

# 第 1 章

## 序章

### 1.1 本研究の背景

画像中の領域を認識するといった画像処理システムの流れとして、現在では受光部分から得られた画像情報をメモリーに転送し、そのデータを逐次的に処理するといった方法が主流となっている。

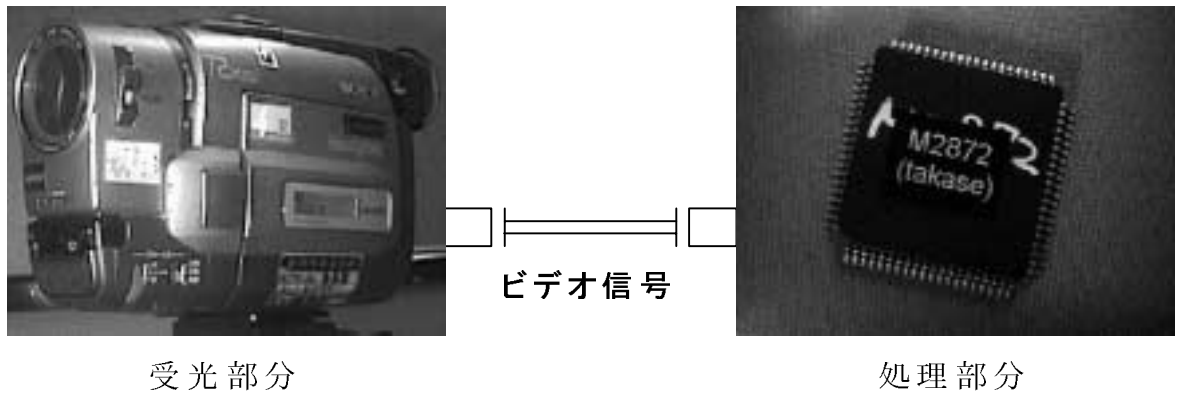


図 1.1 従来の画像処理システム

画像情報をメモリーに転送するための撮像素子には CCD イメージセンサがよく利用される。これは空間的な画像情報を含んだ二次元的な光信号を走査することにより、一次元の時間軸上の電気信号に変換するものである。この撮像素子によって、画像処理システムは急速に発展してきた。

ところが、近年の画像の高解像度化やフレーム数の増加に伴って、扱う情報量は急激に増加している。そのためロボットビジョンのように1枚の画像あたり1[ms]以下の高速なリアルタイムでの動作が望ましいとされる画像処理においては、膨大な量のデータをメモリーに転送する時間、およびそのデータを処理する時間が共に大きな問題となる。半導体技術の発展によって、処理系のデータ処理速度は飛躍的に向上してきたが、その発展にも陰りが見えはじめている近年、処理速度の向上も今後飽和状態になることは明らかである。従って更なるリアルタイム性の向上のためには、これまでの処理方法とは異なる方法を考え出さなくてはならない。

## 1.2 スマートセンサ

これらの問題を解決する方法として、集積回路技術の進歩によって受光部分に処理部分の一部を共に組み込んだ、いわゆるスマートセンサと呼ばれる回路が現実のものとなり現在広く研究が行われている[1]。スマートセンサは画素単位に受光素子と処理回路を配置するため、各画素ごとでの並列処理が可能であり、また走査がなくなるため高速なリアルタイムでの画像処理を実現できるという利点がある。その他システムの小型化などの利点も挙げられる。従来のスマートセンサの研究では、ノイズ除去やエッジ検出といった通常の画像処理の前処理部分に当たる機能を持つ処理系回路を組み込む例が多く報告されている[2,3,4,5]。以下にその一例を示しておく。

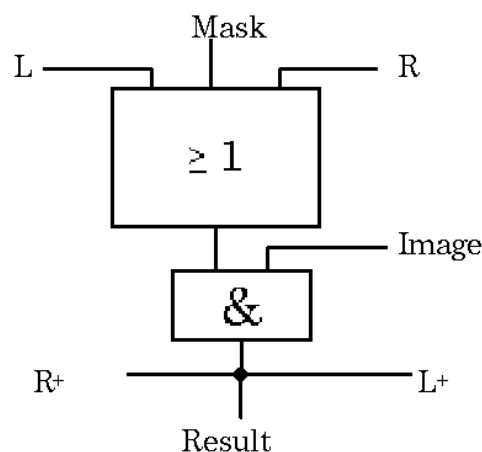
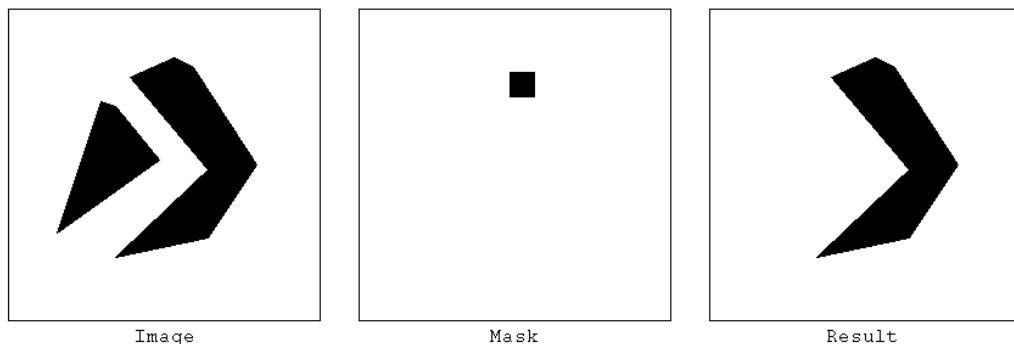


図 1.2 マスキング処理のための処理系

図 1.2 に示す回路は任意の領域をマスクングしその他の領域を消去するためのものである[6]。これを各画素ごとに組み込み、近傍の画素と情報をやり取りすることでマスクング処理を実現する。図 1.2 に注目すると、図の上部の **Mask** が“種”となり上段から‘1’が出力される。そのため次段では **Image** がそのまま通り抜け **Image** が‘1’であれば **Result** も‘1’となり近傍へ伝わっていく。そして **Image** が‘0’となる領域まで伝搬して行き、結果マスクを‘1’とした **Image** ‘1’の領域がそのまま残ることになる。また **Image** が‘1’の領域でマスクを‘1’としなかった領域は“種”ができず **Result** は全て‘0’となり、マスクングができることになる。



(a)



(b)

図 1.3 二次元に拡張したマスクングの例

これを二次元に拡張したものが図 1.3 であり、マスクを付けた領域のみが残っているのが見られる。このようにして、比較的単純な機能のセルを複数並べて、従来の方法より高速に画像処理の前処理部分を行うことができる。

### 1.3 本研究の目的

領域の重心検出といった画像処理でメインとなる処理を行うための前処理を組み込んだスマートセンサは広く行われているが、メインとなる処理そのものを組み込んだスマートセンサの研究はあまり行われていない。というのも重心検出のための処理回路が複雑で大規模なものになってしまうため、1 チップ上に集積できる画素数が減ってしまうからである。たしかに、これではセンサとして実用的とは言えない。

そこで本研究では感知した領域の重心とある程度の大きさを、比較的簡単な手法で高速に検出するための回路構成を提案する。そして提案した回路を受光系と共に集積し、スマートセンサとして実現する。またそのスマートセンサが、ロボットビジョンのようにリアルタイムでの処理が必要とされるものに対しても、十分に応用が可能であることを示す。

本論文の構成は次のようになっている。本研究では感知した領域の重心を検出するスマートセンサとして、2 つの回路を提案した。そこで第 2 章では、感知した領域の重心部分に光電流を集約させることで検出を行う回路構成について提案し、その回路の検証を行う。第 3 章では電位分布の局所性を用い、領域の重心を検出する回路を提案し、検証する。以上の二つの回路から検出された重心の情報をもとに、その座標情報を出力するための回路構成を第 4 章で提案し、検証を行う。以上の結果に対する本研究のまとめを第 5 章で行う。

## 第 2 章

# 光電流集約機構を用いた領域の検出

先ず、この章で提案する回路動作の概念図を図 2.1 に示す。近隣のノードと接続する抵抗網において、各画素で発生した光電流を、感知した領域の重心部分に集約させるように各画素のバイアス電位を変化させることによって、感知した領域の座標と大きさを高速に検出することができる回路を提案する。

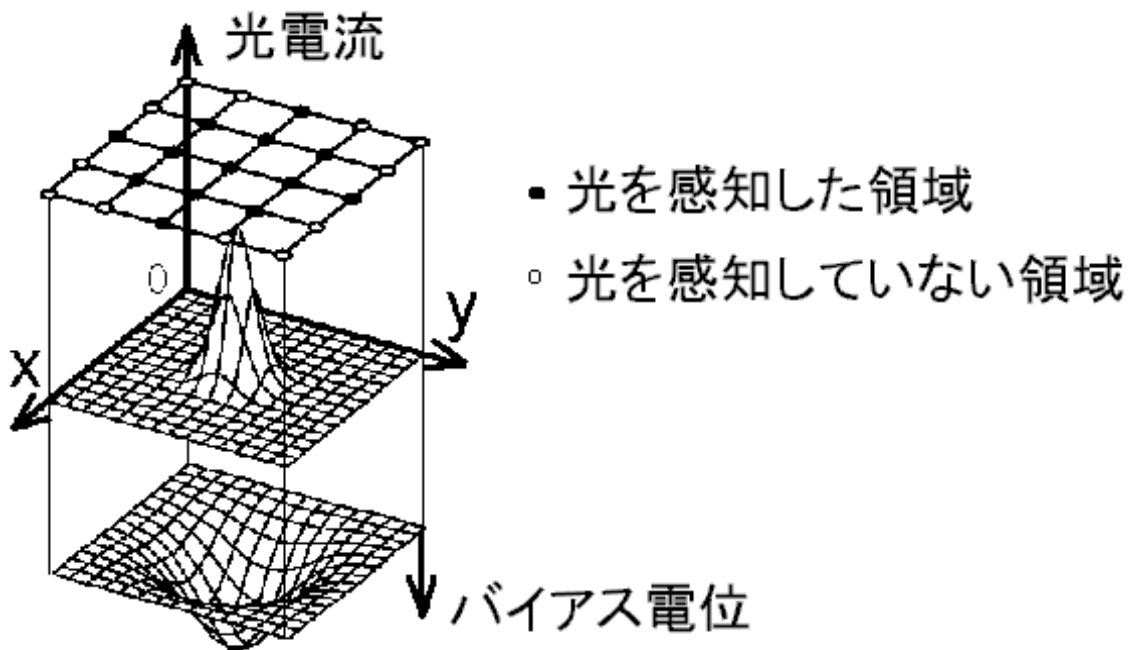


図 2.1 概念図

## 2.1 光電流集約機構の原理

本研究では、図 2.2 に示すような各ノードを抵抗で接続した構造を考える。各ノードにはフォトダイオードが接続されており、そこで発生した光電流によってノードの電位が決定される。このように光電流によって決められる各ノードの電位を利用することによって、感知した領域の重心を検出する方法を考えた。

### 2.1.1 動作原理

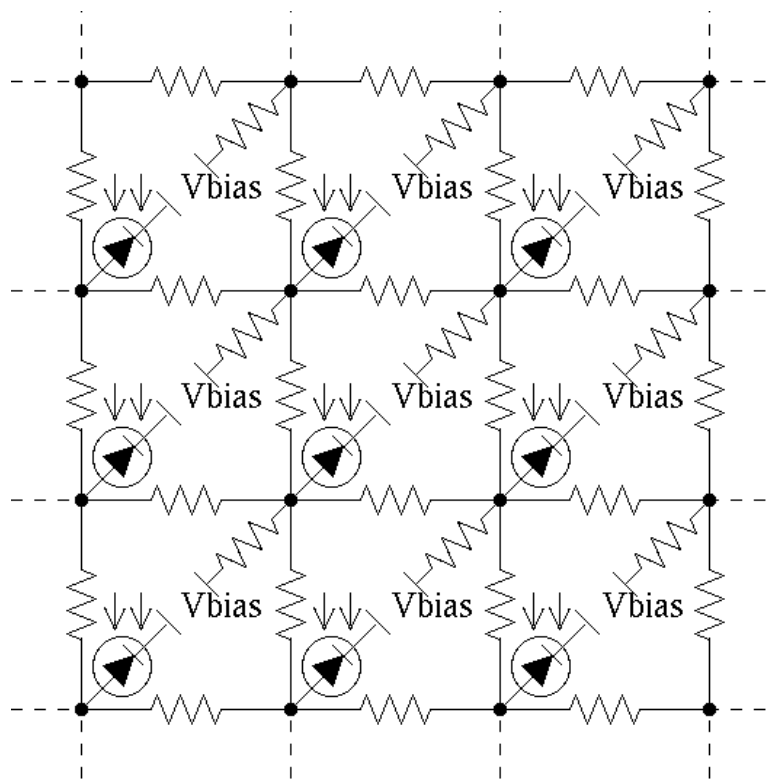


図 2.2 用いる抵抗網

図 2.2 のようなバイアス電位  $V_{bias}=0$  の抵抗網が無限に続くと仮定した場合、光電流によって決められる各ノード電位の中で、電位が極大であるノードは感知した領域の重心部分とほぼ等しくなる[5]。この事に関して、簡単のために一次元回路モデルで説明をする。図 2.3 のように水平抵抗  $R$ 、垂直抵抗  $r$  および、フォトダイオードに見立てた定電流源を配置した回路において、図中に丸で囲まれた三つの連続するフォトダイオードで光電流  $I$  が発生した場合を想定している。この時その三つの領域の重心となるノードの電位  $V_n$  は、同図 2.3

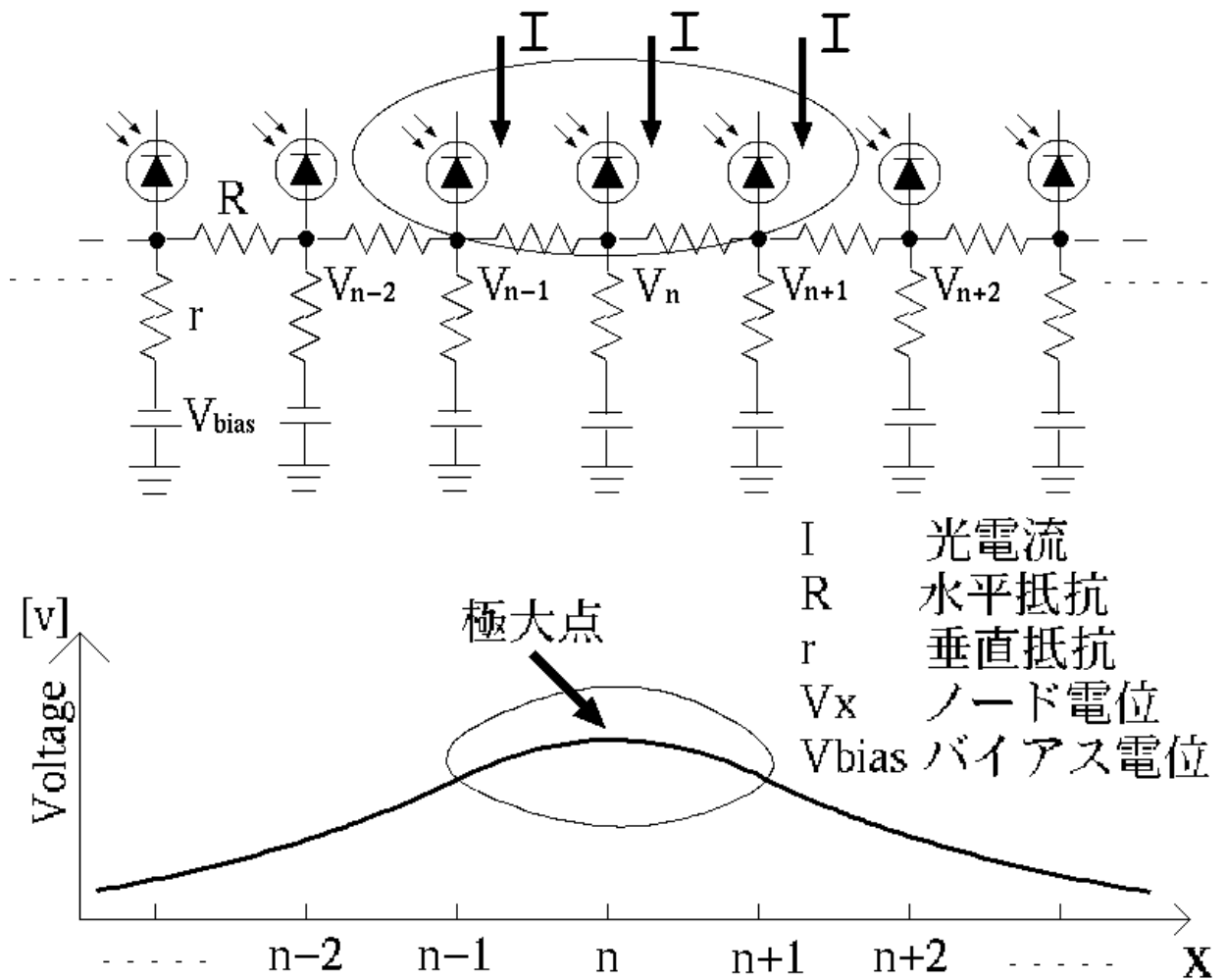


図 2.3 重心と極大点の関係

に示す電位分布から、極大の電位を持つのが分かる。次にこの回路を二次元モデルに拡張したものを考えてみる。二次元に拡張した時の回路は一次元の回路モデルを、各ノードを共通として縦方向・横方向にならべることになる。重ね合わせの原理から考えると、二次元に拡張した回路モデルにおいても極大の電位を持つノードが、感知した領域の重心部分となるのは明らかである。従って、発生した光電流によって決定されるノードの電位分布から、感知した領域の重心を検出することが可能であると言える。

以上のことから、フォトダイオードより得られる光電流によって決定される各ノードの電位分布において、その極大となる電位を持つノードを検出し、ノードのバイアス電位  $V_{bias}$  を下げてやることにより、光電流を集約させ領域の重心を検出することができる。また、感知した領域の大きさに比例した量の光電流を集

約させるようにバイアス電位をコントロールすることができれば、領域の大きさも同時に検出することができる。

### 2.1.2 極大点の判定方法

以上で述べたことから領域の重心を検出するためには、各ノードの電位から極大の電位を持つノードを検出しなければならない。そこで、あるノードの電位が極大の電位であるためにはどのような条件が必要となるのかを考えた。重心となるノードの電位は、抵抗で接続されている4近傍のノードの電位よりも高い電位を持つため、以下の4つの条件を満たす。

$$V_{n,n} - V_{n,n-1} > 0, V_{n,n} - V_{n,n+1} > 0, V_{n,n} - V_{n-1,n} > 0, V_{n,n} - V_{n+1,n} > 0 \quad (2-1)$$

ここで  $V_n$  は領域の重心となるノードの電位、 $V_{n,n-1}$ ,  $V_{n,n+1}$ ,  $V_{n-1,n}$ ,  $V_{n+1,n}$  はその4近傍の電位を示している。

従って、各ノードにおいて上に示した4つの条件をすべて満たしているかを調べることにより極大の電位を検出することができる。



## 2.2 光電流集約機構を実現する回路構成

以上に示してきた原理を実現するための回路構成をここで提案する。図 2.4 に示す回路構成は、一画素に組み込む受光部分と処理部分である。以下にそれぞれの回路部分に関して説明をする。

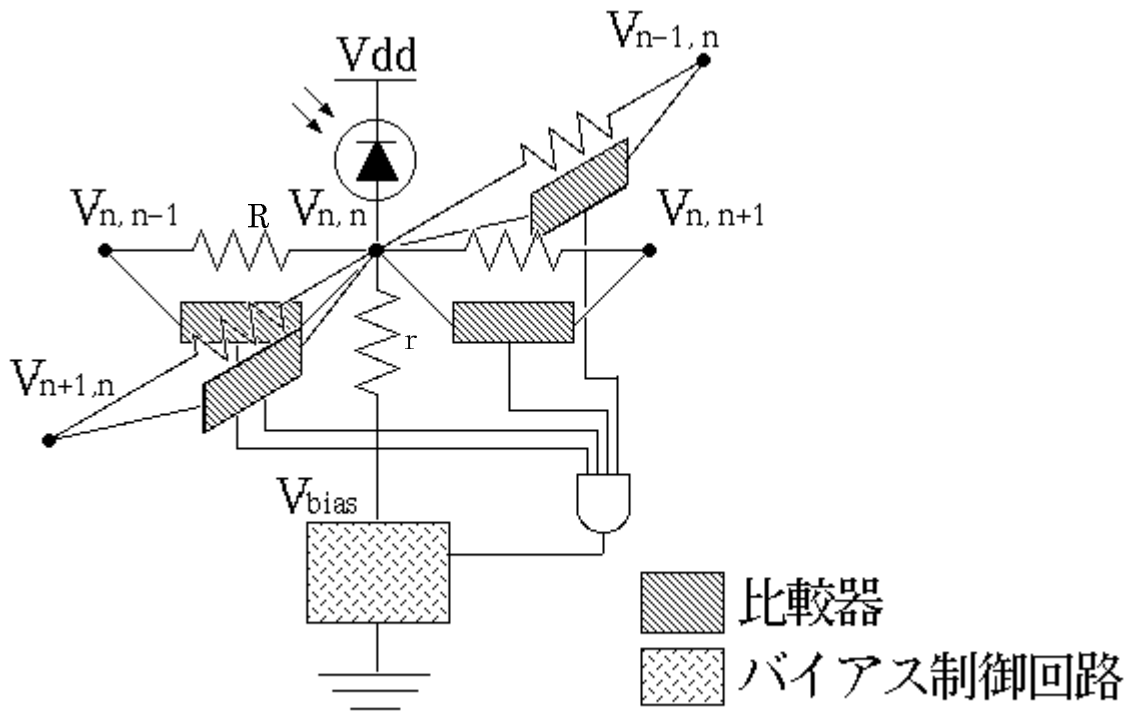


図 2.4 一画素分の回路構成

### 2.2.1 受光部分

まず、領域を感知した時に光電流を発生させるための受光部分に関して説明を行う。受光素子にはフォトダイオードを使用した。チップ上に集積するフォトダイオードはシリコン基板の上に  $n$ -well を作り、その上に  $p$  拡散を打ち込むことによって実現される。光のエネルギーにより、空乏層において電子ホールペアが生成されそこから電流を得ることが出来る。このフォトダイオードから得られる光電流は  $p$  拡散の面積に依存しており、およそ  $10[\mu\text{m}] \times 10[\mu\text{m}]$  の面積で、 $10^{-7}[\text{A}]$  から  $10^{-6}[\text{A}]$  の電流を得られる。ところが、この微少な光電流では後段に接続される回路を十分に動作させるだけの電圧を得ることができない。集積

度の点から考えてもフォトダイオードの面積を大きくすることは、集積できる画素数を減少させてしまう。このため、このフォトダイオードに簡単な電流増幅回路を用いることで光電流の増幅を行った。その回路構成は図 2.5 のようになっている。

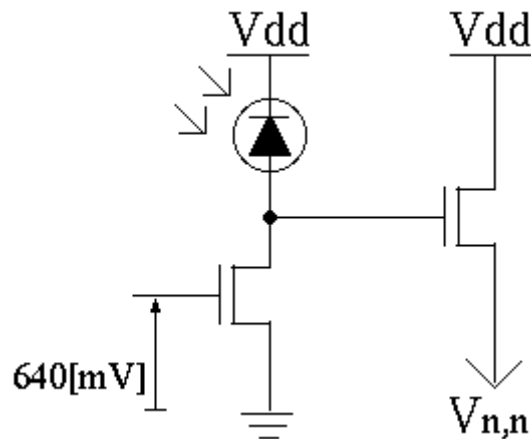


図 2.5 電流増幅回路

### 2.2.2 抵抗網部分

受光部分で発生した光電流が、ここで説明する抵抗網を巡ることにより各ノードに電位が決定するため、抵抗の値は発生するノードの電位に非常に影響を及ぼす重要なパラメータである。ここで使われる抵抗の値を決定するためには以下の三つの点を考慮しなくてはならない。

まず、水平抵抗  $R$  の大きさと垂直抵抗  $r$  の大きさの変化による、電位分布の変化を図 2.6(a)、図 2.6(b)に示しておく。ここで示しているグラフは、図 2.3で示した一次元回路モデルと同じ構成で 100 個のノードを一行に配置し、図 2.6 中で黒く示された領域内にあるフォトダイオードから光電流が発生している状況を考えている。図 2.6(a)では、水平抵抗  $R$  が垂直抵抗  $r$  と比べて十分小さい場合である。この図から各ノードに発生する電位の分布は平坦化しており、極大の電位とそれ以外の電位にほとんど差がなくなっているのが分かる。水平抵抗  $R$  が垂直抵抗  $r$  に対して小さな値になればなるほど、この減少は顕著に見られる。次に水平抵抗  $R$  が垂直抵抗  $r$  の値に近い場合、光電流が発生

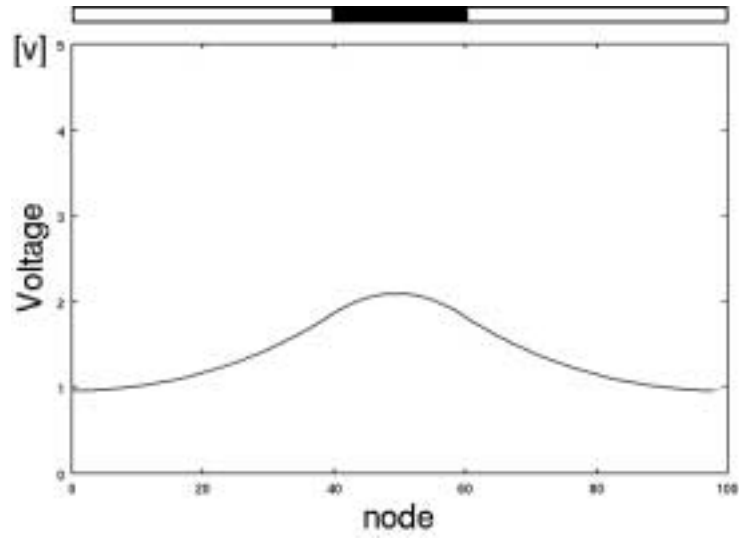
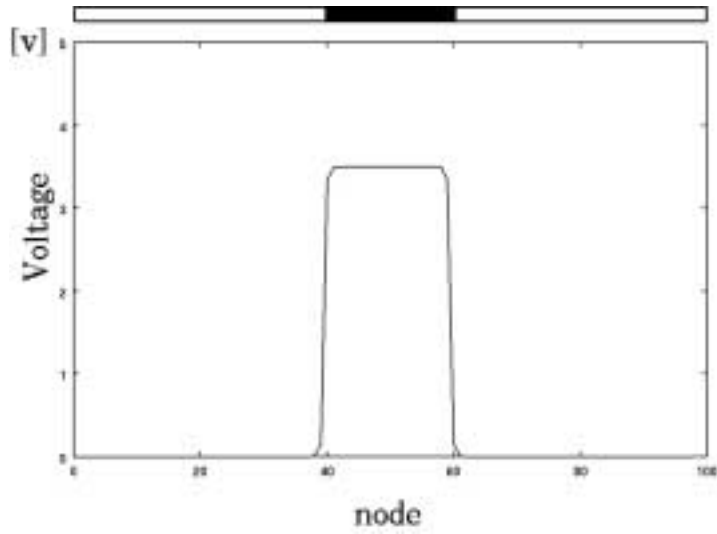
(a)  $R \ll r$  の場合(b)  $R \simeq r$  の場合

図 2.6 抵抗値の違いによる電位分布の違い

しているノードとそうでないノードははっきりと区別できるものの、感知した領域内のノードの電位はほとんど差がないのが分かる。

ここで考慮すべき一つの点として、領域の重心を検出するためには、式(2-1)で示した 4 つの条件を満たす電位を持つノードを検出するわけだが、極

大の電位を持つノードとその近傍のノードの電位差が小さければ小さいほど回路精度が求められ、条件を判定することは難しくなる。従って、先の図で示した電位分布のように重心となるノードの電位と4近傍のノードの電位に差がはっきりと見られない場合、極大の電位であるための条件を満たしているかを判定するのが難しい。そこで、判定しやすくするためには極大点付近において電位差がはっきり見て取れるように各抵抗の値を考慮してやる必要がある。

次に第二の点として、提案するスマートセンサを実現するためには、抵抗をチップ上で作成しなければならない。チップ上に集積する抵抗はトランジスタの線形領域を利用することで実現できるわけだが、抵抗の値が大きくなればなるほど、チップ上で実現する抵抗が必要とする面積は大きなものとなる。従って集積度という面で考えると、実現できる抵抗の値としては限界がある。そこでスマートセンサを実現するための現実的な限界値をチップ作成に用いるプロセスのトランジスタパラメータを用いて検討した結果、配置する画素数の兼ね合いから限界値として1[MΩ]程度と設定した。

三つ目の点だが、先にも述べた通りフォトダイオードから得られる光電流は微小であるため、後段に接続される回路を十分に動作させるだけの電圧を得ることができない。光電流を増幅することにより、改善はされるものの十分に動作させることは難しい。そのため、後段に接続される回路を十分に動作させる電位を得るためには、垂直抵抗を比較的大きな値に設定しなければならない。

以上の三つの点を踏まえて、この回路で使用する抵抗の値を水平抵抗の値を3.3[kΩ]、垂直抵抗の値を1[MΩ]とした。図2.7に決定した抵抗を用いた場合の電位分布を示す。なお、シミュレーションの条件は先ほどのものと同じで、図中の黒く示した領域で光電流が発生していることを想定している。

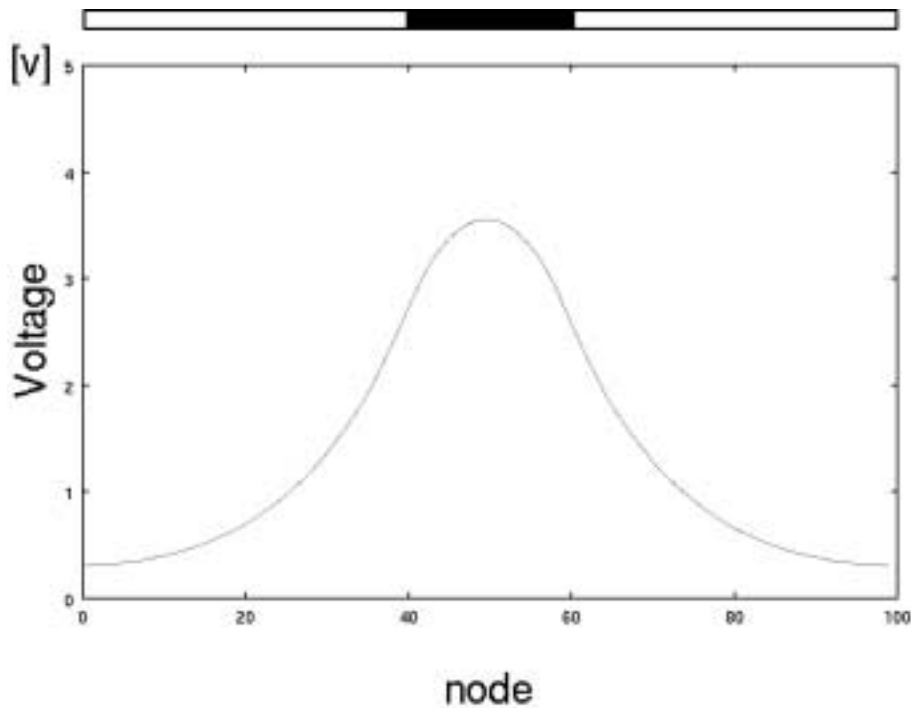


図 2.7 決定した抵抗値での電位分布

### 2.2.3 比較回路部分

この章で提案する回路は、光電流が発生することにより決定されるノードの電位から極大の電位を持つノードを検出することにより、重心の検出を行うわけだが、以下に説明する比較回路部分は、極大の電位を検出するための回路である。回路構成は図 2.8 のように、隣り合うノードの電位を入力とする比較器 [6] を用いた構成である。各比較器によって得られる出力は、極大の電位であるための 4 条件それぞれを真か偽かで表現したものとなる。4 つの条件すべてが満たされたかどうかは、4 つの比較器の出力で論理積をとることによって知ることができる。

この結果を後段に続くバイアス電位制御部分に渡すことにより、極大の電位であると判断された場合にのみ、そのノードのバイアス電位を低くするように調節することができる。

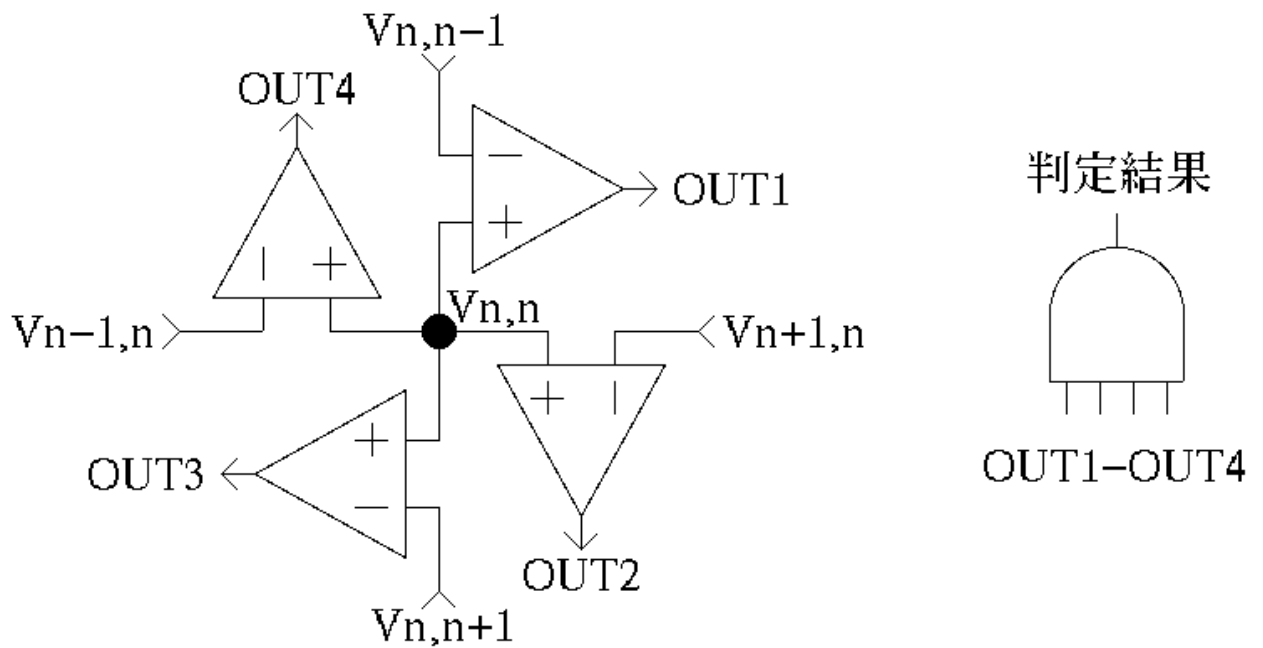


図 2.8 比較回路部分

#### 2.2.4 ノードバイアス電位制御回路

極大の電位を持つと判定されたノードのバイアス電位を制御するための回路構成を図 2.9 に示す。前段の論理積の結果は図中の CTRL に接続される。この回路ではコンデンサ  $C$  の充放電を制御することによって、バイアス電位を調節する手法を考えた。まずリセットの状態でもコンデンサを十分に充電した状態にしておく必要がある。というのは、バイアス電位を下げることによって光電流の集約を実現するため、初期状態においてある程度のバイアス電位を全ノードに対して一様に与え、それを調整するといった手法をとらなければならないからである。前段の結果からノードが極大の電位を持つ場合は、CTRL 信号によりコンデンサ  $C$  の放電が行われ、それ以外の場合は変化させない仕組みである。以上の回路構成によって、バイアス電位  $V_{bias}$  を調節し、極大の電位を持つノードへ光電流の集約させることができることになる。この回路で使用したコンデンサ  $C$  の容量は  $3[\text{pF}]$ 、電流源の電流は  $30[\mu\text{A}]$ 、抵抗  $R$  は  $10[\text{k}\Omega]$  である。

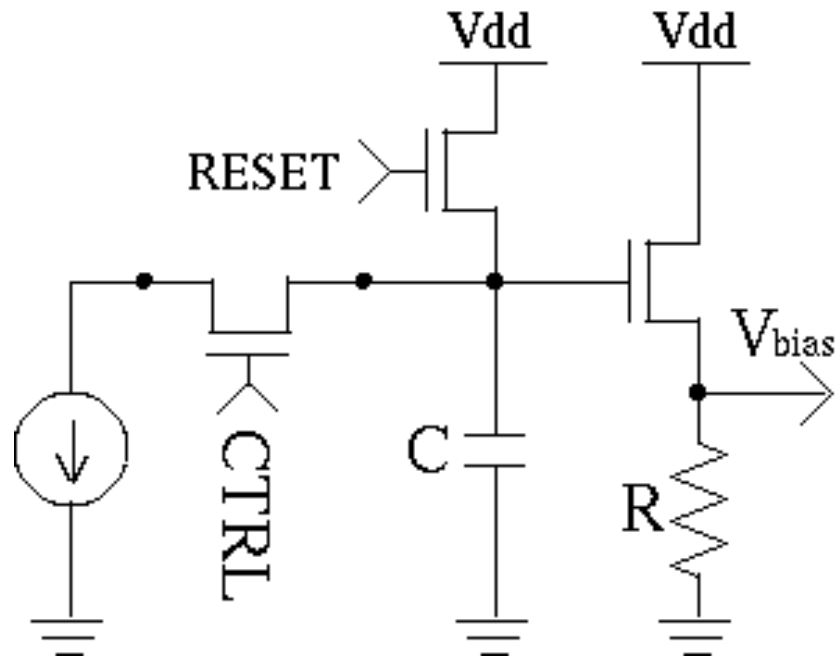


図 2.9 バイアス電位制御回路

### 2.2.5 その他の考慮点

以上の回路構成を用いてスマートセンサを実現するため、回路試作を行った。実際に回路設計を行うにあたり、上記以外の点で考慮しなくてはならない点があるので、ここで説明をする。

まず比較回路部とバイアス制御部分についてだが、ここで提案している回路ではバイアス制御回路によって、全てのノードに一律のバイアス電位をかけてから動作させることを考えているため、その間は比較器からの情報を反映させては、回路全体を正しく動作させることはできない。そこで図 2.10 のように RESET の反転した信号を先ほど述べた論理積の入力の一つに加える。これによってリセットの間は比較器からの情報は必ず 0 となり、コンデンサの放電がリセットと同時に終わることはなくなる。

次に、比較回路をノード間に接続する際について述べる。ノード間に接続される比較器の出力情報は接続される二つのノード共通の出力となるのだが、その出力された情報が持つ意味は、ノードによってまったく違ったものとなる。そこでどちらか一方に対して出力結果を反転した信号を使わなくてはならない。そ

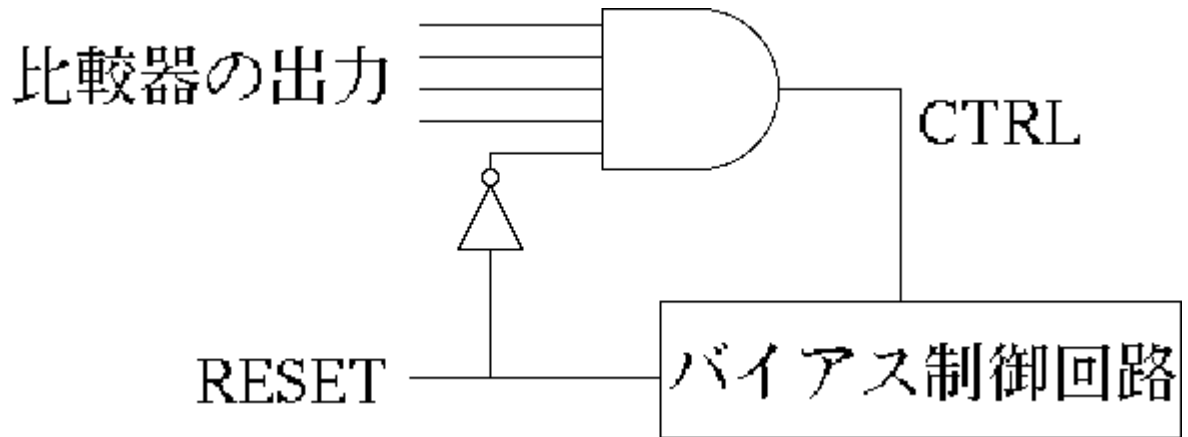


図 2.10 改良した極大点判定回路部分

ここで図 2.11 に示すようにどちらかの出力に対してインバータを使うことで解決できる。今回試作した回路においては、作成した比較器の特性があまり良くなかったので、そのようにはせずに入力の端子を入れ替えた 2 つの比較器をつなげることで解決した。

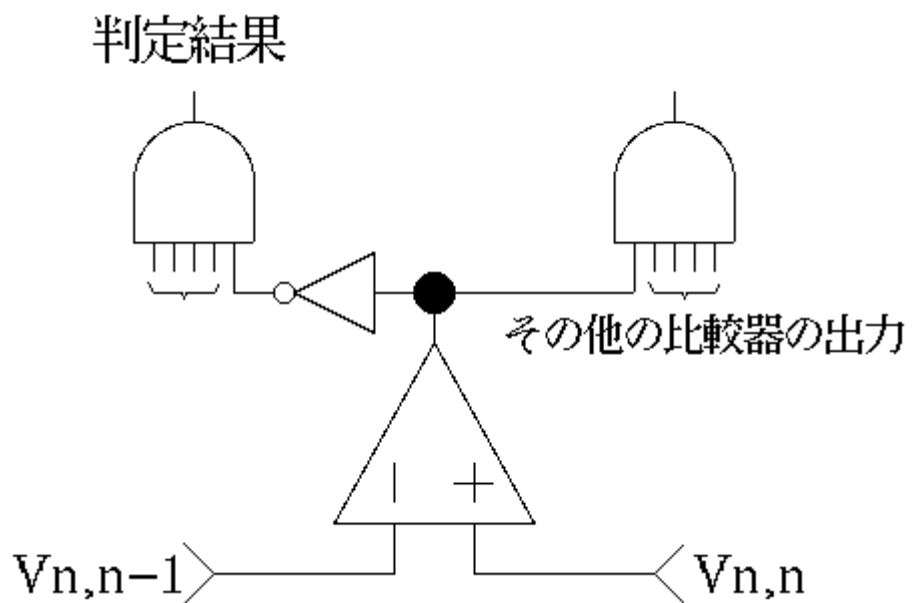


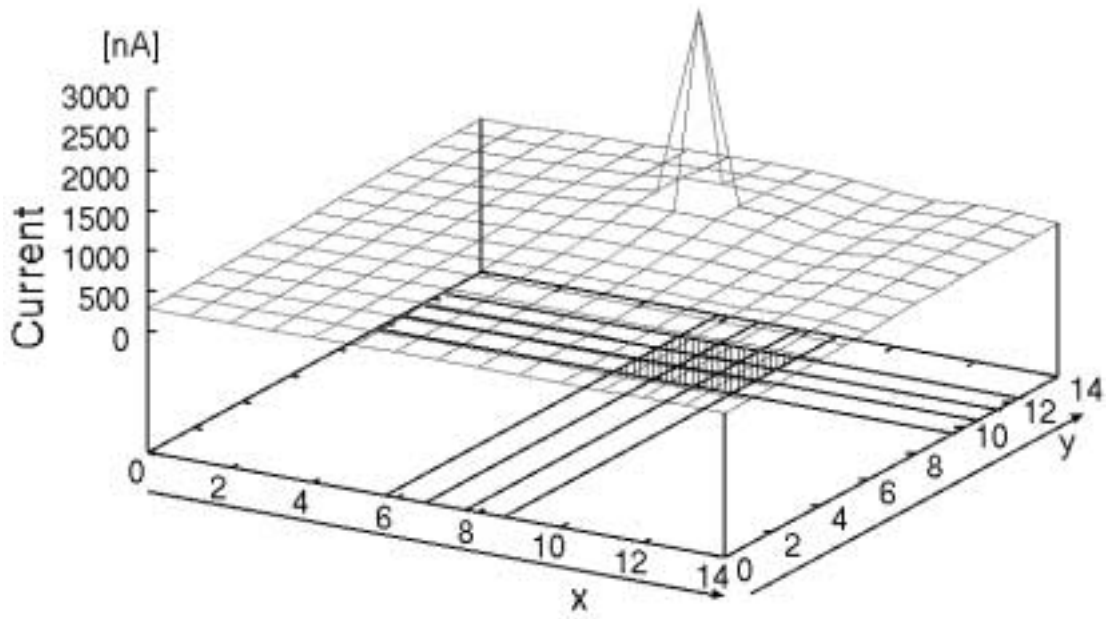
図 2.11 改良した比較回路部分



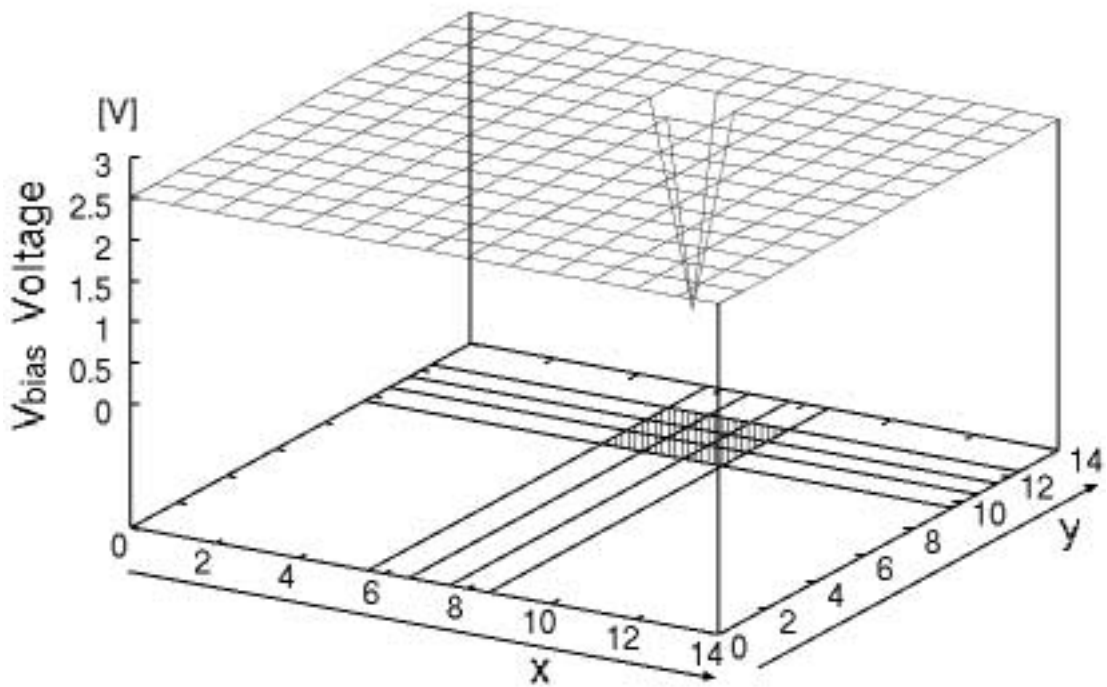
## 2.3 検討した回路の評価

以上に述べた構成回路でスマートセンサを実現するために、CMOS0.6[ $\mu$ m]プロセスを用いて回路試作を行った。比較器などの各要素をネットリストで記述し、用いるプロセスのトランジスタパラメータを用いて H-SPICE による回路シミュレーションを行った結果を図 2.12(a)、図 2.12(b)に示す。ここでは 15x15 のノードを配置し、(7,11)のノードを中心とする 3x3 の領域で 100[nA]の光電流が発生していることを想定して行った。この結果から見て取れるように、感知した領域の重心部分のノードにおいて、バイアス電位が下がり、光電流が集約されており、感知した領域の重心が検出できた。

次にこの検出された情報をもとにその座標情報を出力しなくてはならない。この章で説明してきた回路の試作においては、水平・垂直方向に論理和をとることで座標情報の出力する方法を用いた。その回路構成と動作に関しては 4 章にて説明を行うので、ここでは省略する。座標情報出力回路を組み込み、先ほどと同じ条件でシミュレーションを行った結果では、光電流を発生させ最初に物体が検出されるまでに、およそ 13[ns]の時間を要した。



(a) 集約した光電流



(b) 光電流集約後のバイアス電位

図 2.12 シミュレーション結果

## 2.4 スマートセンサの回路試作

以上で提案した回路を、東京大学大規模集積システム設計教育センターを通し、ローム(株)および凸版印刷(株)の協力のもと CMOS0.6[ $\mu\text{m}$ ]、3層金属配線のプロセスで試作を行った。チップサイズは 4.5[mm]ダイで、その一画素のレイアウトと全体のレイアウトを図 2.13、および図 2.14 に示す。画素数は 12x13 個でトランジスタ数は11,232 個で、開口率は 4.5%である。また、受光部以外の拡散層で光電流が発生するのを防ぐために、受光部分以外を 3層目の金属でシールドを施している。このチップは 1999 年 10 月下旬に納品される予定であったが、協力ベンダーの不都合によりいまだ納品されていない。そのため、実際に光電流を発生させての動作検証等を行っていない。

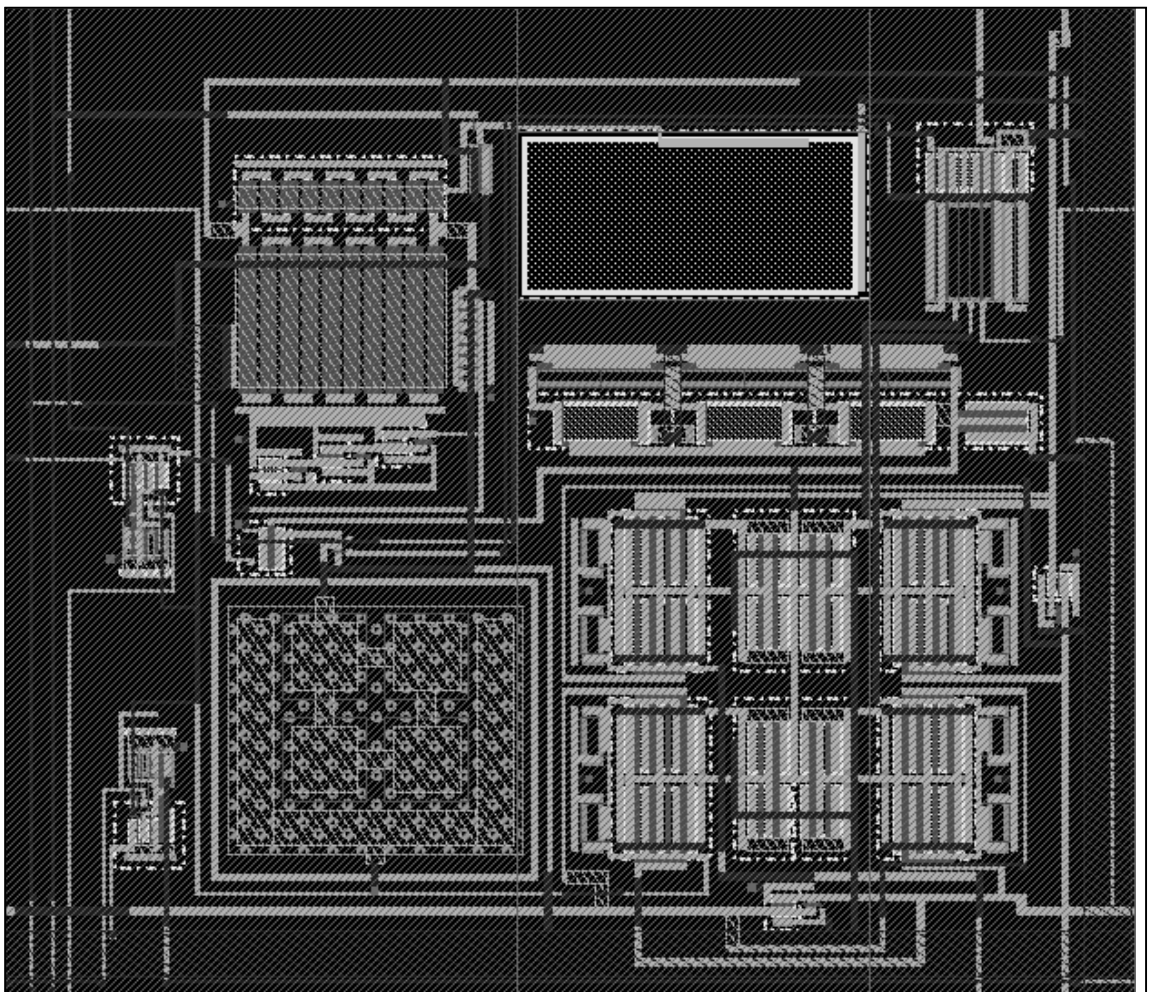


図 2.13 一画素分の回路レイアウト図

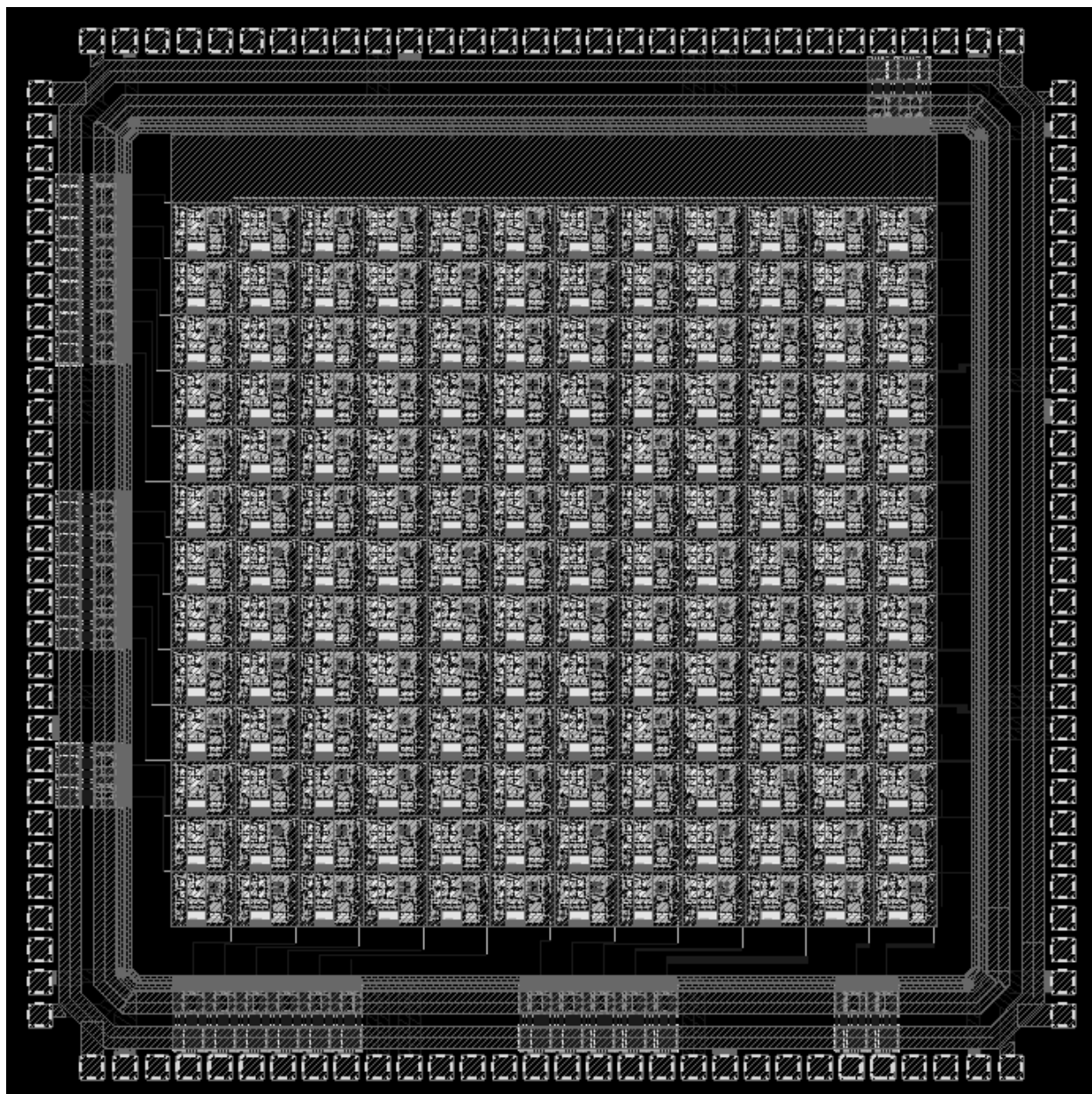


図 2.14 作成したスマートセンサのレイアウト図

## 2.5 この章のまとめ及び問題点

この章で提案した回路は、ノードに発生した電位の状態によって重心の情報を出力するため、非常に高速にその情報を出力することができた。通常、ロボットビジョンの分野において、1枚の画像を1[ms]以下の時間で処理することが望ましいとされているが、この章で提案した回路ではチップ上に処理回路を作成することで得られる並列処理の利点を生かして、1枚の画像に対し約13[ns]と非常に高速な検出が実現できた。従って動作速度の面では、ロボットビジョンとして十分応用が可能である。

しかし、この回路を実際に作成することで、以下の点が問題点として考えられた。まず、実際に回路のレイアウト設計を行ってみると、思いのほか回路規模が大きくなってしまい、集積できる画素数が少なくなったということである。今回試作で用いたプロセスとチップサイズでは12x13個の画素数であったが、これを現在主流となりつつある0.35[ $\mu$ m]のプロセスでチップサイズ9[mm]ダイのサイズを使ったと考えた場合、単純に考えれば画素数は48x52個となる。ロボットビジョンでの応用を考えた場合、状況によって必要とされる画素数はさまざまであるが、一般的とされる画素数100x100前後には至らない。集積度を向上するためには、より簡単な回路を考える必要がある。

次に、感知した領域の大きさに比例する電流量が垂直抵抗に流れることを期待して、ノードのバイアス電位を調整し光電流を集約させたのだが、実際には領域の大きさに比例する電流は流れなかった。実際にバイアス電位を感知した領域の大きさによって制御するのは難しく、これを実現する為にはさらに大きな規模を必要となり、実用的と言える画素数を組み込むことができなくなる。したがって、大きさの情報も検出することを考えるなら新たな手法が必要である。

そこで、次の章において、これらの問題点を改善する新しい回路を提案し、その動作原理、および回路構成に関して説明を行う。

## 第 3 章

# 電位分布の局所性を用いた領域の重心 検出

前章にて提案した回路では、処理系の回路が思いのほか大規模になってしまったという問題点と、感知した領域の大きさに比例する光電流を集約させるようにノードのバイアス電位を調整することが難しく、大きさの情報を得ることができないという問題点があった。この章においてそれらの問題点を解決するために、よりコンパクトな新しい回路構成を提案し、その検証を行う。

### 3.1 電位分布の局所性を用いた重心検出の原理

この章で提案する回路で用いられる原理は、前章で提案した回路に用いた原理と同じである。やはり近隣のノードと接続する抵抗網において、各ノードにつながるフォトダイオードから得られる光電流によって決定される各ノードの電位を利用し、感知した領域の重心と大きさを検出するものである。

#### 3.1.1 大きさ検出の原理

前章で提案した回路では、極大点の情報からそのノードのバイアス電位を調節し集約した光電流の量で、感知した領域の大きさの情報も検出しようとしたわけだが、ここでは、感知した領域の大きさを知る新しい方法として、感知した領域の重心となるノードに発生する電位の大きさを利用することを考えた。

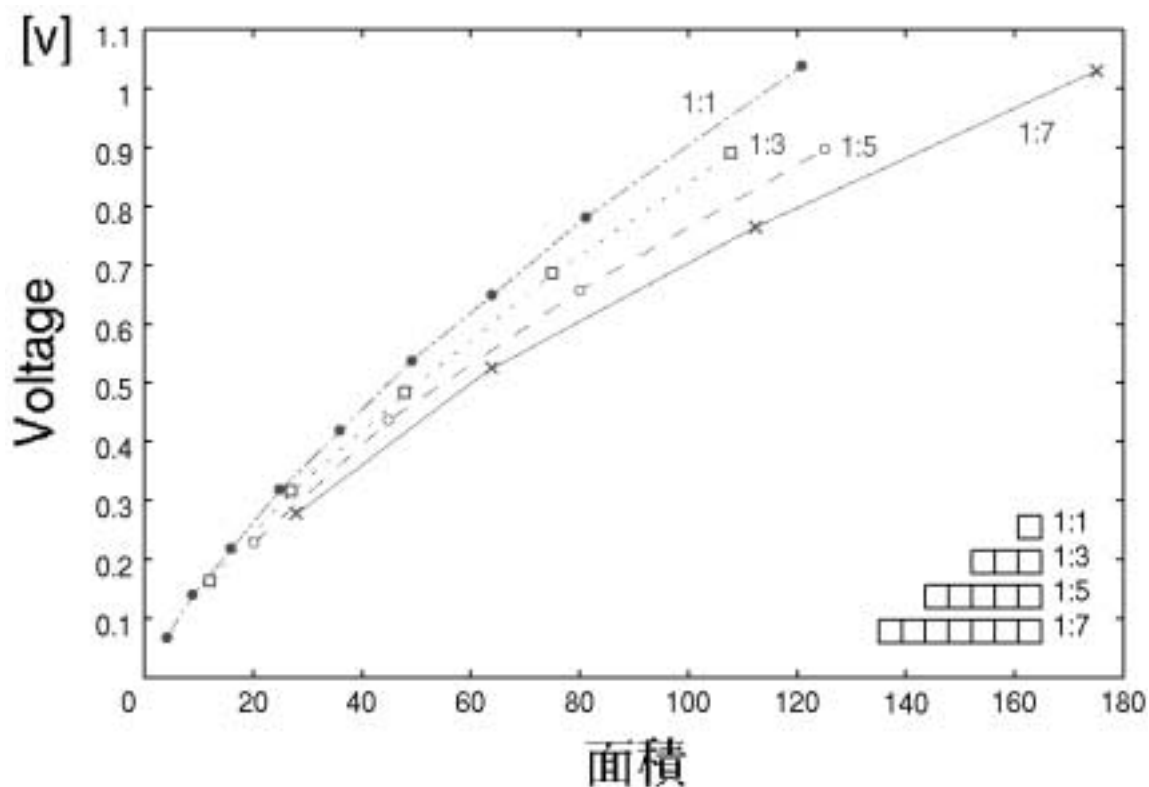


図 3.1 重心となるノードの電位

そこで感知した領域の大きさによって、重心となるノードの電位がどのように変化するかを、C 言語を用いてシミュレーションを行った。ここでのシミュレーションは、 $200 \times 200$  の画素を想定し重心とできる全ての点を重心とした時に得られた電位を平均したものである。図 3.1 にその結果を示す。このグラフでは、感知した領域の形状を縦横比により分類している。

縦横比の違いによって発生する電位の違いは見られるが、感知した領域の面積増加にしたがって重心のノードに発生する電位は増加しているのが分かる。このことから重心の電位を出力することにより、感知した領域のおおまかな大きさが検出できると考えた。図 3.1 に示したグラフでは、縦横比が大幅に違うと重心のノードに発生する電位もかなり違ってくるが、スマートセンサとして実用を考えたときに、実際にそれほど縦横比が違う複数の領域を同処理内で検出しなければいけない状況というのは現実的ではない。従ってスマートセンサとして実用するにあたり、感知した領域の重心となるノードの電位を出力することによって、領域のおおまかな大きさを知ることができるといっても問題はないと考えた。

### 3.1.2 極大点の判定方法

感知した領域の重心を検出するために、光電流の発生による各ノードの電位から極大となるノードを検出しなければならないのだが、ここで提案する検出の方法は前章で用いた方法とは異なる。極大の電位を持つノードであるかどうかは、近傍の電位よりも高いかそうでないかによって決められるわけだが、それは単純に比較の対象をノードの近傍とするのではなく、図 3.2 に示すような時間で減衰するのこぎり波形を全画素共通の比較の対象として用いることでも検出できる。

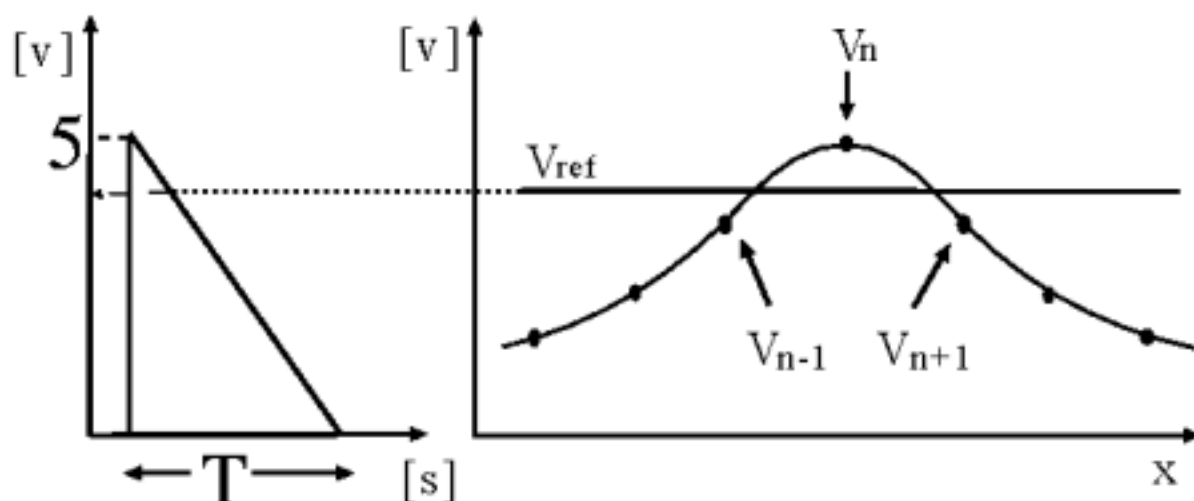


図 3.2 極大電位判定方法

光電流によって発生したノードの電位分布が同図 3.2 中に示すようなものであった場合、極大の電位はある時間において、以下の条件を満たす。

$$V_{n-1}, V_{n+1} < V_{ref} < V_n \quad (3.1)$$

ここで、 $V_n$  は極大の電位、 $V_{n-1}, V_{n+1}$  はその近傍の電位、 $V_{ref}$  はある時間にお



けるのこぎり波形の電位を示している。ここから分かるように、のこぎり波の一周期でこの条件を満たす電位を持つノードを検出することによって、極大の電位を持つノードを検出することができる。また、全画素に対して参照電位となるのこぎり波形を電位の高い方から低い方に下げていくことによって、高い電位を持つ極大点から順に検出していくことができる。また極大の電位を検出した時間でののこぎり波の電位を調べれば、先にも示した通り感知した領域のある程度の大きさも検出することができる。

### 3.2 局所性を用いた重心検出のための回路構成

以上で示した原理を実現するための回路構成を提案する。図 3.3 は一画素に含まれる回路構成を示している。以下にそれぞれの回路部分に関して説明を行う。抵抗網部分と受光部部分に関しては、前章での回路と同じものを用いるため、ここでの説明は省略する。

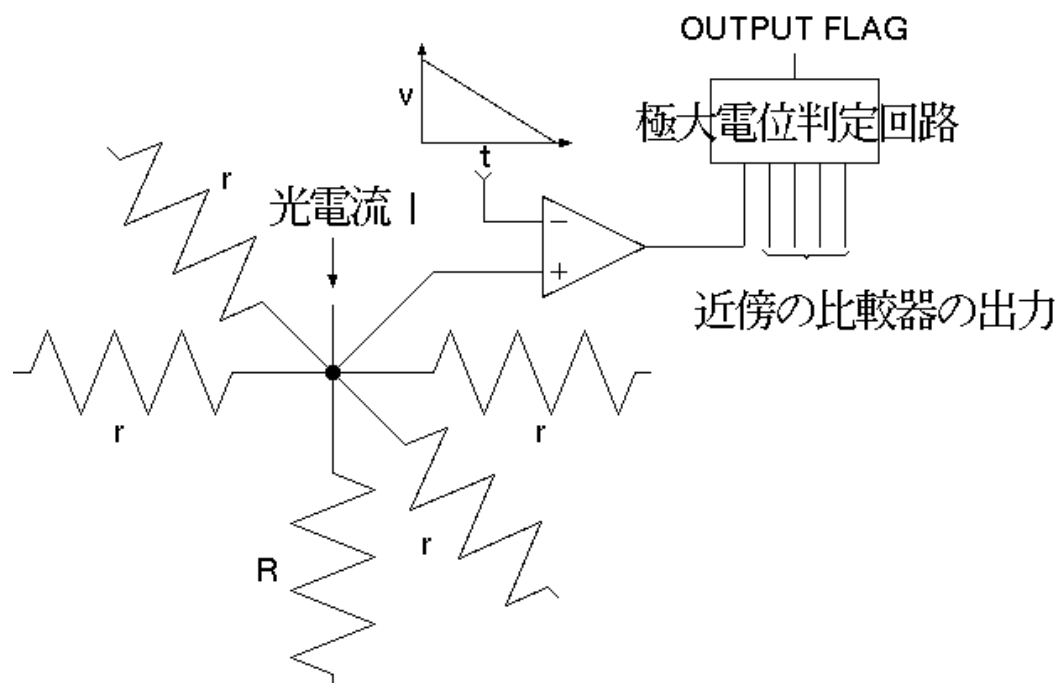


図 3.3 一画素分の回路構成

### 3.2.1 極大電位判定回路

各画素に配置される比較器の入力には、ノードの電位と参照電位となるのこぎり波形生成回路に接続されている。参照電位の一周期の時間すべてにおいて、ノードの比較器からの出力と近傍のノードに接続される比較器の出力を調べ、その組み合わせから極大電位であるかの判定を行う。出力の組み合わせによる判定のための真理値表を表 3.1 に示す。

$V_{n-1}$	$V_n$	$V_{n+1}$	OUTPUT
0	1	0	1
1	1	*	0
*	1	1	0
*	0	*	0

表 3.1 極大電位判定のための真理値表

この真理値表を満たす回路を考えるわけだが、比較器の出力はノードの電位が参照電位よりも大きくなった時点で出力が変化してしまうので、一度極大の電位を持つノードであると判定されたものに関しては、近傍の比較器の出力が変化してもその情報を保持しておく機能を持たなくてはならない。この点を考慮して図 3.4 に示すような論理回路を考えた。このようにフィードバックループを設けることで一度検出した極大の電位を持つノードであることを示すフラグを保持することができる。なお図 3.4 に示す回路では、回路構成を簡単にするため比較器部分において参照電位よりもノードの電位の方が低い時は 5v、大きくなった時に 0v を出力することを前提とした回路構成を用いている。

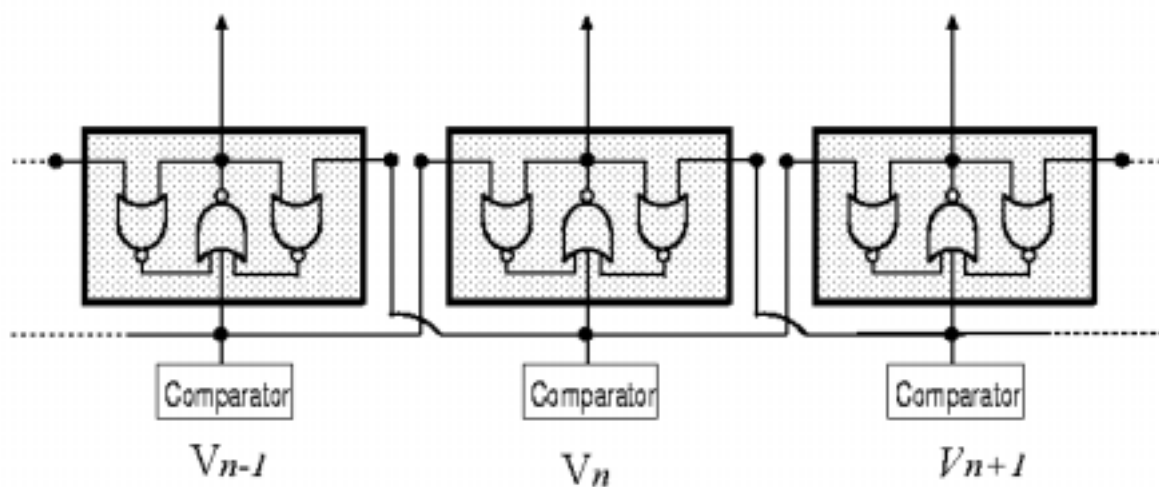


図 3.4 極大電位判定回路

### 3.2.2 のこぎり波形生成回路

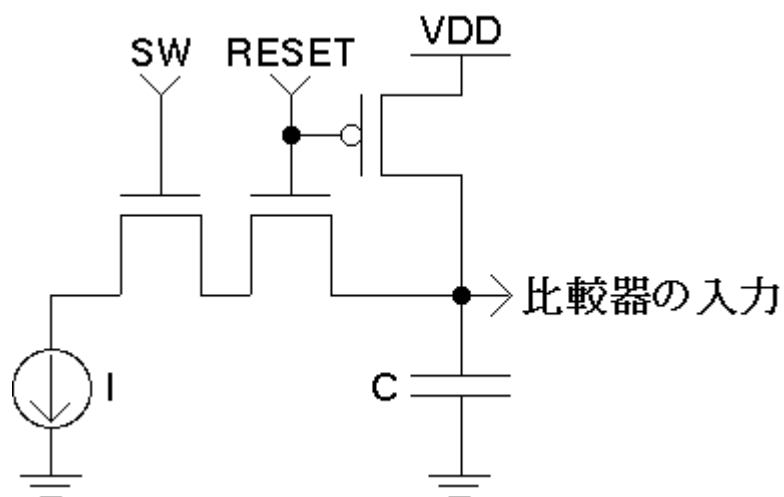


図 3.5 のこぎり波形生成回路

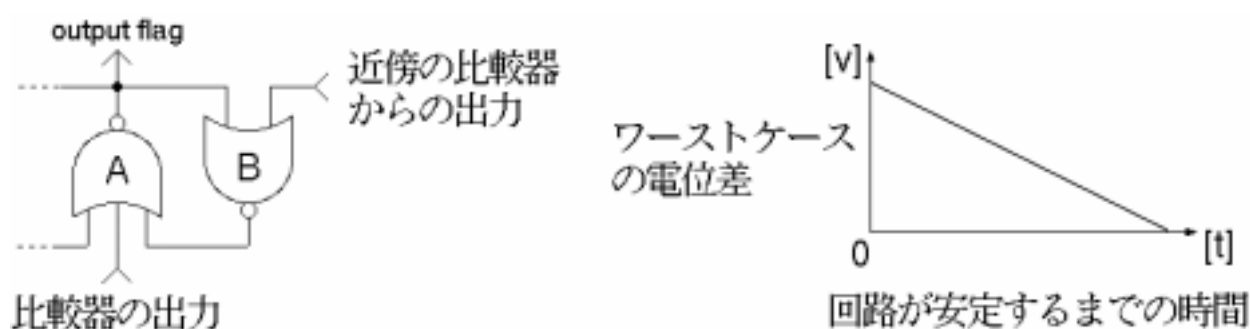
参照電位はチップの外部から入力すれば良いので、あえてチップ上で実現する必要はない。ただ、この章で説明している回路の後段に続く座標情報出力のための回路の都合により、重心が検出された際に参照電位を一時止める機構が必要があるので、そのための回路構成を図 3.5 に示しておく。前章で提案

したバイアス制御回路とほぼ同じ回路構成をしている。先ずコンデンサを十分に充電した状態にし、一定間隔で電荷を引き抜いていくことによって、のこぎり波形を生成することができる。図中の SW には座標情報を出力回路からの出力が接続され、ここに入力される信号によって参照電位を下げるかまたは停止させるかが決められる。またコンデンサを充電している間は、その信号の値が反映されないようにゲート一つ加えることで対処をした。

### 3.2.3 のこぎり波形の周期

以上で示した回路によって、感知した領域の重心を検出するわけだが、ここまで述べてきた回路を正しく動作させるためののこぎり波形の周期を決定しなくてはならない。これは、この章で提案する回路のおおよその動作速度となる重要なパラメータである。以下に周期の決定に関する説明を行う。

周期を決定するためには、極大点判定回路部分を考慮しなくてはならない。先ず図 3.6(a)に示す回路が極大の電位を持つノードの後段に続く極大点判定回路であるとする。すると、極大であるかどうかの判定は図中のゲート A の結果によってされるわけだが、このゲート A の出力が安定したものになる前に、参照電位が近傍の電位よりも小さくなることによってゲート B の出力が反映されてしまつては、期待する情報を得ることができない。



(a)問題となる回路部分

(b)周期決定のための方法

図 3.6 周期決定方法

次に図 3.6(a)に示す回路が極大ではない電位を持つノードの後段に接続される極大電位判定回路であるとする。この場合には、先ほどとは逆にゲート B が安定した出力を出す前に、参照電位がノードの電位よりも低くなることで、ゲート A に接続される比較器の結果が反映されてしまえば、やはり正しい動作は期待できない。従ってこの回路から常に正しい情報を得るためには、ゲート A、ゲート B が安定した状態になってから次の電位を検出しなければならない。これらの現象は、あるノードの電位とその近傍の電位の差が非常に小さい時に考えられる。従ってこのような現象を発生させないためには、この章で提案するスマートセンサの用途にしたがってワーストケースとなる電位差を決定しゲート A、ゲート B のうちで、安定した出力をするまで長い時間を必要とする方の時間を基準にして、図 3.6(b)に示すような傾きを持つのかぎり波形から周期を算出しなければならない。

以上の点を考慮して、本研究ではワーストケースとなる電位差を  $0.1[\text{mV}]$  とし、用いるプロセスのゲート遅延を踏まえて参照電位の周期が  $30[\mu\text{s}]$  の時、正しく動作できることが分かった。実際に試作する回路では十分に余裕をとって周期を  $50[\mu\text{s}]$  と設定している。

### 3.3 検討した回路の評価

以上で述べた回路構成を、前章同様 CMOS  $0.6\mu\text{m}$  プロセスを用いて試作するために H-SPICE によるシミュレーションを行った。その結果を図 3.7 に示す。このシミュレーションでは  $15 \times 15$  の画素数で図中に示す領域において  $100\text{nA}$  の光電流が発生していることを想定した。回路の動作から、同図に矢印で印を付けた部分が重心であると判定された。この結果より、感知した領域の重心がのかぎり波形の周期となる  $50[\mu\text{s}]$  で検出されたことが分かる。

次にこの回路で得られた重心の情報を座標情報出力回路に渡すことになる。ここで提案する回路に用いる座標情報出力のための回路は前章で用いたものとは異なり、検出点に対して逐次処理を行うことで座標情報を出力する回路を用いた。座標情報出力回路の回路構成およびその動作は 4 章で説明を行うのでここでの説明は省略する。

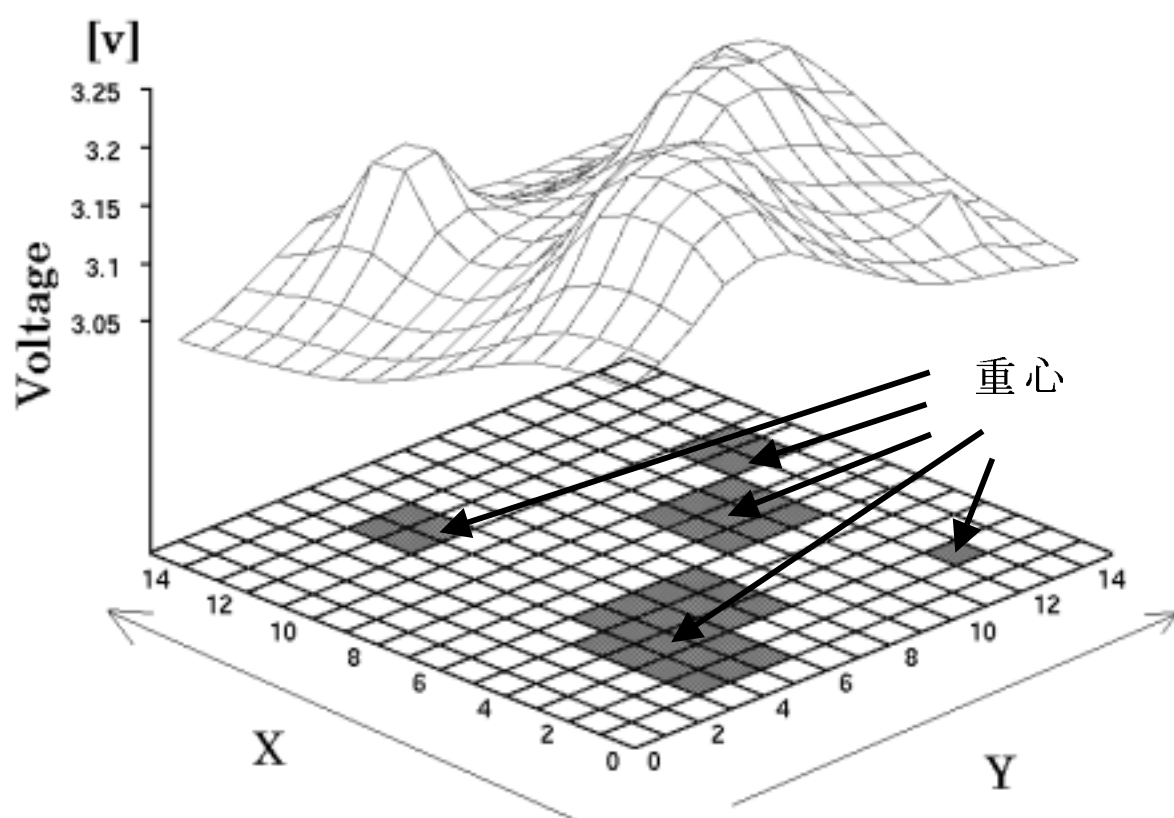


図 3.7 シミュレーション結果

### 3.4 スマートセンサの回路試作

この章で提案した回路は、CMOS $0.6\mu\text{m}$ 、3層金属配線のプロセスで試作を行った前章で提案した回路との比較を行うため前回同様のプロセスを用いてレイアウト設計を行った。来年度にこの試作したレイアウトをもとにチップを試作する予定である。一画素分のレイアウトと回路全体のレイアウトを図3.8、図3.9に示す。チップサイズ $4.5\text{m}$ ダイで画素数は $23\times 23$ 個。用いたトランジスタ数は $69,575$ 個で開口率は $5.7\%$ である。受光部分以外には3層目の金属によりシールドを施している。

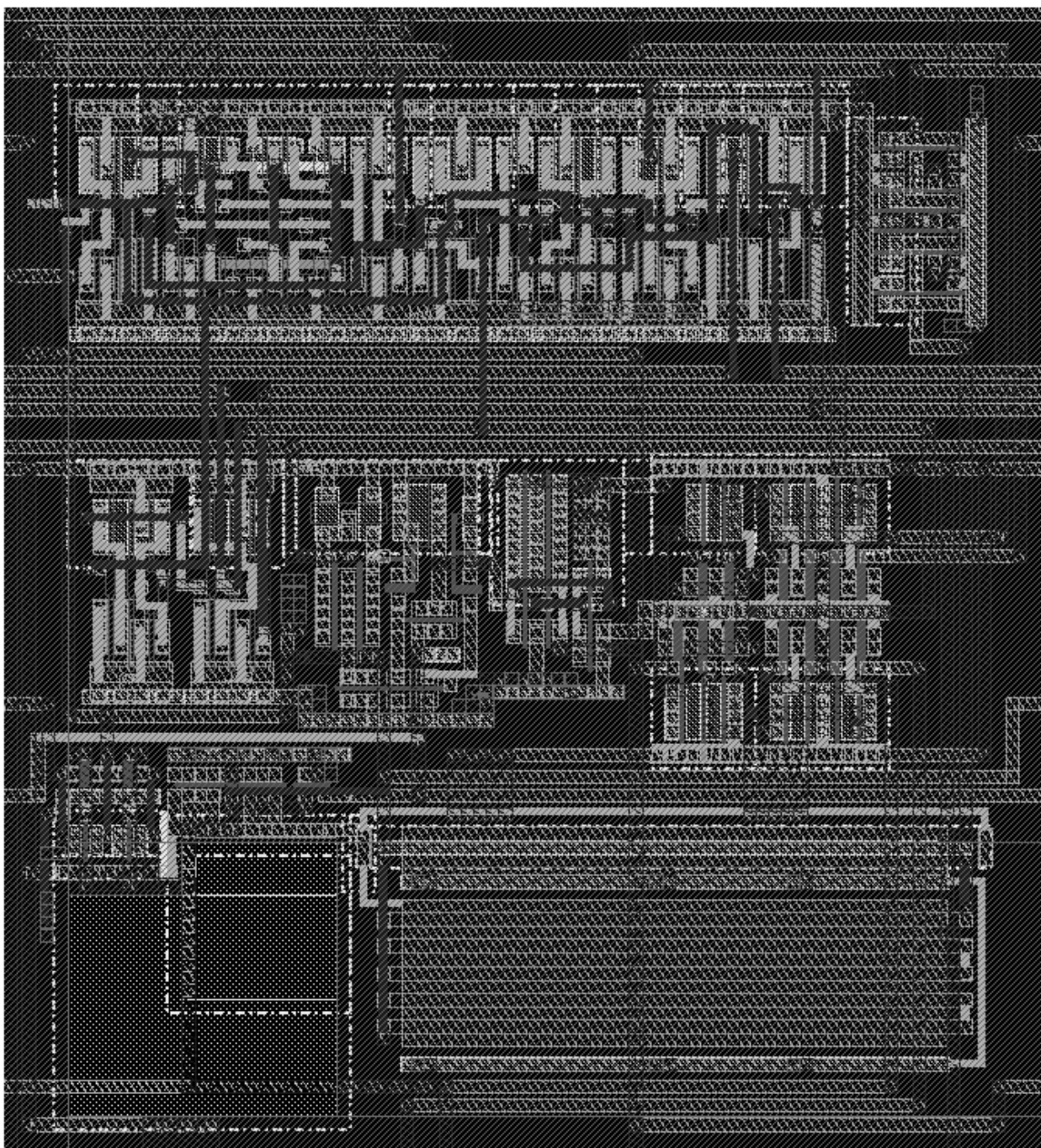


図 3.8 一画素のレイアウト図



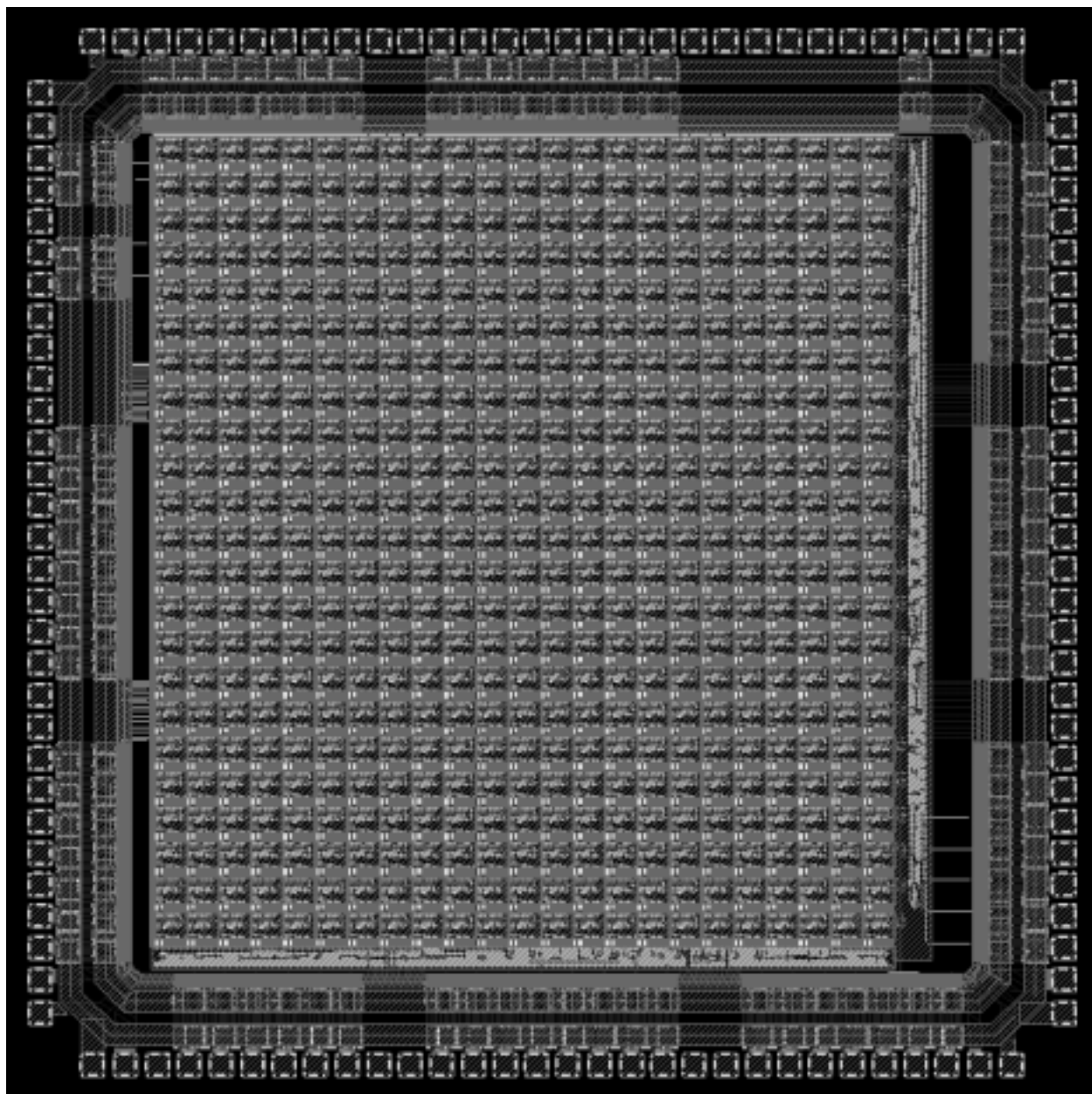


図 3.9 回路レイアウト図

### 3.5 この章のまとめ

この章で提案した回路によって、ノードに発生した電位を利用することにより重心の情報を検出することができた。またその動作速度は、出力する重心の数に多少影響されるが、ほぼ参照電位の周期である  $50[\mu\text{s}]$ での検出が可能である。従って、この章で提案する回路をロボットビジョンに応用することを考えた場合、一枚の画像を処理するのにほぼ  $50[\mu\text{s}]$ になる。これはロボットビジョンにおいて望ましいとされる  $1[\text{ms}]$ 以下を十分に満足した処理速度とである。画素数においても、現在主流である  $0.35[\mu\text{m}]$ のプロセスで  $9[\text{mm}]$ ダイのものを使えばおよそ  $100 \times 100$  個の画素を集積することができる。これはロボットビジョンにおいて一般的とされる画素数  $100 \times 100$  個を満たすことができた。またこの回路では、重心の情報を検出した際に、その時の参照電位を検出することによって、感知した領域の大まかな大きさを知ることができるため、ロボットビジョンとして広い範囲での応用が期待できる。

## 第 4 章

# 座標情報出力のための回路構成

以上に示してきた回路構成によって画素中に領域の存在が確認できた。そこで、検出した重心の情報をもとに座標情報を出力するための方法を考えなくてはならない。全ての画素に対して重心情報の検索をしたのでは、ここまでの回路によって実現した高速な処理が無駄になりかねない。そこでこの章において、高速に座標情報を出力する回路構成を提案し、その動作に関して説明を行う。

### 4.1 水平・垂直方向の論理和出力による座標情報の検出

ここでは水平・垂直方向への論理和出力によって座標情報を高速に検出する回路を提案し、その回路動作に関して説明を行う。図 4.1 に示すように、マトリックス上に配置した画素の各水平・垂直方向で重心情報の出力結果をカスケード式に論理和をとる構成を考える。前章までに示した回路による処理の結果に一つでも重心を示すフラグが存在した場合、その行および列の論理和の結果が 1 となって出力される。この結果をエンコードすることによって座標情報が出力される。この回路構成は組合せ論理回路であるので、非常に高速で座標情報を出力できる。しかも、通常画素数が増加すると検索する時間も増加するところであるが、図に示すように画素ごとで並列処理を行うため、検索時間が画素数に依存することはない、どの画素数に対してもほぼ一定の高速な時間で処理が可能である。2 章において提案した回路では、この座標情報出

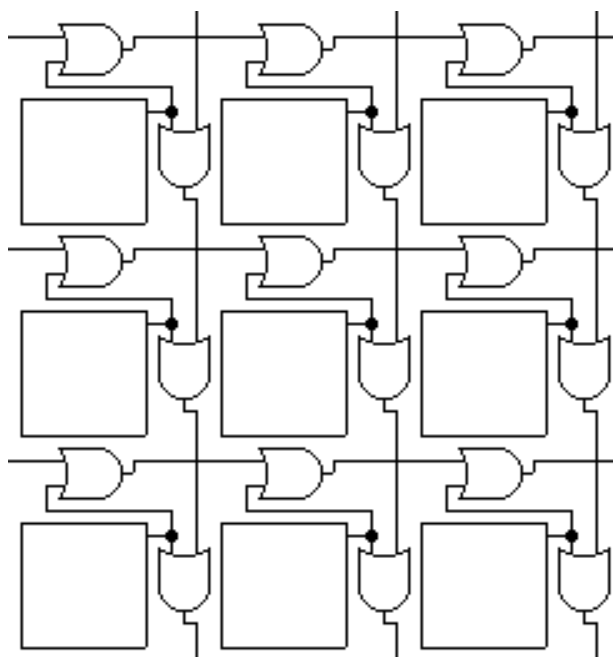


図 4.1 論理和出力による座標情報検出

力回路と組み合わせることにより、高速に座標情報を出力している。

しかしこの手法では一つの問題が考えられた。図 4.2 に示すように、重心を示すフラグが同時に検出された場合にその問題は発生する。

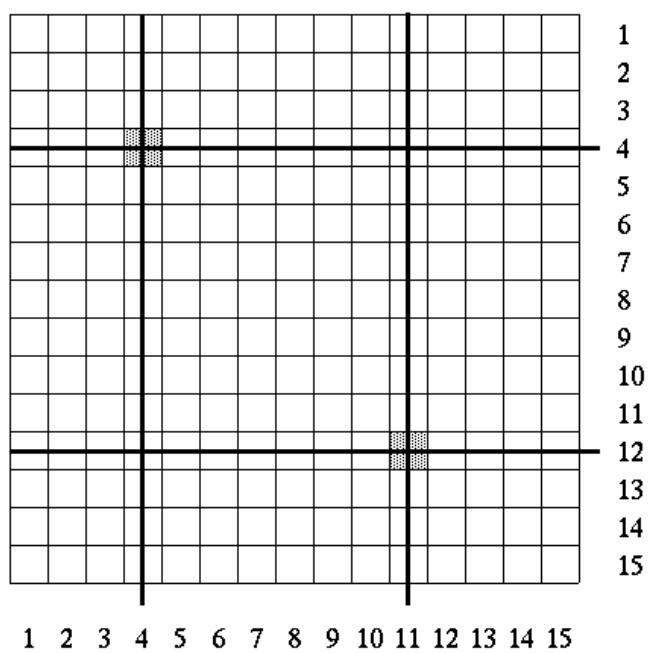


図 4.2 問題が発生する画像パターン

この座標情報出力のための回路においては複数の重心が同時に検出された場合に、出力された水平座標上と垂直座標情報がどの組み合わせをなすのかを決定することができない。たとえば図 4.2 で示す例では、出力が (4,4), (4,12), (11,4), (11,12) のどれが正しい情報であるのかが判定することができない。この現象が発生する理由として、検出された重心情報から座標情報を出力する際に、水平成分と垂直成分が関連付けされていないことが考えられる。従って、この座標情報出力回路では複数の重心が同時に検出されなければ正しく動作することが期待できるが、同時に複数の重心が検出された場合には、正しい座標を特定することができない。

同時に複数の重心が検出されたときに、正しい座標情報を検出するためには、この回路になんらかの改良を加えるか、もしくは新しい手法を用いなければならない。そこで次に、同時に複数の重心が検出された場合でも正しく座標情報を出力する手法を提案する。

## 4.2 検出点に対する逐次処理による座標情報の検出

同時に複数の重心が検出された場合に、1 ステップでその座標を出力するその他の手法として、各画素に特有の情報を持たせ、それを出力するのが手法があげられる。しかしこれでは画素数が多ければ多いほど特有となる情報のパターン数が増加するため、一画素の規模が大きくなるのは明らかである。ゆえに、1 ステップで同時に検出された重心の座標情報を正しく出力するには一画素の回路が大規模になりすぎ集積度の点で問題となり、スマートセンサには向かないと結論づけた。だからといって、重心を検出するために全画素に対して逐次的に処理をしたのでは、大幅な処理時間の増加が免れない。そこで新しい出力の手法として、逐次的に処理を行うにしても高速に動作できる手法を提案する。ここでは、通常的全画素に対して逐次処理を行う手法とは異なり、重心として検出されたノードに対してのみ逐次処理を施す回路構成を提案する。以下にその回路構成に関して説明を行う。

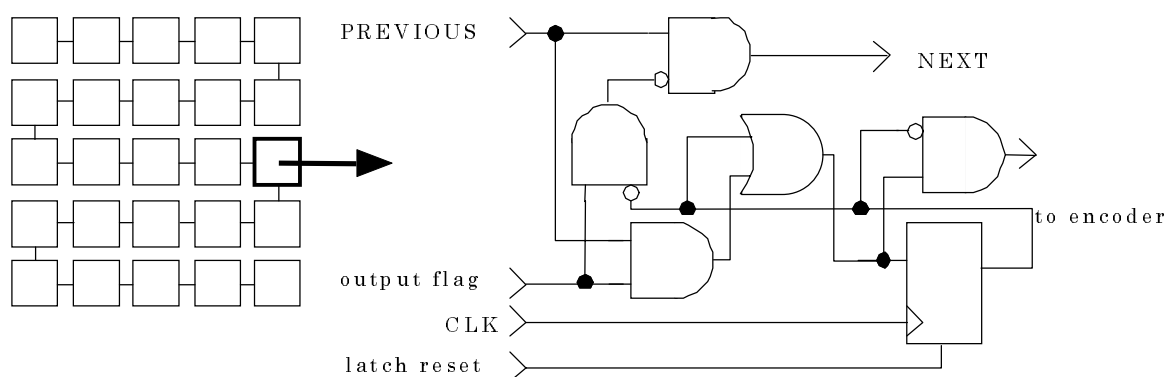


図 4.3 重心に対してのみ逐次処理を行う座標情報出力回路

#### 4.2.1 回路構成と回路動作

まず、ここで提案する座標情報出力回路の回路構成を図 4.3 に示しておく。ここに示す回路を各画素に組み込み、それを同図中に示すように一列に配置する。図中の **NEXT** は、次の画素に含まれるこの回路の **PREVIOUS** が接続され、前章までに提案した回路から得られる重心を示すフラグは図中の **output flag** に接続される。この回路は、一列につながる全ての座標検出回路の **PREVIOUS** を **0** にリセットしてから用いられる。重心の座標検索の開始を示す信号がきた時に、一列につながるこの回路の先頭にある **PREVIOUS** に **1** が入力されることで、重心の座標の検出を開始する。重心情報を示すフラグがない場合は、素通りして次の画素に含まれる座標検出回路の **PREVIOUS** に

1を入力される。ある点において重心を示すフラグが検出された場合には、重心の画素の含まれる PREVIOUS に 1 が伝播してきた時、回路に含まれるフリップフロップがクロックに同期して値が入力するまでの間、エンコーダに 1 の信号を渡し座標情報を出力する。この動作により同時に複数の重心が存在した場合、その後につながる画素に対して図 4.4 に示すようにマスクをかけた状態となる。従って同時に複数の重心が検出された場合でも、正しく座標情報を出力することができる。この回路では検出した重心の数だけのクロックしか必要としない。そのため、座標情報を出力するのに必要となる処理時間はクロック周期と重心の数の積で処理が可能であり、高速な検出が期待できる。

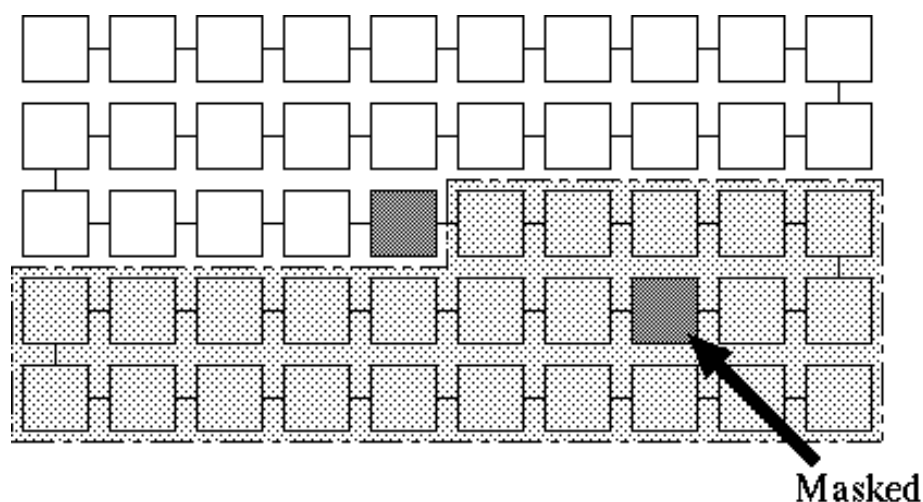


図 4.4 座標情報出力回路の動作

#### 4.2.2 電位分布の局所性を用いた重心検出回路との併用

3 章で提案した回路では、この座標情報出力回路を用いて重心の座標の検出を行った。この座標情報出力回路と 3 章で提案した回路を組みあわせるために、いくつか 3 章で述べた回路に改良を加えたのでそれについての解説をここでしておく。

先ずこの座標情報出力回路では座標情報を出力すべき画素を検索するわけだが、検索している途中で新たな重心を示すフラグが検出されてしまえば、正しい動作をすることができない。そのため座標情報を検出すべき点を検索し

ている途中は新たに重心を示すフラグが検出されないようにしなければならない。そこで、4章ののこぎり波形生成回路の説明で少し触れた通り、参照電位を検索が終了するまでの間停止させることでこの問題が解決できる。

すなわち、参照電位を一時停止させ、検索終了時に再開させるための信号が必要となる。先ず参照電位を停止させるための信号についてだが、これは重心を検出した時に止めなくてはいけない。そのために、3章で示した回路の出力結果を全画素に対して論理和をとることにした。これにより、重心が検出されるとすぐに論理和ゲートは1を出力し、参照電位を停止するためのタイミングを知ることができる。しかし、この方法では複数の領域が存在する場合に対処することができない。というのも、先に1つでも重心が検出されてしまえば、その後に検出される重心のタイミングを知ることができないためである。そのため、図4.5に示すような回路を各画素ごとに接続し、この回路を通して得られる結果を、全画素に対して論理和をとることによって問題を解決できる。座標情報出力回路に含まれるフリップフロップは、そのノードが一度重心であると検出されてしまえば、リセットされるまで値を保持するため、このように接続することによって検出が終了した時点で、論理和の出力を0に戻すことができる。これにより、感知した領域が複数ある時にでも対処することができる。

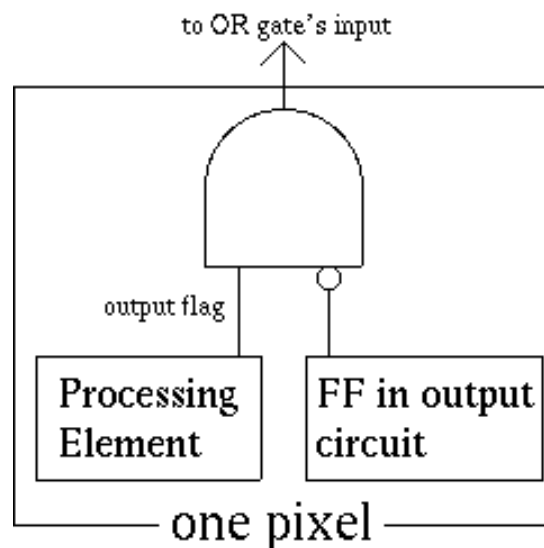


図 4.5 参照電位を停止させるための信号



次に参照電位を再び下げ始めるタイミングについてであるが、そのためには座標情報出力回路が検索を終了したタイミングを知る必要がある。座標情報出力回路は一系列につながる先頭の PREVIOUS から最後につながる座標情報出力回路の NEXT に 1 を伝播させることで検索が終了する。従って、一系列に接続された座標情報出力回路の最後にある NEXT から 1 が出力された時が検索の終了した時である。従って、この信号を用いることで、参照電位を下げ始めるタイミングを知ることができる。

以上によって参照電位を停止および再開させることができる。しかし、参照電位を停止させた時にすぐに座標情報を検出すべきノードを検出することができない。これは、セトリングタイムを考慮してのことである。また、参照電位を下げ始めるタイミングについても、検索が終了した時点ですぐに参照電位を下げ始めたのでは、座標情報出力回路を正しく動作させることができない場合がある。それは、複数の領域が存在し、各重心の電位に差がほとんどない場合である。ここで提案している座標情報出力回路は、先にも示した通り一系列につながる全ての座標検出回路の RPREVIOUS を 0 にリセットしてから用いなければならない。そのため、リセットし終わる前に次の重心が検出されてしまつては、正しく回路を動作させることができないからである。

これらの点を解決するために、検出を開始するタイミングを 1 クロック分遅らせ、また検索が終了してから全画素の座標情報出力回路の PREVIOUS を 0 にセットし直すまでの時間として 1 クロック分だけそれぞれ遅らせる。本研究ではこれを実現するための回路を、Verilog HDL を用いて回路の作成を行い、それを回路に組み込むことにより解決をした。

#### 4.2.3 検出点に対する逐次処理座標検出回路の応用

ここで、この座標情報出力回路の応用した形を考える。この回路では先にも示した通り、各画素に含まれる座標情報出力回路を一系列に並べ、先頭の PREVIOUS から最後の NEXT に 1 を伝播させることで座標情報を出力すべき点を検索するのであるが、この構成では画素数を増加させた場合に、直列につなげた回路の伝播遅延が増加してしまう。試作に用いたプロセスでは、1 つ

の座標情報出力回路に 1 を伝播させるのに  $180[\text{ps}]$  の時間がかかるのであるが、仮に 100 万画素に対して検索をした場合、それだけで  $180 \mu\text{s}$  の時間を要することになる。これでも十分に高速であるとは言えるが、多くの領域を検出する場合を考えると、そのたびに  $180 \mu\text{s}$  の時間がかかっては高速とは言えなくなってくる。そこで、ここで提案している座標情報出力回路を応用して、各画素に組み込んだ座標情報出力回路を図 4.6 に示すように接続した。先ほどまでの一列に並べる方式と違い、図のようにつなげることにより、各行ごとの並列処理が可能となり、より高速な動作をすることができる。ただし各行で並列に処理を行った場合には、複数の領域が同時に検出された時にこの回路を正しく動作させることができない。そこで、どこまでの行が検出し終わっているかを各回路に情報として渡しておかなければならないのだが、これは図に示す前行の論理積の結果を各画素に配置される座標情報出力回路に入力することで知ることができる。

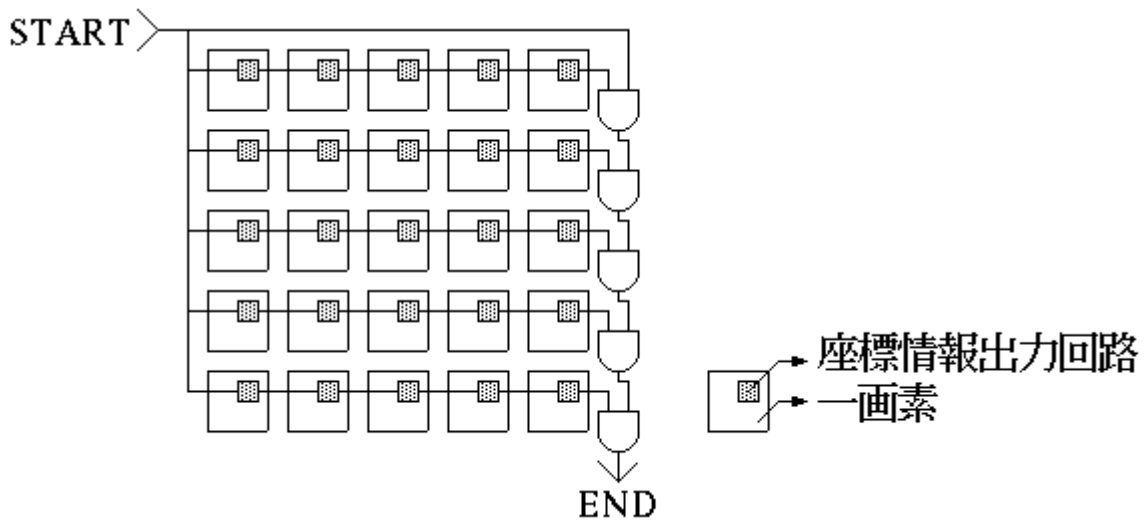


図 4.6 座標情報出力回路の応用

### 4.3 この章のまとめ

この章では、2 章及び 3 章で提案した回路から出力される重心の情報を検出し、それを座標情報に変換するための回路構成を 2 つ提案した。水平・垂直方向の論理和による座標情報出力回路では、比較的簡単な組合せ論理回路で構成しているため、非常に高速な検出をすることができた。しかしながら、同時に複数の領域が検出された場合には、回路では座標情報を正しく検出することができないという欠点があった。したがって、この回路では同時に複数の領域を検出しないという前提のもとで、非常に高速に座標情報を検出することができると言える。

次に、検出点に対する逐次処理による座標情報出力回路では、全画素に対して逐次処理を行う手法とはことなり、重心と検出された点に対してのみ、座標情報を出力するための処理を行うため、その処理速度は検出された重心の数とクロック周期の積で表わすことができ、非常に高速な検出が可能である。また、水平・垂直方向の論理和による座標情報出力回路において問題であった、同時に複数の領域が検出された場合に発生する問題においても、逐次処理という手法を取ることにより解決することができた。従って、この検出点に対する逐次処理による座標情報出力回路ではどのような場合においても非常に高速に座標情報を検出することができる。またこの回路を応用して、より高速に座標情報を出力する手法も提案した。

## 第 5 章

### まとめ

本研究では、近隣のノードと接続する抵抗網において、各ノードに接続されるフォトダイオードから得られる光電流によって決定されるノードの電位を利用し感知した領域の重心とその大きさを検出するためのスマートセンサを実現すべく 2 つの回路構成を提案した。以上を持ってその回路の原理や動作、その回路構成に関する報告を終了とするわけだが、その前にこの二つの回路を実際にロボットビジョンとして応用が可能であるかをそれぞれ検証する。

2 章で提案した回路では、各ノードの状態によって重心の情報を出力するため、動作は極めて高速なものなり、1 枚の画像に対して約 15[ns]で検出が可能であることが分かった。ロボットビジョンでの応用を考えた場合、1 枚の画像にかけられる時間が 1[ms]以下が望ましいとされているので、この処理時間を十分に満たしていることが分かる。また画素数に関しては、現在主流となっている 0.35[ $\mu\text{m}$ ]のプロセスで 9mm ダイのチップを用いれば、約 48x52 の画素を集積することができる。ロボットビジョンで一般的とされている画素数 100x100 には至らないが、100x100 以下の画素でも利用用途によっては十分に応用が可能であることが多いため、利用用途によっては、この回路も十分に応用が可能であると考えられる。

3 章で提案した回路では、極大の電位を判定する手法を 2 章の手法と異なるものを提案し、より簡単な回路構成で重心の検出を可能とした。1 枚の画像を処理するのに、検出する重心の数にもよるが、ほぼのこぎり波形の周期である

50[ $\mu$  m]で検出が可能であることが分かった。ロボットビジョンへの応用を考えた場合、先にも延べた処理時間を十分に満たした処理時間で検出が可能である。また画素数については、先ほどと同じように考えれば約 100x100 の画素を配置することが可能である。また、検出した重心のノードの電位から感知した領域の大まかな大きさも知ることができるため、ロボットビジョンとして広い範囲での応用が可能であると言える。

また出力された重心のフラグから、高速にその座標を出力する回路も 2 つ提案した。このことに関してもまとめておく。水平・垂直方向へ論理積による座標出力では、組合せ論理回路で非常に簡単な構成により実現できる。また、各画素ごとで並列に処理することが可能であるため、画素数には依存せず高速な検出が可能であることが分かった。だが、複数のフラグが同時に検出された場合には、正しい情報を特定することができないという問題があった。従って、同時に複数のフラグが検出されないという条件の下で、高速な検出が可能であると言える。

次に、検出点に対する逐次処理による座標出力回路では、全画素に対して逐次処理を行う手法とは異なり、重心と検出された点に対してのみ座標情報を出力するための処理を行う。検出点に対して逐次処理を行うことによって水平・垂直方向への論理和出力回路で問題となった点を解決することができる。動作速度に関しては、検出点の数分のクロック数しか必要としないため非常に高速な動作が可能である。

## 付録

本稿で紹介した回路以外で、著者が博士過程前期の2年間でチップ試作を行った回路について紹介する。図Aに示す写真はCCDカメラ等から送られてくる画素ごとの色情報から、特定の色を抽出する回路を集積したチップの外観である。



図A 著者が設計したチップ,リアルタイム色抽出処理回路(モトローラCMOS1.2 $\mu$ mプロセス、チップサイズ2.3mmダイ)

この回路について簡単に説明をしておく。CCDカメラ等から送られてくる画素ごとの色情報に対して、それが対象の色であるかどうかを判断し、2値化した画像に変化する。その結果を、この後段に接続するラベル付けや形状認識の処理を行う処理系に渡す。このようなシステム構成によって色抽出を行うことにより、画素情報をメモリに取り込む必要がなく、色抽出の計算処理にかかっていた時間を大幅に低減することができる。

図Bに、この色抽出回路の論理ブロック図を示す。R、G、B、の各色に対し

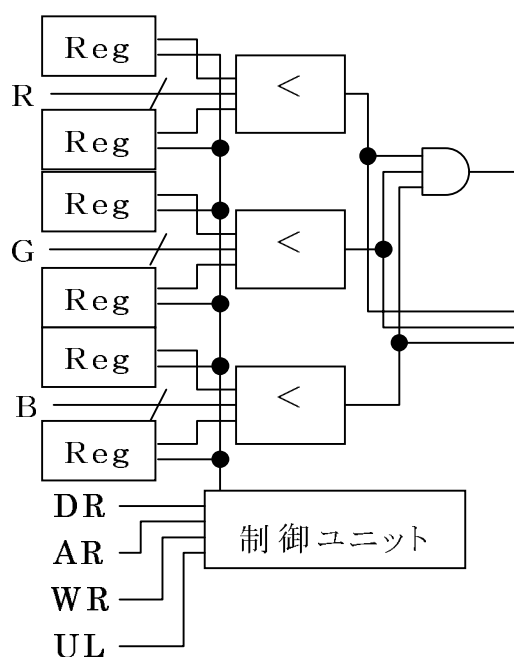


図 B 色抽出処理回路の論理ブロック図

でそれぞれ2つのレジスタを持ち、それぞれ、設定したい色の許容上限値と許容下限値を設定する。出力は、入力されたRGB値がこの2つのレジスタに示される値の範囲内にあるかどうかによって決定される。この回路の入出力は以下のようにになっている。

**[入力] RGB(各 6bit):**カメラから得られる画素ごとの値

**DR(6bit):**レジスタに設定するデータの入力

**AR(2bit), UL(1bit):**レジスタ選択信号

**WR(1bit):**書き込み信号

**[出力] 各色が設定範囲内にあることを示すフラグ(各 1bit)**

**RGB すべてが設定範囲内にあることを示すフラグ(1bit)**

なお入力の色情報は RGB 系以外に限らず、照明条件の変化に強い、色相・彩度・強度を用いる HSV 系などでもよい。

以上のような回路を Verilog HDL で設計を行った。セル数は163個で、動作速度は 10.3ns であった。

## 参考文献

- [1] A.Gruss *et al.*, "A VLSI Smart Sensor for Fast Range Imaging," *Proc, IEEE Int. Conf. on Intelligent Robots and Systems*, 1992.
- [2] J.E,Eklund *et al.*, "VLSI Implementation of a Focal Plane Image Processor-A Realization of the Near-Sensor Image Processing Concept," *IEEE Trans. VLSI SYSTEM*. Vol.4, No.3, pp.322-335, Sep.1996.
- [3] G.A.Horridge and P.Sobey, "An artificial seeing system copying insect vision," *INTERNATIONAL JOURNAL OF OPTOELECTRONICS*. Vol.6, NOS.1/2 177-193.19. 1992.
- [4] W.Bair & C.Koch, "Real-time motion detection using an analog VLSI zero crossing chip," *Proc. SPICE, Visual Information Processing : From Neurons to Chip*, Vol.1473, pp. 59-65, 1991.
- [5] L.Dron, "Multiscale veto model: a two-stage analog network for edge detection and image reconstruction," *International Journal of Computer Vision*, Vol.11, No.1, pp.45-61, August 1993.
- [6] A.Åstrom *et al.*, "Global Feature Extraction Operations for



Near-Sensor Image Processing," *IEEE Trans. Image Processing*.  
Vol.5, No.1, Jan.1996.

[7] C.Mead, "Analog VLSI and Neural Systems" *Resistive  
Circuit*, p134- p136, 1993.

[8] David A. Johns, Ken Martin, "Analog Integrated Circuit  
Design" *Chapter7, Comparators* p304-p307, 1991

## 口頭発表

- 高瀬信二,秋田純一,前多和洋,北川章夫,鈴木正國,”ビデオ画像の画素単位でのリアルタイム色抽出処理回路の設計”,電子情報通信学会春季総合大会,SA-2-1,1999年3月,慶応大学日吉キャンパス

## 謝辞

本研究を行うにあたり、多くの方々に御助言、ご協力を頂きました。この場をお借りして感謝の意を表したいと思えます。様々な面で御助言、御指導をしていただきました故鈴木正國教授に心から感謝いたします。研究、生活面におきまして、大変お世話になりました北川章夫助教授に深く感謝いたします。研究の指導、生活面におきましても大変お世話になりました秋田純一助手に深く感謝いたします。2ヶ月と短い間ではありましたが、研究面でアドバイスをして頂いた深山正幸助手に深く感謝いたします。機器の発注や生活面で御指導いただきました柿本芳雄技官に深く感謝いたします。

VLSI設計室の管理やUNIXの操作指導をして頂いた集積回路工学研究室卒の中山和也助手に感謝いたします。研究面でも生活面でもよき相談相手となって頂いた共同研究者の渡辺晃氏、遠山治氏に深く感謝致します。3年間の研究生活を有意義なものにし、よきアドバイスや相談にのって頂いた小川明宏氏、中橋憲彦氏、早川史人氏、藤井直樹氏に感謝致します。生活面でも研究面でもアドバイスをして頂いた今井豊氏、数馬晋吾氏、藤田隼人氏、水野浩樹氏、村上崇氏に感謝いたします。また、1年と短い間ではありましたが、同じ研究室として研究を行ってきた高松直樹氏、中村公亮氏、大門慎治氏、佐々木勝光氏、笠井稔彦氏、蓮達弘氏、水木誠氏、辻川隆俊氏に感謝いたします。大学でのLSI設計を可能にして頂いた大規模集積システム設計教育研究センターに感謝いたします。

最後に、任期半ばにしてお亡くなりになった故鈴木正國教授の御冥福をお祈りいたします。