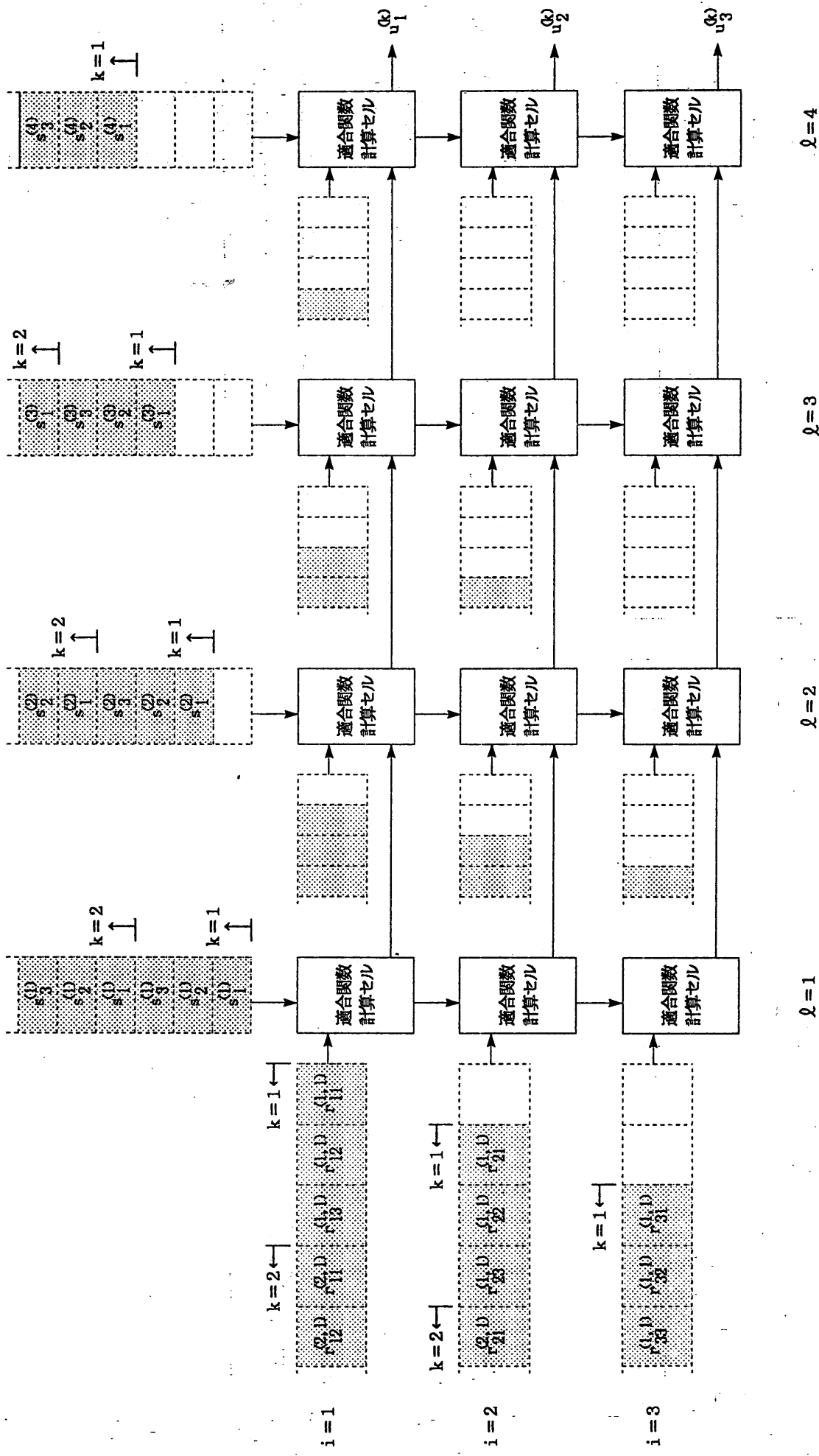
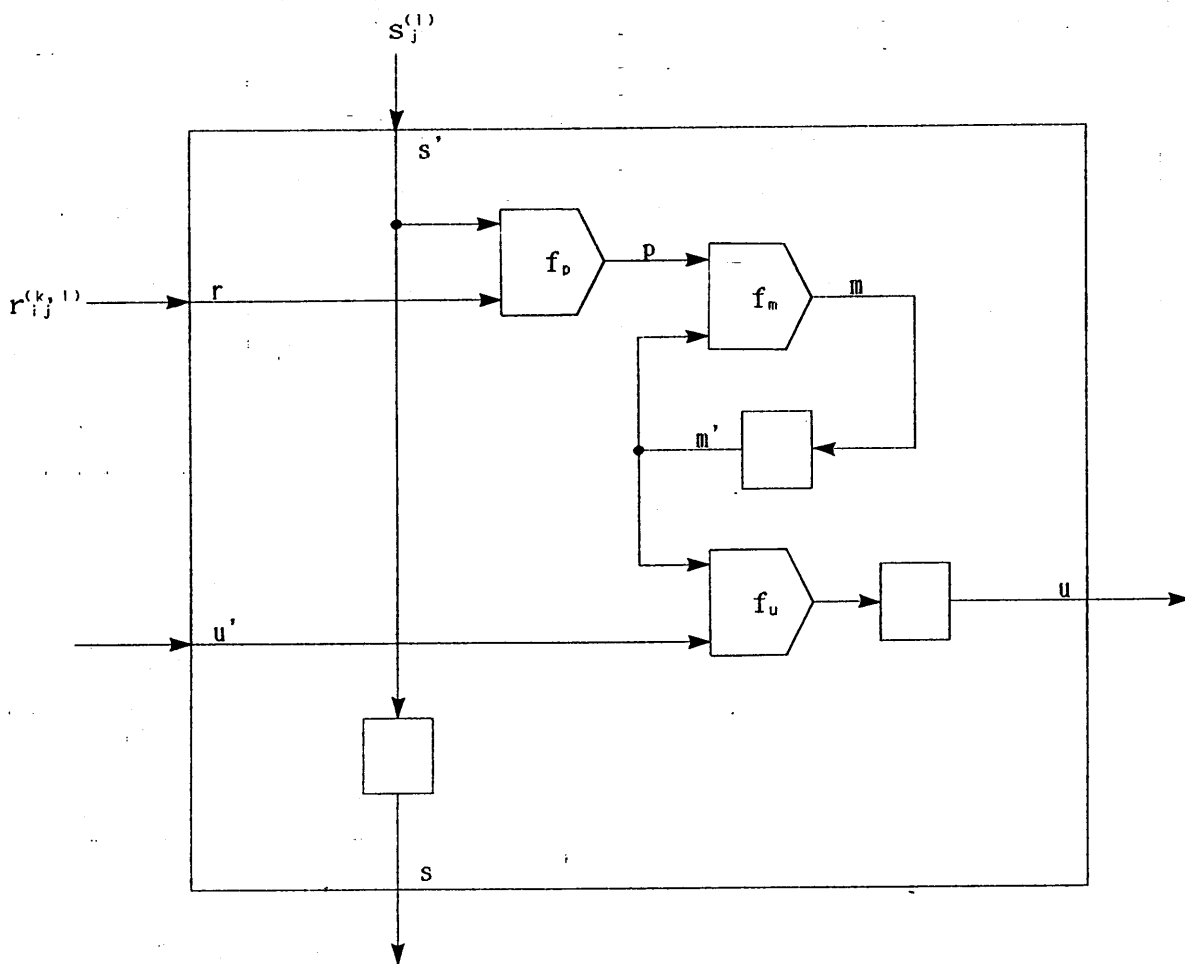


図 3.1 関係グラフ (M = 3, N = 2)



は、データの存在を表わす。

図 3.2 適合関数を計算するセル構造回路網



$s$ :  $s_j^{(1)}$  の値  
 $r$ :  $r_{ij}^{(k,l)}$  の値  
 $m$ : 格納している最大値  
 $u$ : 次のセルに送る  $u^{(k)}$  の値

各関数の意味

$f_p$ :  $p = s \times r$

$f_m$ :  $m = \text{if } (T \bmod N) == 1 \text{ then } p$   
 $\quad \text{else if } m' > p \text{ then } m'$   
 $\quad \text{else } p$

$f_u$ :  $u = \text{if } (T \bmod N) == 1 \text{ then } u' + m'$   
 $\quad \text{else } 0$

[  $T$ : 時刻 (最初のデータが入力された時刻が  $T = 1$ ) ]

[  $N$ : 各グループ内の候補の個数 ]

図 3.3 適合関数計算セル

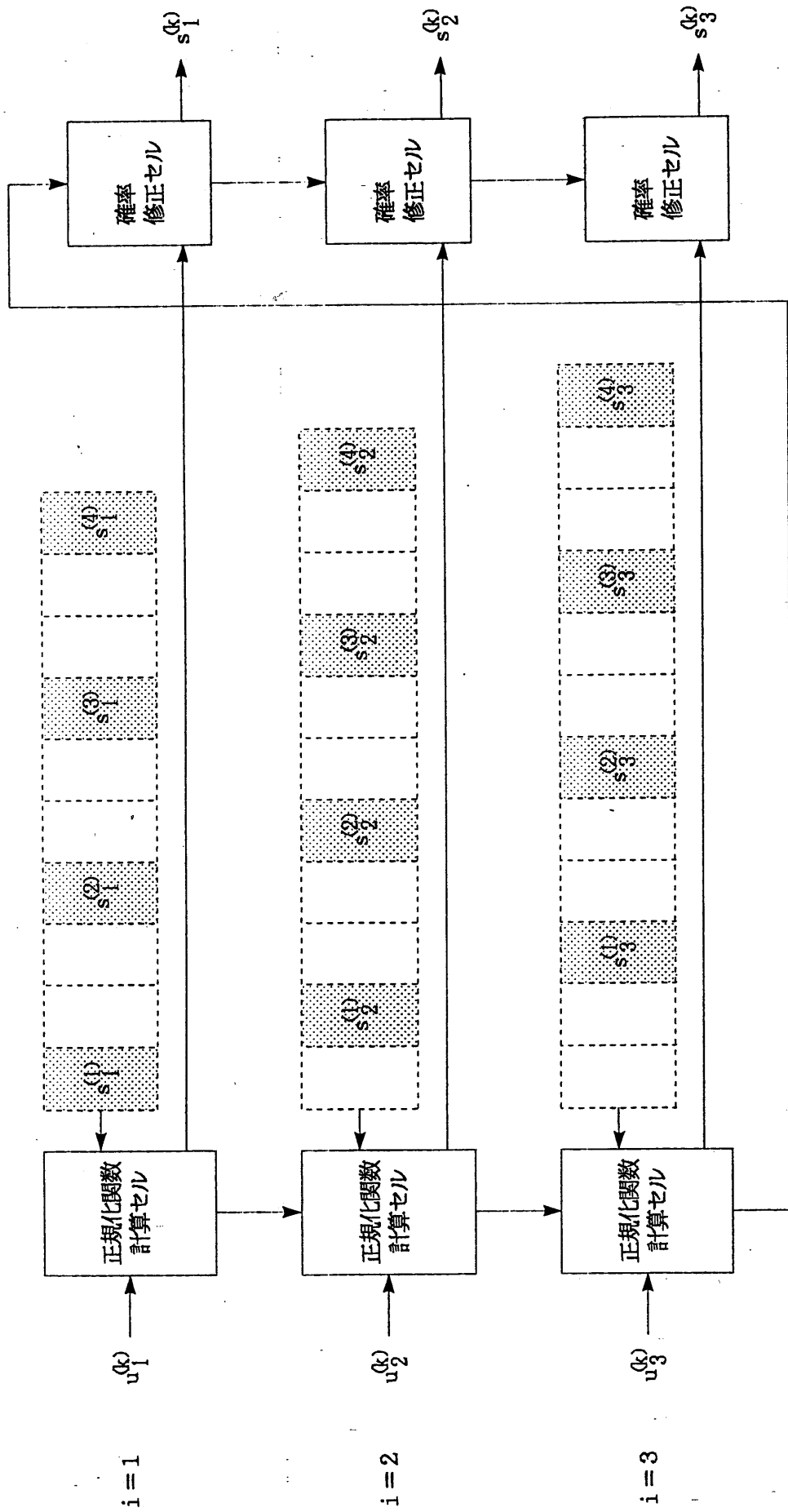
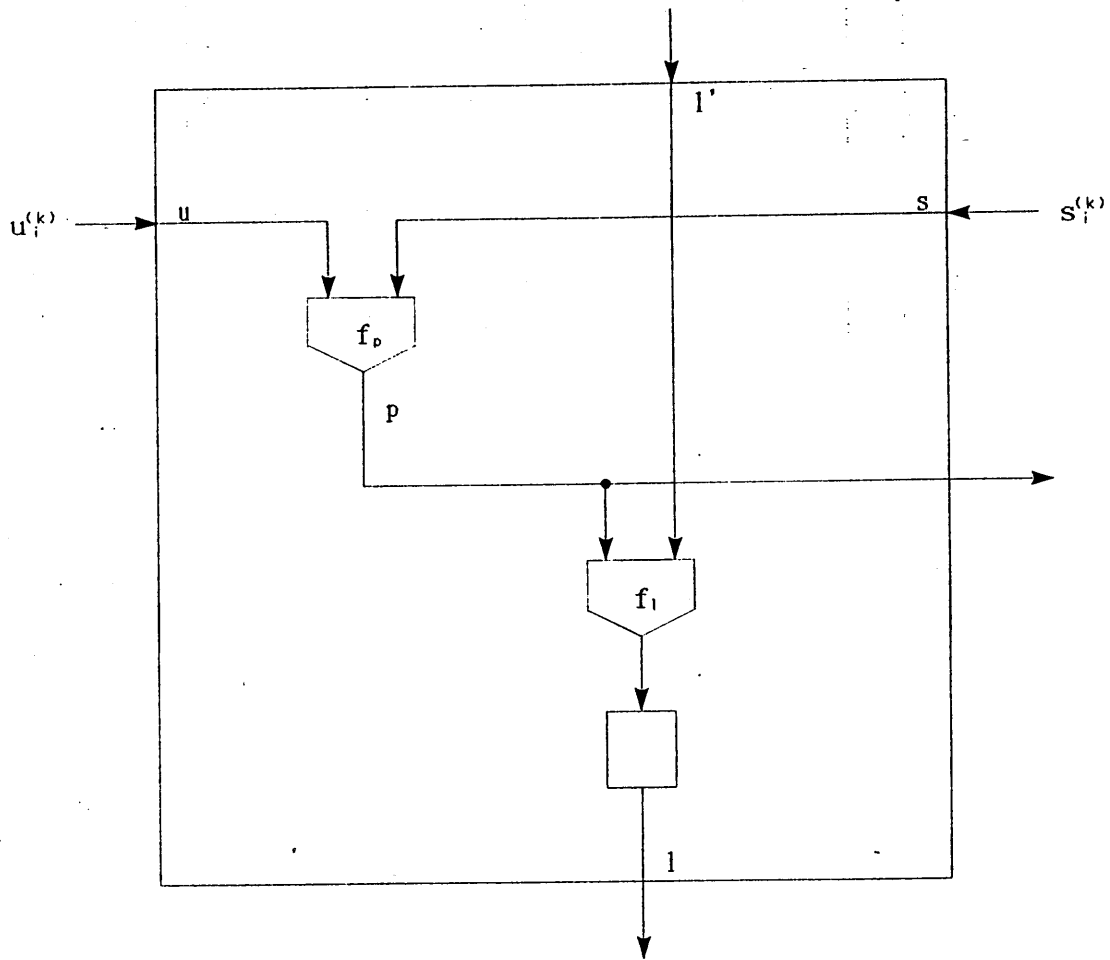


図 3.4 確率を修正するセル構造回路網



$l$  : 次のセルに送る  $\lambda^{-1}$  の値

$u$  :  $u_i^{(k)}$  の値

$s$  :  $s_i^{(k)}$  の値

各関数の意味

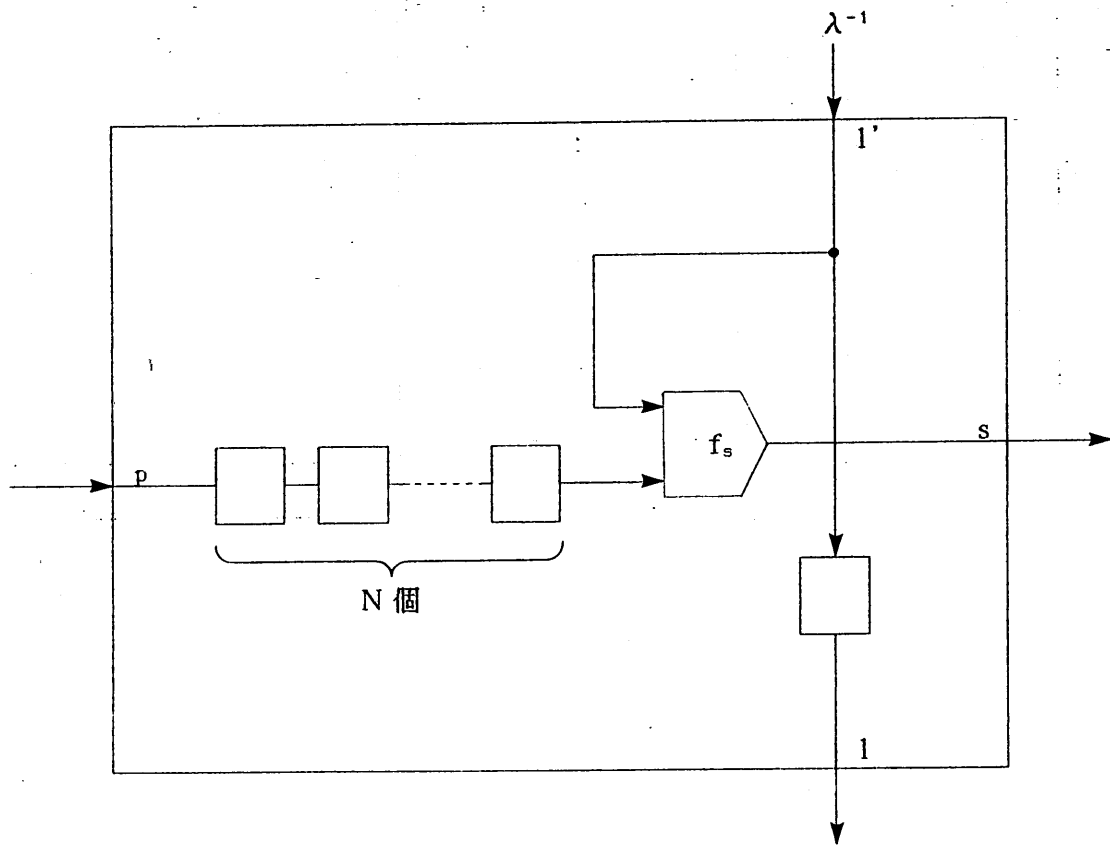
$f_p$  :  $p = s \times u'$

$f_l$  :  $l = p + l'$

[  $N$  : 各グループ内の候補の個数 ]

[  $\lambda$  : 正規化関数 ]

図 3.5 正規化関数計算セル



$l$ : 次のセルに送る  $\lambda^{-1}$  の値  
 $p$ :  $u_i^{(k)} \times s_i^{(k)}$  の値  
 $s$ : 修正された  $s_i^{(k)}$  の値

関数の意味

$$f_s : s = p / l'$$

[  $\lambda$ : 正規化関数 ]

図 3.6 確率修正セル



## 第4章 結論

本研究では、シストリックアルゴリズムを用いて文字認識を高速に実行するセル構造回路網を構成することを目的とし、いくつかの検討を行なった。本研究をまとめると次のようになる。

(1) パターン整合法による文字認識アルゴリズムとして連想整合法を取り上げ、文字認識を並列に実行するセル構造回路網を構成した。処理の並列化により、計算上、逐次処理と比較して $1/26 \sim 1/265$ の処理時間で文字を認識できることがわかった。

(2) 構造解析法への応用の準備として、確率的弛緩法を用いた簡単な問題である重み付きグラフの問題を取り上げ、この問題を解くセル構造回路網を構成した。グループ数を $M$ 、各グループ内のノードの数を $N$ とすると、計算上、処理の並列化により、逐次処理と比較して $1/MN$ の時間で処理が完了することがわかった。

## 謝辞

本研究を行なうにあたり、数多くの御指導を賜りました木村正行教授、阿曾弘具助教授に深く感謝の意を表わします。

また、日頃、数多くの御助言、御討議を頂いた孫寧氏、八代博昭氏、瀧本英二氏、田原透氏、中村憲司氏に心から感謝致します。

最後に、御討論・御協力頂いた木村研究室の皆様に感謝します。

参考文献

- [ 1 ] 阿曾弘具：セル構造情報処理系とプログラミング  
昭和62年度電気関係学会東海支部連合大会
- [ 2 ] 孫寧：活字漢字の高速認識に関する研究  
東北大学大学院工学研究科情報工学専攻 修士学位論文 昭和63年度
- [ 3 ] 江島、木村：弛緩法における修正作用素の検討  
信学論 ( D ) Vol. J 68 - D No. 12
- [ 4 ] 江島、木村：ストローク間の関係を基にした文字認識アルゴリズム  
知識工学と人工知能 35 - 1 ( 1984. 7. 12 )

発表論文リスト

- [ 1 ] 大町、孫、阿曾、木村：印刷文字認識の並列処理に関する基礎研究  
電気情報通信学会 春季全国大会 1988年3月 (発表予定)

付録 候補選出に要する比較回数の検討

得点の与えられた  $N$  個の候補の中から、得点の高い  $K$  個 ( $K \leq N$ ) の候補を選出するのに必要な比較回数の平均値 (期待値) を求める。ただし、 $N$  個の候補の得点は一様に分布しているものとする。

以下、 $N$  個の候補のうち  $i$  番目の候補を  $c_i$  と表わし、候補  $c_i$  を処理する場合について考えてみる ( $K + 1 \leq i \leq N$ )。このときすでに、 $c_{i-1}$  までは処理が終了している。

この時点で選ばれている上位  $K$  個の候補を、得点の高い順に  $a_1, a_2, \dots, a_K$  とする。 $c_i$  は、まず  $a_K$  と比較される。もし  $c_i$  の得点の方が  $a_K$  の得点より低ければ、それで処理は終了する。比較回数は 1 回である。逆に  $c_i$  の得点の方が高ければ、次に  $a_{K-1}$  の得点と比較される。もし  $c_i$  の得点の方が低ければ、それで処理は終了し、比較回数は 2 回である。 $c_i$  の得点の方が高ければ、順に  $a_{K-3}, \dots, a_1$  と比較されていくことになる。比較回数をまとめると、次のようになる。(ただし、 $a_j, c_i$  の得点をそれぞれ  $a_j, c_i$  と表わしている。)

- ①  $c_i \leq a_K$  のとき、比較回数は 1 回
- ②  $a_{j+1} < c_i \leq a_j$  ( $2 \leq j \leq K - 1$ ) のとき、比較回数は  $(K + 1 - j)$  回
- ③  $a_2 < c_i$  のとき、比較回数は  $K$  回

いま、①～③が成り立つ確率をそれぞれ求めてみる。得点の分布は一様であると仮定しているので、 $(i - 1)$  個の候補の得点の分布は理想的には図 a. 1 のよ

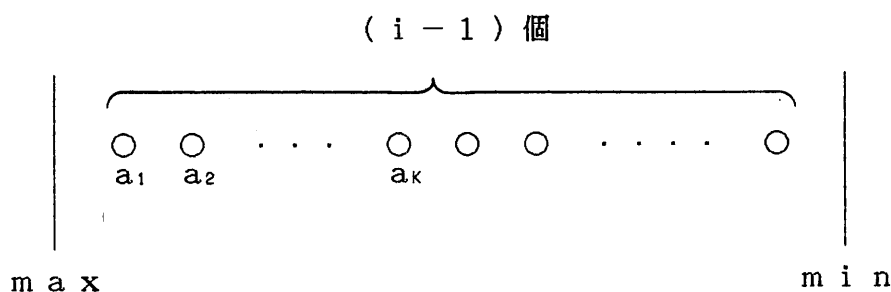


図 a. 1 候補の分布 ( $K + 1 \leq i \leq N$ )

うに均等であると考えられる（○が候補  $c_1 \sim c_{i-1}$  を表わしている）。ただし、 $\max$ ,  $\min$  は、 $c_i$  ( $1 \leq i \leq N$ ) のとり得る得点の最大値、最小値である。図より、 $a_1 < c_i \leq \max$ ,  $a_2 < c_i \leq a_1$ ,  $a_3 < c_i \leq a_2$ ,  $\dots\dots$  となる確率は理想的にはすべて  $1/i$  である。従って、

①  $c_i \leq a_k$  となる確率は  $(i - K) / i$

②  $a_{j+1} < c_i \leq a_j$  ( $2 \leq j \leq K - 1$ ) となる確率は  $1 / i$

③  $a_2 < c_i$  となる確率は  $2 / i$

であるから、比較回数の期待値は、

$$\begin{aligned}
 & 1 \times \frac{i - K}{i} + \sum_{j=2}^{K-1} \left\{ (K + 1 - j) \times \frac{1}{i} \right\} + K \times \frac{2}{i} \\
 &= 1 + \sum_{j=1}^{K-1} \left\{ (K + 1 - j) \times \frac{1}{i} \right\} \\
 &= 1 + \left\{ (K + 1) (K - 1) - \frac{K (K - 1)}{2} \right\} \times \frac{1}{i} \\
 &= 1 + \frac{K^2 + K - 2}{2i} \quad \text{回} \qquad \qquad \qquad (a. 1)
 \end{aligned}$$

となる。

次に、 $2 \leq i \leq K$  の場合に候補  $c_i$  を処理することを考えてみる。

このときのデータの配置は、 $K + 1 \leq i \leq N$  のときと同様に考えると、理想的には図 a. 2 のようになる。  $c_i$  を処理する際の比較回数は、

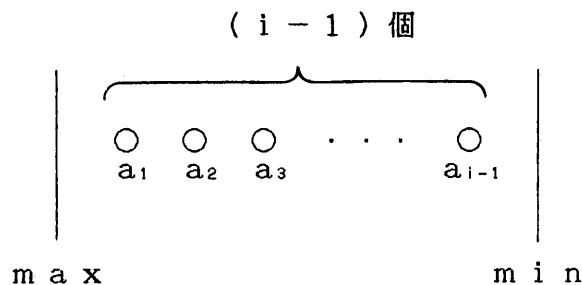


図 a. 2 候補の分布 ( $1 \leq i \leq K$ )

①  $\min \leq c_i \leq a_{i-1}$  のとき、1回

②  $a_{j+1} < c_i \leq a_j$  ( $2 \leq j \leq i-2$ ) のとき、 $(i-j)$ 回

③  $a_2 < c_i$  のとき、 $(i-1)$ 回

となる。①～③が成り立つ確率も同様に求めると、

①  $\min \leq c_i \leq a_{i-1}$  となる確率は  $1/i$

②  $a_{j+1} < c_i \leq a_j$  ( $2 \leq j \leq i-2$ ) となる確率は  $1/i$

③  $a_2 < c_i$  のとき、となる確率は  $2/i$

となるから、比較回数の期待値は、

$$\begin{aligned} & 1 \times \frac{1}{i} + \sum_{j=2}^{i-2} \left\{ (i-j) \times \frac{1}{i} \right\} + (i-1) \times \frac{2}{i} \\ &= \sum_{j=1}^{i-1} \left\{ (i-j) \times \frac{1}{i} \right\} + \frac{i-1}{i} \\ &= i-1 + \frac{i(i-1)}{2} \times \frac{1}{i} + \frac{i-1}{i} \\ &= \frac{i^2 + i - 2}{2i} \quad \text{回} \end{aligned} \tag{a.2}$$

となる。

また、候補  $c_i$  を処理するときは、最初のデータであり、比較する対象が存在しないので比較回数は0回である。これは、(a.2)式で  $i=1$  とおいたものに等しい。

従って、 $N$ 個の中から  $K$ 個の候補を選出するのに必要な比較回数の期待値は、

(a.1), (a.2)式より、

$$\begin{aligned} & \sum_{i=1}^K \left( \frac{i^2 + i - 2}{2i} \right) + \sum_{i=K+1}^N \left( 1 + \frac{K^2 + K - 2}{2i} \right) \\ &= \sum_{i=1}^K \left( \frac{i}{2} + \frac{1}{2} - \frac{1}{i} \right) + (N-K) + \frac{K^2 + K - 2}{2} \left( \sum_{i=1}^N \frac{1}{i} - \sum_{i=1}^K \frac{1}{i} \right) \\ &= \frac{K(K+1)}{4} + \frac{K}{2} - \sum_{i=1}^K \frac{1}{i} + (N-K) \\ & \quad + \frac{K^2 + K - 2}{2} \sum_{i=1}^N \frac{1}{i} - \frac{K^2 + K - 2}{2} \sum_{i=1}^K \frac{1}{i} \\ &= N + \frac{K(K-1)}{4} + \frac{K^2 + K - 2}{2} \sum_{i=1}^N \frac{1}{i} - \frac{K(K+1)}{2} \sum_{i=1}^K \frac{1}{i} \quad \text{回} \end{aligned} \tag{a.3}$$

となる。  $K=5$  とすれば、

$$N - 29.25 + 14 \sum_{i=1} \frac{1}{i} \quad \text{回} \quad (\text{a. 4})$$

である。N が大きいとき、これは

$$N - 30 + 14 \log_e N \quad \text{回} \quad (\text{a. 5})$$

と近似できる。