



# 境界探索法による２値画像ラベリングの 実時間処理回路の検討

指導教官

鈴木正國教授

北川章夫助教授

秋田純一助手

提出者

金沢大学 工学部

電気・情報工学科

渡辺 晃

平成11年2月24日

# 目次

<b>1</b>	<b>はじめに</b>	<b>4</b>
<b>2</b>	<b>アルゴリズムの検討</b>	<b>5</b>
2.1	構成 . . . . .	5
2.2	色抽出 . . . . .	6
2.3	ラベリングのアルゴリズム . . . . .	8
2.3.1	ラベリング . . . . .	8
2.3.2	2-pass 方式 . . . . .	8
2.3.3	境界探索法 (1-pass 方式) . . . . .	9
2.4	アクセス回数の検討 . . . . .	10
2.4.1	実験方法 . . . . .	10
2.4.2	サンプル画像 . . . . .	11
2.4.3	シミュレーション結果と考察 . . . . .	13
<b>3</b>	<b>回路設計</b>	<b>14</b>
3.1	デザインフロー . . . . .	14
3.2	設計仕様 . . . . .	15
3.2.1	特徴 . . . . .	15
3.2.2	入力/出力ポート . . . . .	16
3.2.3	タイミングチャート . . . . .	18
3.2.4	各ユニットの動作 . . . . .	20
3.2.5	各ユニットの内部ステート . . . . .	21

3.3	論理合成 . . . . .	24
3.4	配置配線 . . . . .	24
3.5	動作シミュレーションと考察 . . . . .	26
4	まとめ	27
	謝辞	28

# 第 1

## はじめに

近年集積回路における高速化の発展はめざましく、計算機の高速化にともなって画像処理に関する研究が盛んに行われるようになってきている。従来は計算機を用いたソフトウェア的なアプローチによる研究が中心であったが、現在の計算機の高速化により多くの画像処理が短時間で行えるようになってきた。しかし、それでもなおリアルタイムでのアプリケーションには処理時間がネックとなるケースが多いのが現状である。

2値画像情報に対して有意画素のつくる連結領域を認識し、各領域の重心や面積を計算するという処理にはラベリングという手法が用いられる。また現在種々のラベリングのアルゴリズムが考案されており、ソフトウェア的なアプローチによるラベリングに関する研究は多数報告されている。しかし一般に画像情報は情報量が大きいため、ロボットビジョン等の実時間で処理を行う必要があるアプリケーションでは、ハードウェア的なアプローチにより高速にラベリングを行うシステムが要求される。

本研究ではリアルタイムで重心・面積計算を行う処理回路を効率の良いアルゴリズムを用いて、ハードウェアで実現することを検討する。

## 第 2

# アルゴリズムの検討

## 2.1 構成

図 2.1 は視覚情報処理システムの全体の構成である。各ブロックは以下の役割を受け持っている。

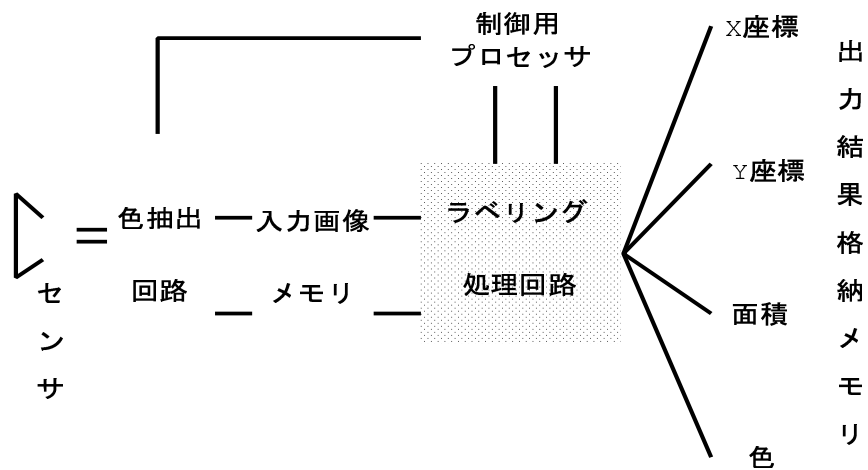


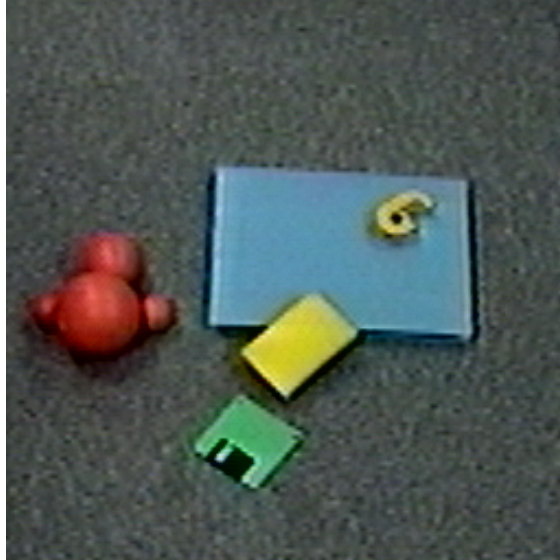
図 2.1: システム構成

- 画像センサ … 実画像を取り込んで色抽出回路に信号を送る。
- 色抽出回路 … 取り込まれた濃淡画像にしきい値処理を施して特定の色成分のみを抽出し、入力画像メモリに書き込む。抽出する色数は4色を想定している。
- 入力画像メモリ … 色抽出回路により抽出された4色および背景色を含めた5値の情報が格納されている。ラベリングの際のラベル番号の書き込みもこのメモリに対して行われる。
- ラベリング処理回路 … 実際にラベリング処理を行い、各領域の重心・面積を計算し領域の色を調べ、その結果である領域の x 座標の総和、y 座標の総和、面積、色情報を出力段のメモリに格納する。
- 演算結果格納メモリ … ラベリング処理回路の演算結果である各領域の x 座標の総和、y 座標の総和、面積の総和、色情報をそれぞれ格納する。
- 制御用プロセッサ … 色抽出回路やラベリング処理回路など、システム全体の制御を行う。

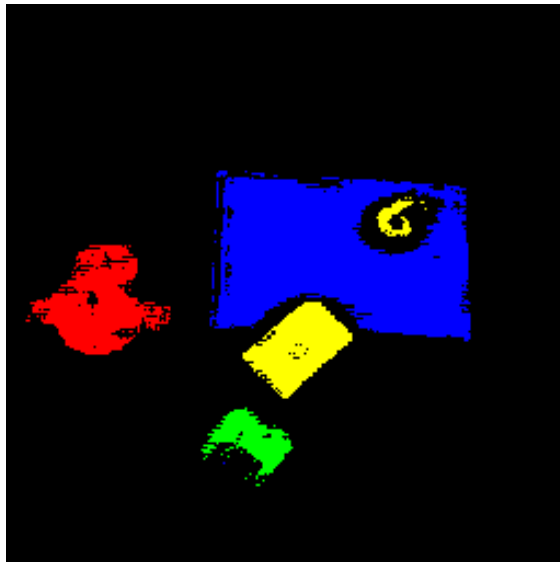
本研究ではこのシステムを中心となるラベリング及び重心・面積計算処理回路について検討し、設計を行う。

## 2.2 色抽出

入力画像として、図 2.2 のような4色に色抽出された画像を用いる。この処理は入力されたカラー画像をその成分毎にしきい値処理して行われる。こうして色抽出された画像は入力画像メモリに書き込まれる。ここでは、0が背景を表し、-1～-4が各色を表すことにする。こうすることでラベルを1から順に与えることができる。



(a) 実画像



(b) 色抽出後

図 2.2: 実画像と色抽出後の画像



## 2.3 ラベリングのアルゴリズム

### 2.3.1 ラベリング

ラベリング (ラベル付け) とは図 2.3 に示すように異なる連結領域にそれぞれ異なるラベルを割り当てる処理のことをいう。この処理を行うことによって画素のつくる連結領域を区別することができ、領域毎の面積や重心の座標といった情報が得られるようになるのである。また、連結成分の定義には 4 近傍、8 近傍等があるが、本研究では 4 近傍法を用いる。その理由は回路規模を小さくしたいことと、処理時間を短縮したいためである。

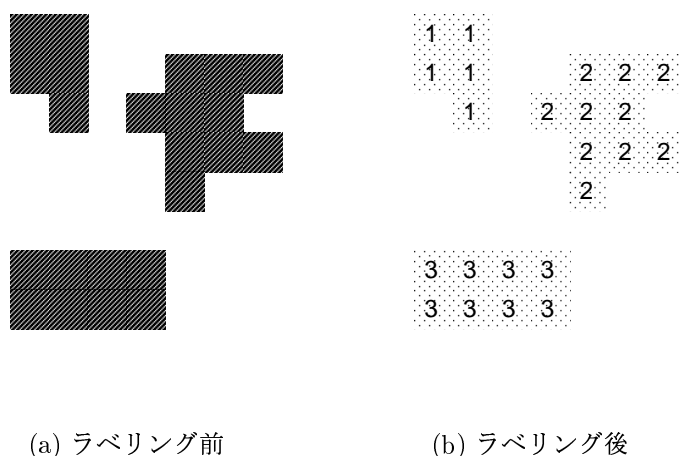
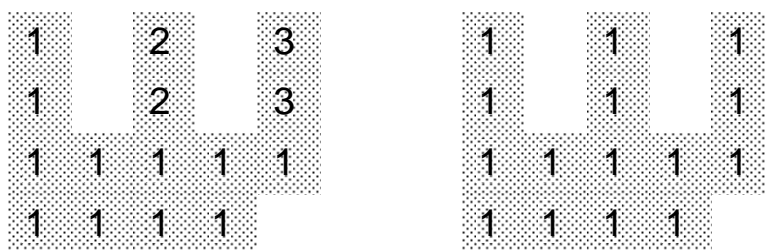


図 2.3: ラベリング

### 2.3.2 2-pass 方式

ラベリングのアルゴリズムとしてよく知られているものに 2-pass 方式がある。これは全画素に対する逐次走査を 2 回行うことでラベル付けをするものである。図 2.4 にこのアルゴリズムの処理を簡単に示す。まず最初の走査で仮のラベルを与えておき (図 2.4(a))、後から得られた連結情報は対応テーブルに記録しておく。そして 2 回目の走査

でこのテーブルを参照しながらラベルを書き換えていくのである (図 2.4(b))。この方法では、走査回数は入力画像の内容に影響を受けにくいものの、どんな画像に対しても必ず2回の全画素走査が行われるため、走査する画素数は総画素数に比例して増加していくという問題がある。



(a) 1 回目の走査

(b) 2 回目の走査

図 2.4: 2-pass のアルゴリズム

### 2.3.3 境界探索法 (1-pass 方式)

2 回の逐次走査を行う 2-pass 方式に対し、1 回の逐次走査でラベリングを行うアルゴリズムも知られている。ここでは境界探索 (追跡) 法もしくは 1-pass 方式と呼ばれるアルゴリズムを検討する。このアルゴリズムを図 2.5 に示す。

まず、左上から逐次走査を始める。画素が背景色 (0) の間は何もせずに次の画素へと進む。そして、有意領域にさしかかったらそこを起点に反時計廻りに境界探索を開始し、左側のエッジにラベルを与える (図 2.5(a))。このようにして 1 周してきたら再び逐次走査に戻る。逐次走査でラベル付けされていない画素を見つけたとき、左隣の画素に既にラベルが与えられていたならばそのラベルを現在走査中の画素にも与える (2.5(b)) に境界探索法はこのようにしてラベル付けを行う。このアルゴリズムの長所としては、逐次走査が 1 回で済むこと、したがって背景部分が多い画像など単純な画像では走査回

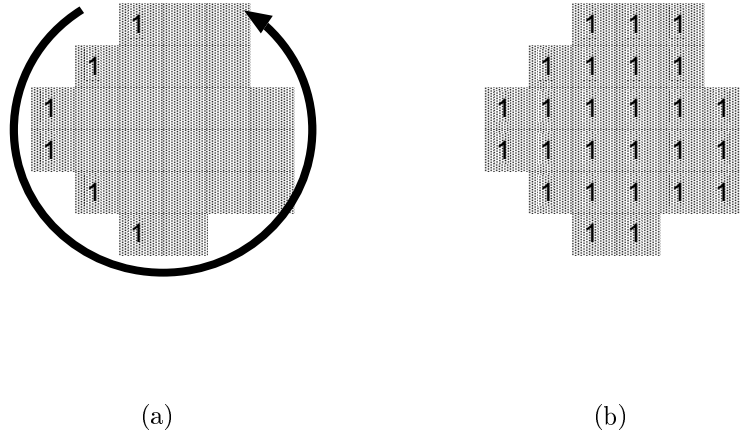


図 2.5: 境界探索法のアルゴリズム

数が非常に少ないこと、ラベルの対応テーブルが不要のため回路規模が大きくなることなどが挙げられる。これらの特徴から、ラベリングのアルゴリズムとして境界探索法の採用を検討することにする。一方短所としては、複雑な領域構成を持つような画像では走査回数が膨大になる可能性があることである。この問題を確認するためにシミュレーションを行った。

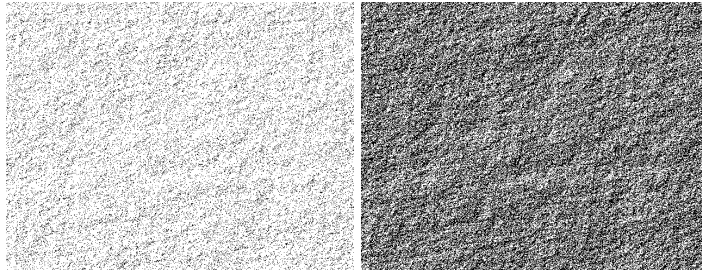
## 2.4 アクセス回数の検討

### 2.4.1 実験方法

設計に先立ち、処理回路の動作速度に最も大きな影響を与えると思われるメモリへのアクセス回数について検討した。これは、メモリアクセスに要する時間は処理回路内部の演算時間に比べてはるかに長いためである。入力画像としてランダムに作製したサンプル画像を多数用意して、計算機上でシミュレーションを行った。

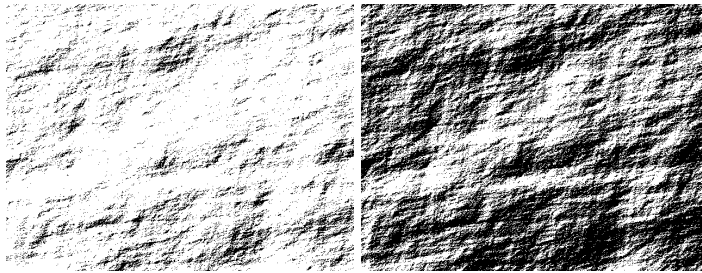
## 2.4.2 サンプル画像

サンプルに用いた画像のサイズは  $640 \times 480$  である。また、簡単のために領域と背景のみしか存在しない 2 値画像を用いた。パラメータとして用いたのは、有意領域の占める割合である  $p$ 、空間周波数  $f$  に対してパワースペクトルが  $1/f^n$  に比例するとしたときの高周波成分をあらわすパラメータである  $n$  の 2 つである。 $n$  は感覚的には画像のなめらかさを表し、0 に近づくにつれてホワイトノイズの画像になっていき、通常の画像では 1 ~ 2 程度となることが知られている。図 2.6 にランダムに作製したサンプルの一部を示す。 $p$  が大きくなると有意画素の割合が大きくなり、 $n$  が大きくなるにつれて領域が固まって存在するようになることがわかる。



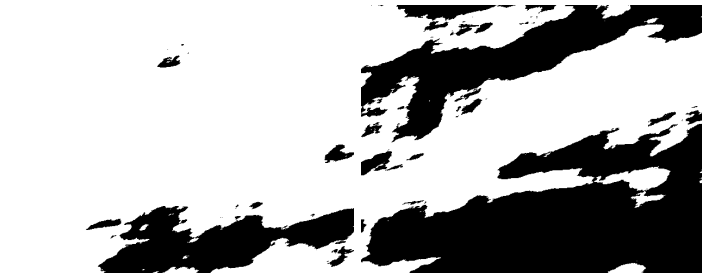
(a)  $p = 0.1, n = 0.4$

(b)  $p = 0.5, n = 0.4$



(c)  $p = 0.1, n = 1.0$

(d)  $p = 0.5, n = 1.0$



(e)  $p = 0.1, n = 2.0$

(f)  $p = 0.5, n = 2.0$

図 2.6: サンプル画像

### 2.4.3 シミュレーション結果と考察

結果を図 2.7 に示す。z 軸はアクセス回数を総画素数  $640 \times 480 = 307,200$  で割ったものである。この結果をみると、入力画像によっては画素数の 4 倍位のアクセス回数になるが、これはかなり特殊な画像を入力した場合である。色抽出を行った一般の画像では特定の色成分が大きな割合を占めるとは考えにくく ( $p \rightarrow$  小)、また、領域はある程度固まって存在する ( $n \rightarrow$  大) と考えられるので、一般にはアクセス回数はそれほど大きくはならないと思われる。また、図 2.2 の色抽出後の画像に対しシミュレーションを行ったところ、総画素数  $256 \times 256 = 65,536$  に対しアクセス回数は 79,198 回 (約 1.2 倍) であり、アクセス回数の極端な増加は見られなかった。このように、通常の画像ではアクセス回数はそれほど大きくならないことが示されたので、ラベリングのアルゴリズムとして境界探索法を用い、回路の設計を行うことにする。

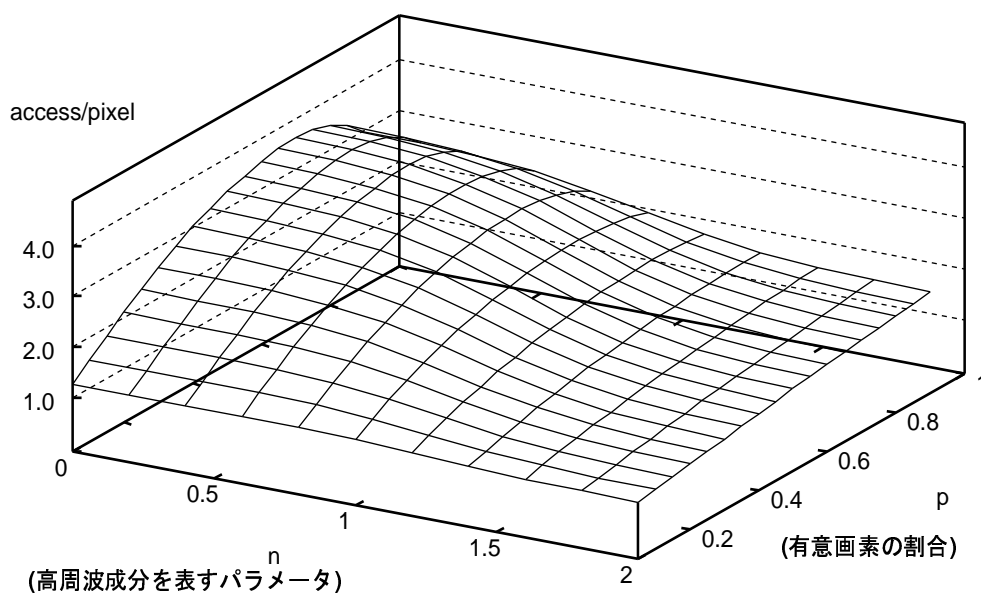


図 2.7: メモリアクセスの回数

## 第 3

# 回路設計

### 3.1 デザインフロー

以下に Verilog-HDL(*Hardware Description Language*: ハードウェア記述言語) を用いたトップダウン設計の流れを簡単に示す。

各段階での主な作業は以下の通りである。

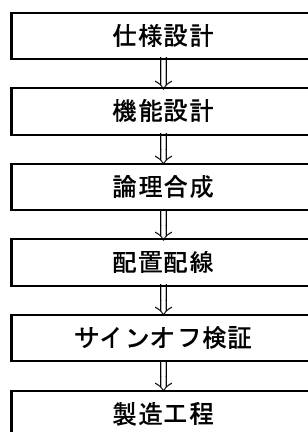


図 3.1: トップダウン設計の流れ

- 仕様設計 … システム全体に対する仕様を検討する。要求される機能や特性を検討する。
- 機能設計 … アーキテクチャを検討し回路を機能別のモジュールに分割して、各ユニットの動作をHDLで記述する。また、テストパターンを作製しシミュレーションにおいて所望の結果が得られているか確認する。
- 論理合成 … 制約条件を与えて動作記述をゲートレベルに変換し、論理最適化によって不要なゲートを除去する。合成後にはゲート遅延を含めたシミュレーション(タイミング解析)を行い動作を確認する。
- 配置配線 (レイアウト設計) … 合成後のネットリストを用いてチップ内でのセルの配置、信号や電源・グラウンドの接続、詳細の配線を行う。
- サインオフ検証 … 最終段階での検証。配線情報も考慮して動作を調べる。また、デザインルールチェックを行い、製造工程へ渡すための最終的なデータを作製する。
- 製造工程 … V D E C に設計データを提出し、半導体ベンダにてチップを製造する。

なお、本処理回路はロームの4.5mm角チップ(CMOS 0.6 $\mu$ mルール, 3層メタル)で設計する。

## 3.2 設計仕様

前章では入力画像として640×480画素を想定し、アクセス回数が大きな問題にはならないという結論に達した。また、設計当初はラベル数も8ビット(約250まで)を予定していた。しかし、現仕様では今回試作するチップの信号ピン数の上限を越えてしまうことがわかった。そこでいくつかの工夫をした。

### 3.2.1 特徴

**入力画像のサイズ** 設計当初の予定より若干縮小し、256×256画素とする。

**色数** 4色と背景色の合わせて5色とする。



**ラベル数** 7ビットとし、最大 123 までにした。(124 ~ 127 は、まだラベル付けされていない色を表すのに用いる。)

**重心の座標** 重心の座標は、領域内の画素の座標の和を面積で割って求められる。しかし、除算回路を含めると回路規模が大きくなりすぎてチップ内に収まらない可能性が高いので、除算は行わずに x 座標、y 座標の和を出力する。x 座標、y 座標それぞれの総和は 20bit とするが、座標の和はかなり大きな値になりうることが予想されるので、下位 4 ビットを切捨てることにした。ただ、誤差の累積を防ぐため下位ビットは処理回路内部にレジスタとして持たせることにした。この結果、x 座標、y 座標の和はそれぞれ 1048576 までカウントできる。

**オーバフロー** ラベル数、x 座標、y 座標の和、面積のいずれがあふれてもオーバフロー信号を出力するようにした。

**孤立点の除去** ノイズ等によって発生する孤立点 (連結成分を持たない面積 1 の点) の除去も同時に行うことにする。

### 3.2.2 入力／出力ポート

図 3.2 に入力／出力ポートの一覧を示す。

#### 入力信号

- CLK … メインクロック
- RESET … 1 のときリセット

#### 入出力信号

- DATA\_BUS\_IN[6:0] … 入力画像メモリのデータバス
- DATA\_BUS\_GX[15:0] … 重心の x 座標用メモリのデータバス
- DATA\_BUS\_GY[15:0] … 重心の y 座標用メモリのデータバス
- DATA\_BUS\_S[13:0] … 面積用メモリのデータバス

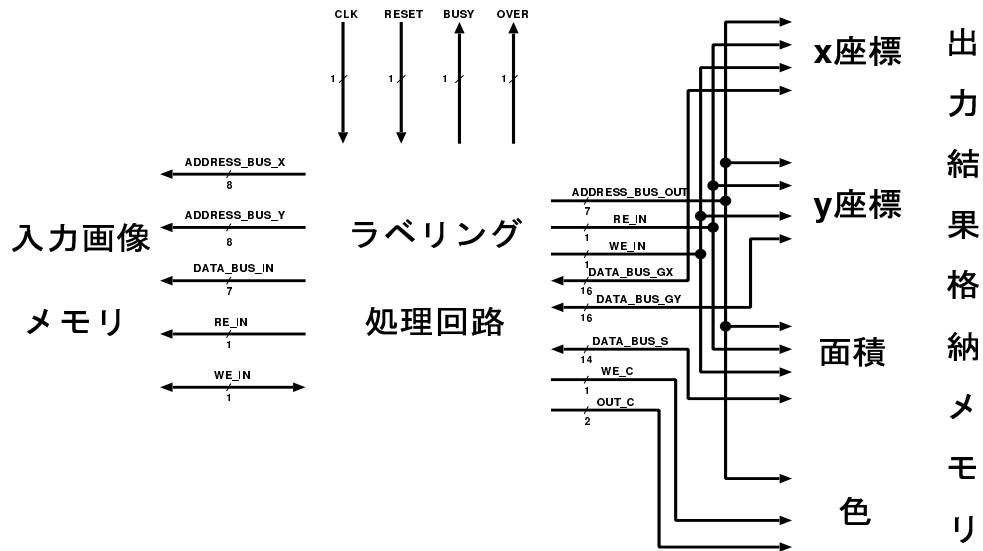


図 3.2: 入力／出力ポート

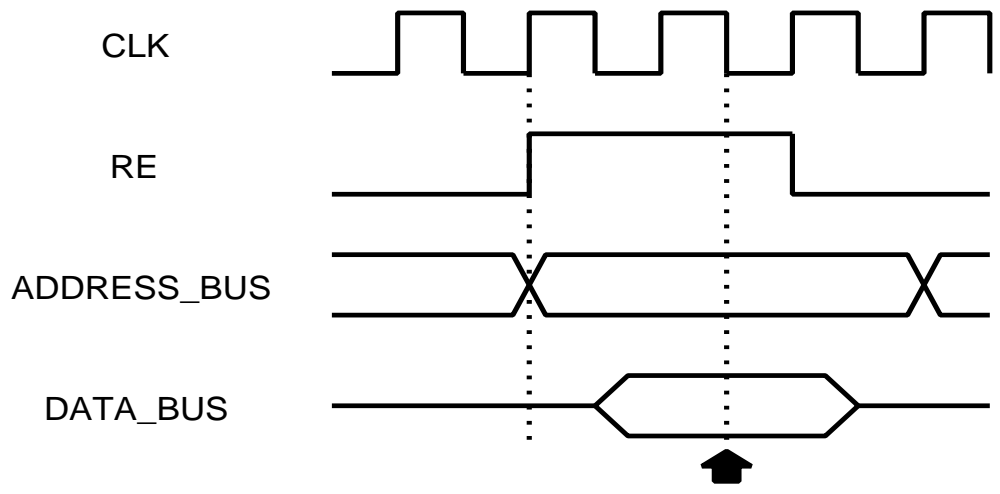
#### 出力信号

- OUT\_C[1:0] … 色情報用メモリのデータバス (出力のみ)
- ADDRESS\_BUS\_X[7:0] … 入力画像メモリのアドレスバス (X)
- ADDRESS\_BUS\_Y[7:0] … 入力画像メモリのアドレスバス (Y)
- ADDRESS\_BUS\_OUT[7:0] … 演算結果格納用メモリのアドレスバス
- RE\_IN … 入力画像メモリに read するとき 1
- WE\_IN … 入力画像メモリに write するとき 1
- RE\_OUT … 重心・面積用メモリに read するとき 1
- WE\_OUT … 演算結果格納用メモリに write するとき 1
- WE\_C … 色情報用メモリに write するとき 1

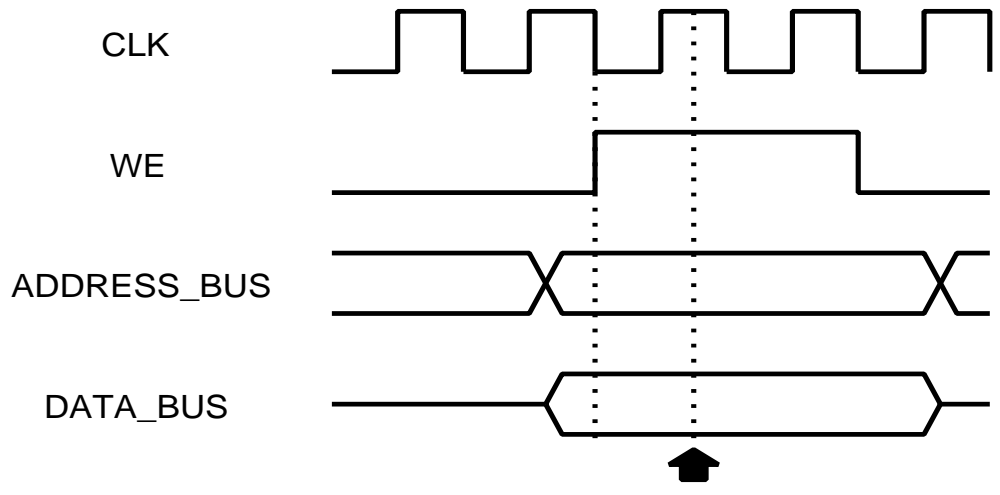
- BUSY … 本処理回路が演算中のとき 1
- OVER … 重心・面積・ラベル数のいずれかがあふれたとき 1

### 3.2.3 タイミングチャート

- 本処理回路を構成する各ユニットはクロックに同期して状態遷移するステートマシンとなっている。
- read enable 信号 (RE\_IN, RE\_OUT) はクロックの立ち下がりエッジに同期して出力される。
- write enable 信号 (WE\_IN, WE\_OUT, WE\_C) はクロックの立ち上がりエッジに同期して出力される。
- データバス (DATA\_BUS\_IN, DATA\_BUS\_GX, DATA\_BUS\_GY, DATA\_BUS\_S) の値は read enable が 1 になってから 1.5 クロック後に取り込まれる。
- read enable 信号が 1 になってから 1.5 クロック以内にデータバスの値が安定していなければならない。



(a) read 時のタイミング



(b) write 時のタイミング

図 3.3: タイミングチャート

### 3.2.4 各ユニットの動作

設計の能率を向上させるため回路を機能別に分割した。処理回路のブロック図を図 3.4 に示す。各ユニットの主な役割は以下の通り。

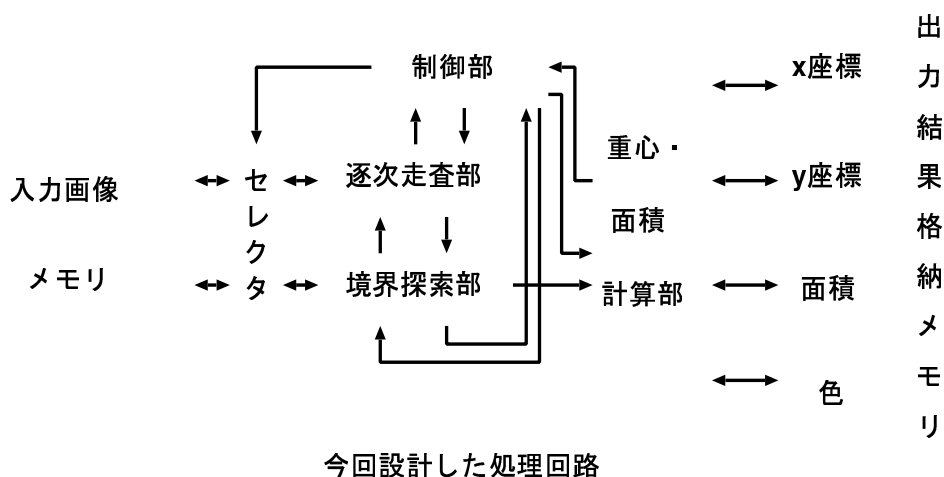


図 3.4: ブロック図

- 制御部 … 制御用プロセッサからのクロック、リセット信号を受け各ユニットの駆動や停止等の制御、BUSY 信号やオーバーフロー信号の出力等を行う。
- 逐次走査部 … 有意画素が見つかるまで、左上から順に逐次走査を行う。
- 境界探索部 … 有意画素が見つかったとき、ラベルや色情報の判定を行い、領域の端点であれば境界探索を行ってラベルを書き込む。この処理を領域を 1 周するまで行う。
- 重心・面積計算部 … 境界探索部のラベルの書き込みに合わせて重心の座標の加算や面積のインクリメント、色情報の書き込みを行う。また、出力情報の下位 4 ビットを格納するレジスタを含んでいる。
- セレクタ … 入力段のメモリにアクセスするユニット (逐次走査部、境界探索部) を制御部の信号により切り替える。

### 3.2.5 各ユニットの内部ステート

今回設計した回路は、その多くがステートマシンとなっている。以下に各ユニットの状態遷移図を示す。

#### 制御部

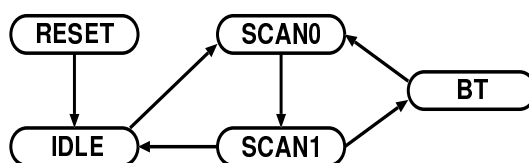


図 3.5: 制御部の状態遷移図

- RESET … 初期化を行う。
- IDLE … 待機状態。
- SCAN0, SCAN1 … 逐次走査を行う。
- BT … 境界探索を行う。

#### 逐次走査部

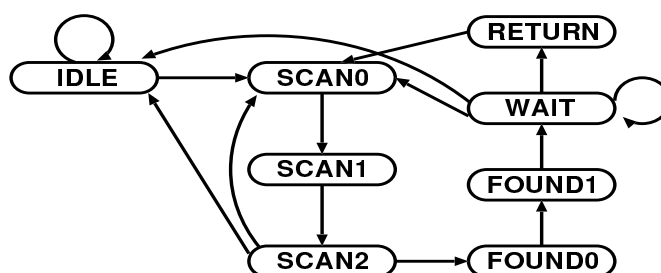


図 3.6: 逐次走査部の状態遷移図

- IDLE … 待機状態。
- SCAN0, SCAN1, SCAN2 … 画素が背景色 (0) の間、走査を続ける。
- FOUND0, FOUND1 … 背景色 (0) 以外の画素が見つかったときの処理を行う。
- WAIT … 境界探索を行っている間の中断状態。
- RETURN … 境界探索から逐次走査へ復帰する際の処理を行う。

### 境界探索部

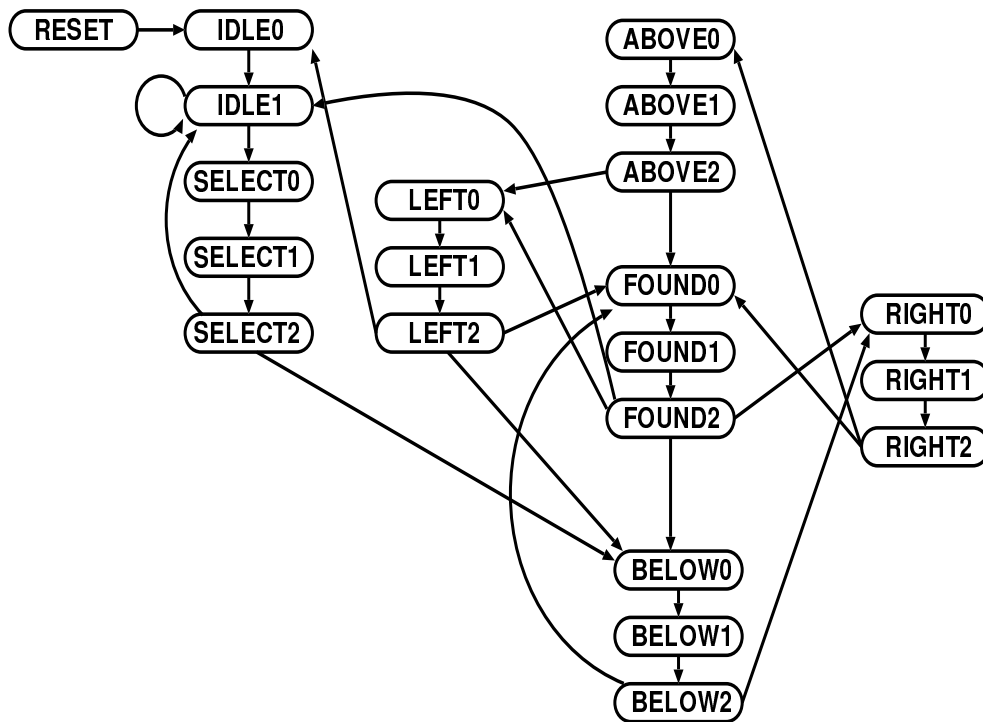


図 3.7: 境界探索部の状態遷移図

- RESET … 初期化を行う。
- IDLE0, IDLE1 … 待機状態。

- SELECT0, SELECT1, SELECT2 … 見つかった画素が領域の開始点かどうかを判定する。
- LEFT0, LEFT1, LEFT2 … 左に有意画素があるかを調べる。
- BELOW0, BELOW1, BELOW2 … 下に有意画素があるかを調べる。
- RIGHT0, RIGHT1, RIGHT2 … 右に有意画素があるかを調べる。
- ABOVE0, ABOVE1, ABOVE2 … 上に有意画素があるかを調べる。
- FOUND0, FOUND1, FOUND2 … 隣に有意画素が見つかった際の処理。ラベルの書き込みなどを行う。

#### 重心・面積計算部

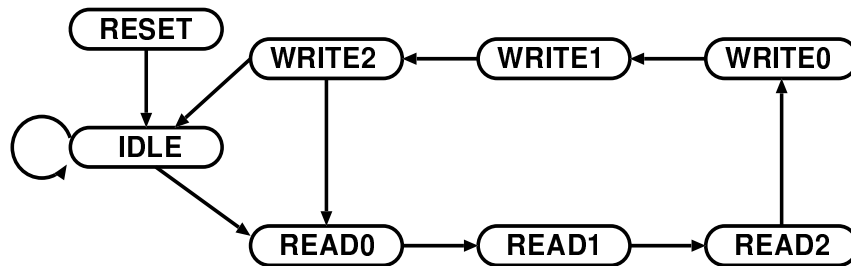


図 3.8: 重心・面積計算部の状態遷移図

- RESET … 初期化を行う。
- IDLE … 待機状態。
- READ0, READ1, READ2 … 重心の座標、面積を read する。
- WRITE0, WRITE2, WRITE3 … 重心の座標、面積をインクリメントして write する。



### 3.3 論理合成

論理合成には Synopsys 社の論理合成ツールである Design Analyzer を用いた。使用した総セル数は約 8,000 であった。合成終了後、パッドを挿入して遅延情報を考慮したタイミング解析を行い、動作を確認した。この過程では回路はゲートレベルに変換される。例として、逐次走査部のスキマティックを図 3.9 に示す。

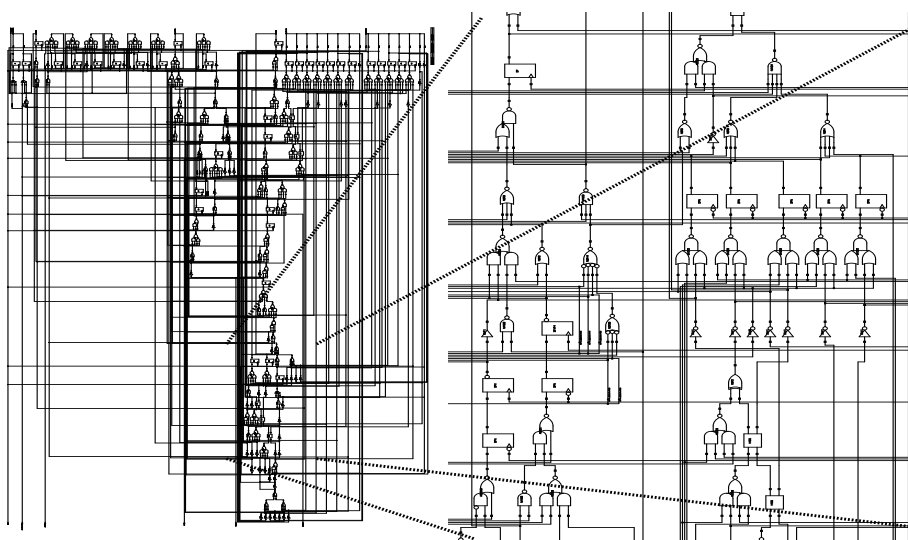


図 3.9: 逐次走査部のスキマティック

### 3.4 配置配線

配置配線には Avant!社の自動配置配線ツールである Apollo を使用した。配線終了後、DRC (*Design Rule Check*) を行い、設計ルール違反の無いことを確認した。また、最終的なチップのレイアウトを図 3.10 に示す。<sup>1</sup>

<sup>1</sup> なお、本チップ試作は東京大学大規模集積システム設計教育研究センターを通しローム (株) 及び凸版印刷 (株) の協力で行われたものである。

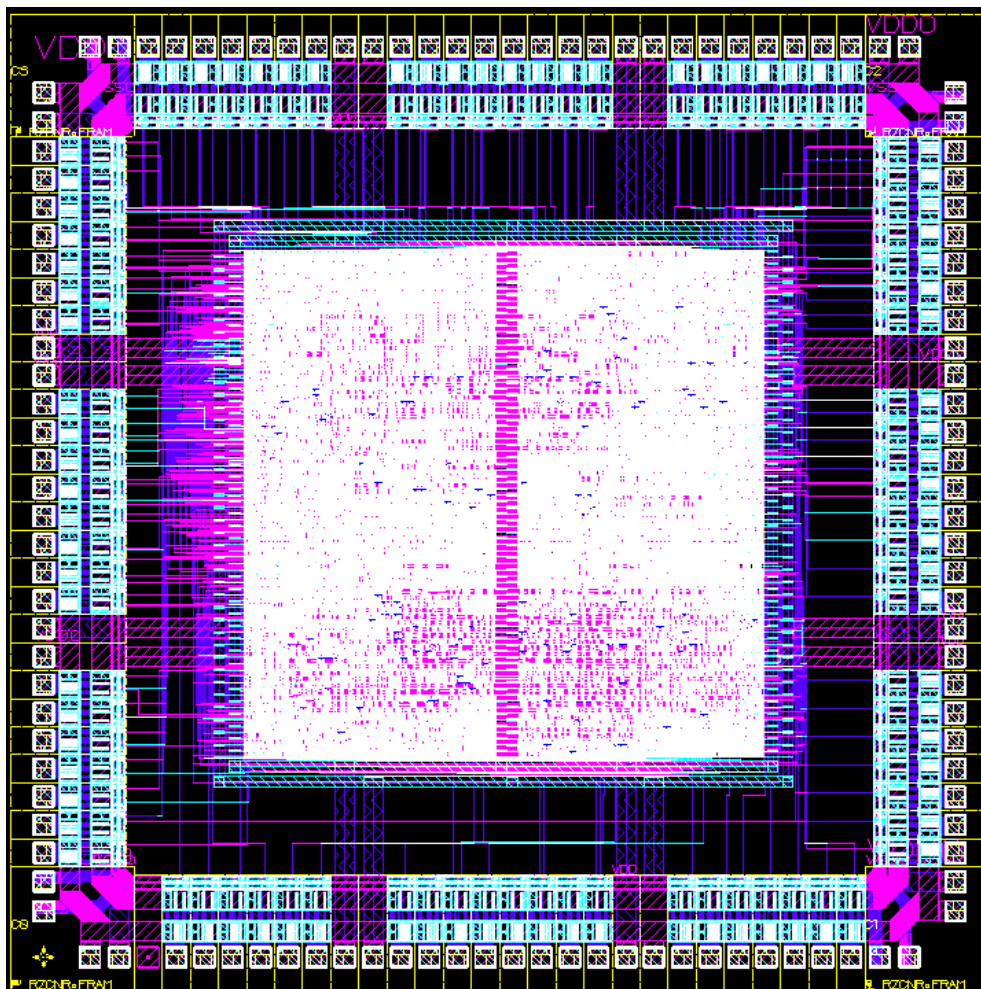


図 3.10: チップのレイアウト (ローム 4.5mm 角 CMOS0.6 $\mu$ m ルール,3層メタル)

### 3.5 動作シミュレーションと考察

論理合成を行った後のネットリストと図 2.2 のサンプル画像を用いてゲートレベルでのシミュレーションを行い、動作確認をした。本処理回路は常にメモリにアクセスしているためにその動作速度は使用するメモリの性能に大きく依存することになる。本回路の最大動作周波数は約 73MHz であったが、ここでは一般的な値を用いた。メモリのアクセス時間を 50ns としたとき、約 15ms で処理することができた。一般のビデオ画像が 30 コマ/秒 (1 画像当たり約 33ms) であることを考慮すると、この結果は、本処理回路が十分実時間処理可能であることを示すものといえる。

## 第 4

### まとめ

ロボットビジョンへの応用を目指して、実時間処理可能なラベリング処理回路の検討を行った。

アルゴリズムとして境界探索法を採用し、計算機シミュレーションによりアクセス回数の見積もりを行い、その有効性を確認した。

VDECのチップ試作サービスを利用してトップダウン設計によりラベリングLSIの設計を行った。

ゲートレベルで処理回路の動作を確認し、処理時間の見積もりを行い、実時間処理の可能性を示した。

今後は完成したLSIチップが到着次第テストを行い、動作確認を行った後、実際の視覚情報処理システムへの搭載を検討していく予定である。

## 謝辞

本研究を行うにあたり、多くの方々からの御助言、御指導を頂きました。厚く御礼申し上げます。

指導教官の鈴木正國教授には研究活動、学生生活全般において多くの御指導を頂きました。深く感謝致します。北川章夫助教授には様々な興味深い専門分野の知識を教えてくださいました。深く感謝致します。秋田純一助手には研究活動において多大な御助言、御指導を頂き、多くのことを学ぶことができました。深く感謝致します。また、研究室の行事等でお世話になりました柿本芳雄技官に深く感謝致します。

チップの試作において御協力頂いたVDEC、ローム(株)、凸版印刷(株)の関係者の皆様に深く感謝致します。

私の研究において多くの御協力を頂き、VLSI設計の道に導いて下さった大学院修士課程2年の前多和洋氏に感謝致します。また、学生生活を有意義で楽しいものにして下さった大学院博士課程のT.K.Chakraborty氏、大学院修士課程の田口和彦氏、夏目雅弘氏、水橋嘉章氏、早川史人氏、高瀬信二氏、小川明宏氏、中橋憲彦氏、行事等でお世話になりました藤井直樹氏、私を激励し、研究を応援して下さい下さった金沢大学医学部保健学科助手の中山和也氏に心から感謝致します。また、この一年間苦楽を共にし、充実した日々を共に過ごした卒研究生の今井豊君、数馬晋吾君、佐藤堅君、房川実君、藤田隼人君、水野浩樹君、村上崇君に感謝します。

最後に、大学生活を有意義なものにしてくれた友人達、あらゆる面で支えてくれた家族、御助力頂いた多くの皆様に感謝致します。

どうもありがとうございました。

## 参考文献

- [1] 鳥脇純一郎. “画像理解のためのデジタル画像処理 (II)”. 昭晃堂, 1988.
- [2] 石山豊, 久保文雄, 高橋裕信, 富田文明. “境界追跡型ラベリングボード”. 電子情報通信学会論文誌 D-II, J78-D-II, No.1, pp.69-75, 1995.
- [3] 桜井至. “HDLによるデジタル設計の基礎”. テクノプレス, 1997.