2次元抵抗網中の電位分布の局所性を用いた高速重心検出回路の構成に関する研究

指導教官
鈴木正國教授
北川章夫助教授
秋田純一助手
深山正幸助手

提出者
金沢大学工学部
電気・情報工学科
遠山 治

提出日　平成 12 年 2 月 23 日

目 次

第1章　序論

1.1　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。しかし、近年の技術発展に伴い画像の高解像度化や扱うフレーム数が増加し扱う情報量も増加している中、これらの処理をリアルタイムで行おうとした場合、後段に続く処理系での処理時間があるため画像データを転送する時間は高いフレームレートを必要とする場合これによって上限が決められ、多量のデータを転送することは、困難なものとなってくる。また、半導体技術の発展により、処理系のデータ処理能力は飛躍的に向上して来たが近年ムーアの法則にそった発展にはかげりが見えはじめ、いずれは集積度やクロックの速度の発展は飽和状態となることが予想される。そのため、さらなる性能向上のためにはこれまでの処理方法と異なる革新的な手法を考える必要がある。

1.2　スマートセンサ

この問題を解決する１手法として、集積回路技術の進歩から、受光系に処理系の一部を集積した、いわゆるスマートセンサと呼ばれるものが現実のものとなって来た。[1] スマートセンサは画素単位で受光素子と処理系を組み込むため、画素ごとに並列処理ができる上に、受光系から処理系へのデータの転送も高速に行える。また、システムの小型化などの利点も有る。したがって、ロボットビジョンなどリアルタイムでの画像処理が望まれる分野での応用が十分に考えられる手法である。

1.3　本研究の目的

本研究では、先ほどのスマートセンサの手法を用いて、感知した領域の重心とその大きさを検出するということに焦点を絞り、この機能を実現する回路構成を提案する。またこれをスマートセンサとして実用的なレベルのものとして実現する。そして現在、ロボットビジョン等リアルタイムでの処理が必要とされる分野で望まれている画像処理能力との比較を行い、設計した回路の評価をする。

第2章　重心の検出

2.1　重心検出の原理

図 2.1 のように配線した抵抗網の各節点にフォトダイオードを接続し、そこから流れる光電流による各接点の電位分布を調べると図 2.1 のようになる。実際は 2 次元抵抗網を用いるのだが説明の繁雑を避けるため、ここでは 1 次元モデルを用いる。この電位分布において極大の電位を持つノードは、光を受けた幾つかのフォトダイオードの集まりの重心のノードとほぼ一致することがわかっている。[2] この局所性に注目し、フォトダイオードから流れる光電流によって得られる電位分布から、極大値を持つノードを検出することで、重心の場所情報を得ることができると考えられる。

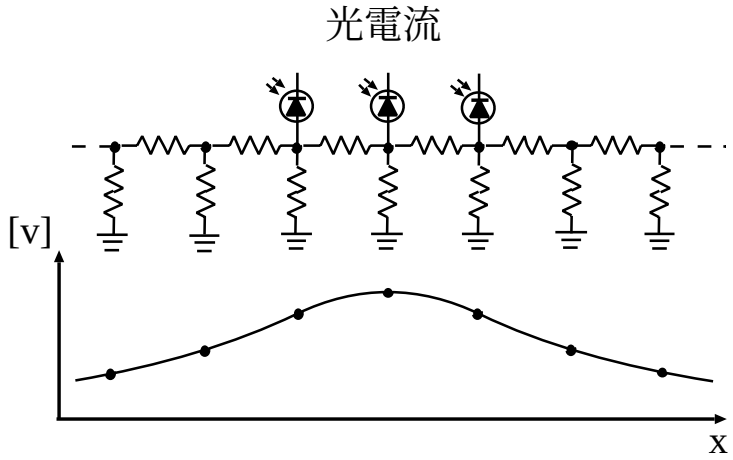

図 2.1: 光電流と抵抗網による電位分布 (1 次元モデル)

また、極大の電位の値から感知した領域の面積が分からないだろうかと考え、これらの間の関係を調べるために図 2.2 に 2 次元モデルにおいて、入力として与える領域の縦横比を様々に変えて、発生する極大の電位を調べた、C 言語によるシミュレーションの結果を示す。この結果から、縦横比が大きく違う場合には、面積が同じであっても大きく異なる電位の値になるが、縦横比が比較的近い場合にはほぼ同じくらいの電位が得られることが分かった。実用上、縦横比の大きく違うものの認識をすることは少ないと考えられるので極大となる電位の値を調べれば、感知した領域の面積のおおよその大きさが分かると考えられる。

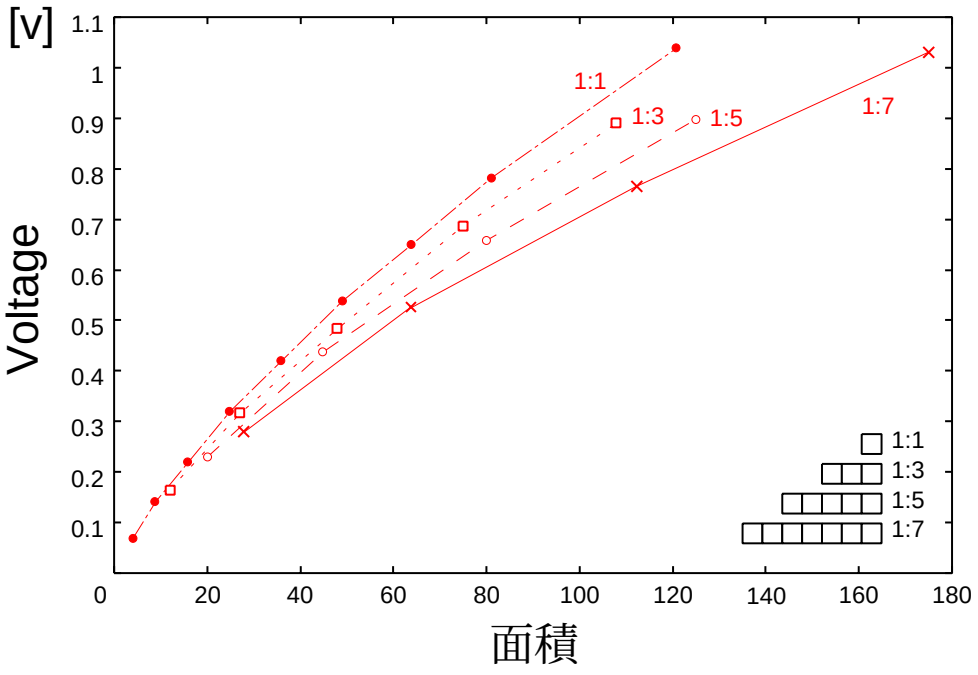

図 2.2: 入力する領域の縦横比と発生する電位の関係 (2 次元モデル)

2.2　極大の電位を持つノードの検出

今回提案した重心検出と大きさ検出の機能を持った回路を図 2.3 に示す。ここでは各部の細かな説明は省き大まかな検出の流れを説明する。

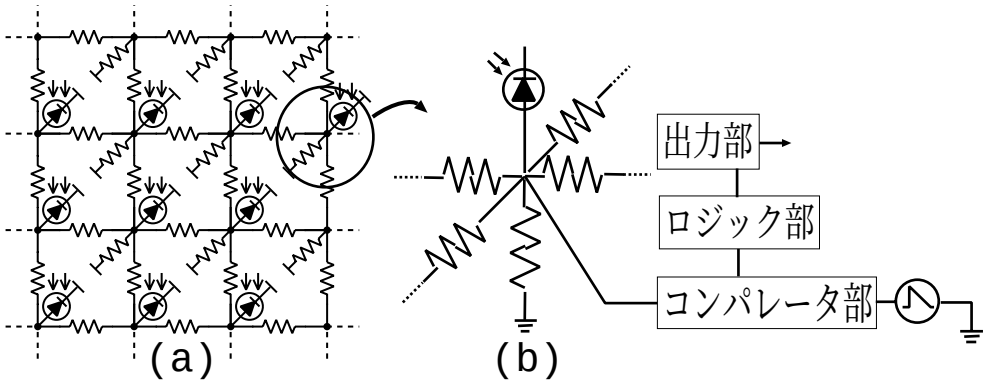

図 2.3: 今回提案した回路　(a) 全体の構成 (b)1 画素の構成

極大の電位を持つノードは、そのノードの電位がその 4 近傍のノードの電位よりも高くなっている。今回提案した回路では、これを検出するための仕組みをコンパレータ (図 2.3 のコンパレータ部) を用いて実現することを考えた。まず、ノードの電位の分布が 0v から 5v の間におさまるよう抵抗の値を調節する。ここで、コンパレータの一方の入力を各ノードの電位、もう一方を一定周期で電圧が 5v から 0v まで減少するノコギリ波とする。ノコギリ波は比較のための基準電位として全てのコンパレータに共通に入力され、これによって、すべてのノードについて並列に比較を行うことができる。

あるノードの電位を V_n、その近傍のノードの電位を V_{n-1}, V_{n+1}、ノコギリ波の電位を V_{ref} とする。ここで、ノコギリ波の電位を徐々に下げることで各ノードの電位を調べていくと、極大の電位を持つノードでは、図 2.4(1 次元モデル) に示すように、あるタイミングで次の関係式を満たす。

$$V_{n-1}, V_{n+1} < V_{\mathrm{ref}} < V_n$$

この式をみたす各ノードの電位情報は、極大の電位を持つノードにつながるコンパレータとその近傍のコンパレータの出力から ‘1’,‘0’ の組合せとして得られるので、この組合せだけ 1 を出力する論理回路 (図 2.3 のロジック部) を用いれば、極大の電位を持つノード、つ

まり光を受けたフォトダイオードの集まりの重心位置を検出できると考えられる。

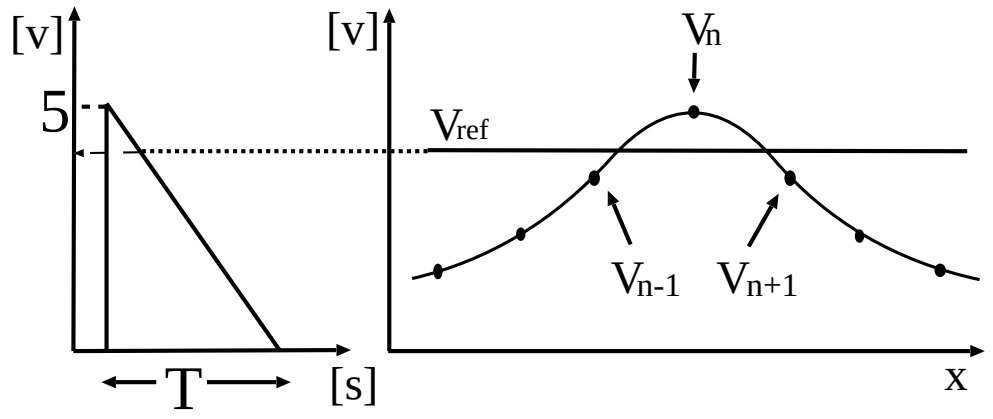

図 2.4: コンパレータとノコギリ波を用いた検出

2.3　受光部と抵抗網

受光部となるフォトダイオードと電位分布を得るための抵抗の値は今回考えた回路の基礎を成す部分であり、重要な部分である。以下にこれらの値をどのように決定したかを説明する。

実際のフォトダイオードは p 基板上に n ウェルをつくり、そこに p 拡散を作ることで実現され、この構造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変化する。しかし今回考えた回路では、この部分はセンサーの 1 画素分の入力を構成するので、センサーの解像度と得られる電流のトレードオフを考えると、やはり、チップの面積は限られているので、ここで大きな面積を割くわけにはいかない。そこで $10[\mu\mathrm{m}]\times10[\mu\mathrm{m}]$ 程度の大きさで、得られる電流が $10^{-6} \sim 10^{-7}[\mathrm{mA}]$ 程度であるものを用いることとした。フォトダイオードから得られる光電流はこのように非常に小さい値であるので、後段のコンパレータを駆動するための電位を抵抗網の各節点部分から得るためには、グランドにつながる垂直方向の抵抗として比較的高いものを用いる必要がある。一般に高抵抗 (ただし線形特性を持つもの) をチップ上に実現するにはトランジスタの線形領域となる部分を用いるわけだが、この手法によって得られる抵抗は高抵抗なものほど面積が大きなものとなる。今回のように網目状に抵抗をいくつも配置する必要がある場合、チップ上に実装することを考えると、面積と集積度の関係から $1\mathrm{M}[\Omega]$ 程度が限界である。しかしこれでもコンパレータを駆動するための電圧を得るためには、まだ不十分である。そこで、足りない分は光電流を増幅することで補うこととした。垂直方向の抵抗の値が $1\mathrm{M}[\Omega]$ と決まった上で、水平方向の抵抗の値を決める。この値によって、得られる電位分布が変わる。(図 2.5,2.6,2.7)

• 垂直方向の抵抗に比べ、水平方向の抵抗が低い場合

この場合は横方向のつながりが強くなりフォトダイオードから流れ出る電流は垂直方向の抵抗へとはあまり流れていかず、周りへと捌けていくようになるので、図 2.5 のように比較的つぶれた分布になる。

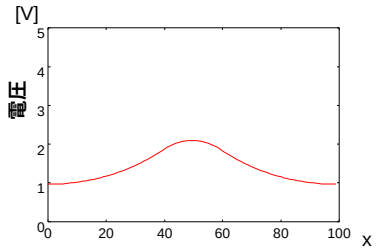

図 2.5: 水平方向の抵抗が垂直方向の抵抗に比べ低い場合

• 垂直方向の抵抗に比べ、同じくらいかそれよりも高い抵抗の場合
この場合は垂直方向のつながりが強くなり、電流の大部分はそのまま垂直方向の抵抗へ流れていくので図 2.6 のように、台形状の電位分布になる。

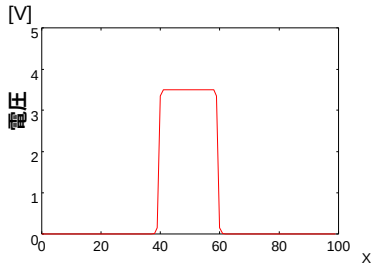

図 2.6: 水平方向の抵抗が垂直方向の抵抗に比べ高い場合

次段のコンパレータ部で、ある点とその 4 近傍との電位の違いをノコギリ波の電位との比較によって調べることを考え、近隣同士の電位の差が十分広くなるように、特に極大点の付近において、近隣同士の電位が近い値にならないようにする必要がある。したがって図 2.7 のような電位分布になるように、水平方向の抵抗を決める必要があり、以上の条件を満足する値を C 言語を用いて様々なシミュレーションを試した結果から水平方向の抵抗は 3.3k[Ω] とした。

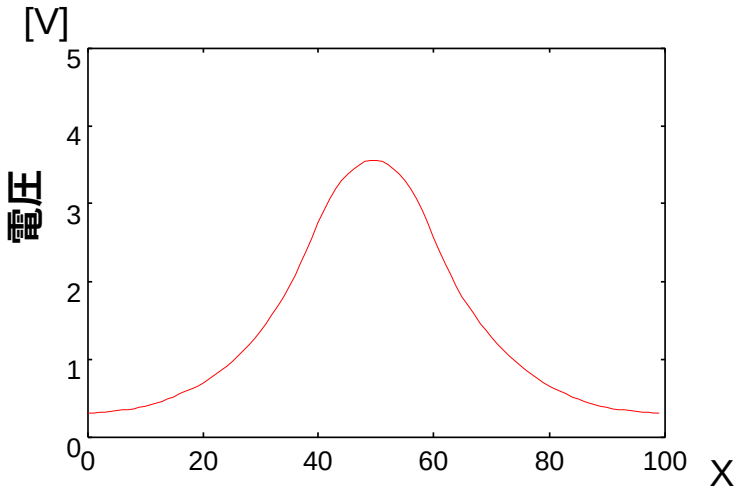

図 2.7: 理想的な電位分布

H-SPICE によるシミュレーションで用いた受光部と抵抗網の回路を図 2.8 に示す。フォトダイオードは、電流源とダイオードを並列につないでこれを等価回路とした。

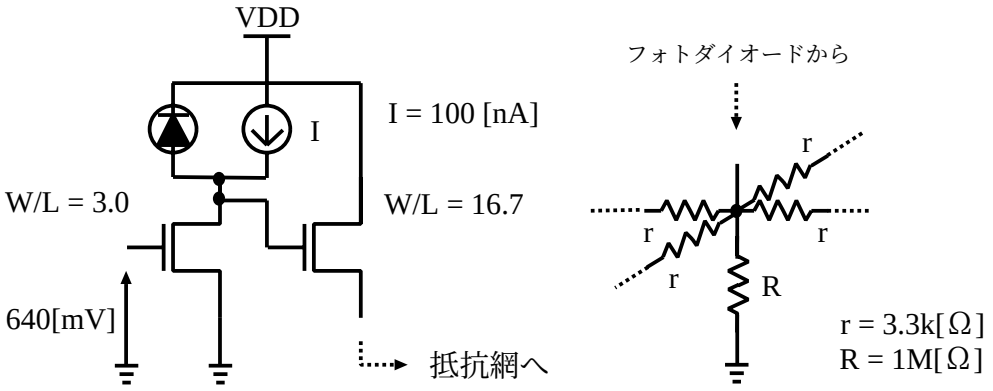

図 2.8: フォトダイオード部の等価回路

2.4　コンパレータ部

この部分は 2.2 節で説明したように、あるノードとその 4 近傍のノードの電位の違いをノコギリ波を比較の基準電位とし、それより大きいか、小さいかを調べることで検出する部分である。コンパレータの一方には一定周期で電圧が 5v から 0v まで減少するノコギリ波が入力されており、これを基準電位として各ノードの電位が比較される。あるノードの電位を V_n その近傍の電位を V_{n-1}, V_{n+1} ノコギリ波の電位を V_{ref} とすると、極大の電位を持つノードではあるタイミングで次の関係式を満たす。

$$V_{n-1}, V_{n+1} < V_{\mathrm{ref}} < V_n$$

この各ノードの電位の大小情報はコンパレータ部からロジック部へ '1','0' の情報となって渡される。
また、後段のロジック部を構成しやすいようにコンパレータの出力 Cmp_out は

$$Cmp\_out = \begin{cases} 1 & (V_n < V_{\mathrm{ref}}) \\ 0 & (V_n > V_{\mathrm{ref}}) \end{cases}$$

とした。

2.5 ロジック部

ロジック部には各ノードとその近傍のコンパレータ部の出力が接続されており、この出力の組合せから、各ロジック部につながるノードが極大値を持つ点であるかどうかを判定する。判定されるタイミングは、各ロジック部につながるノードがノコギリ波の電位よりも低くなった時、つまり各ロジック部につながるコンパレータの出力が '1' から '0' となった時である。前段のコンパレータの出力から考えられる、ロジック部が満足すべき真理値表は、以下のとおりである。（ただし簡単のため 1 次元モデルでの場合）Cmp_out_n はある点のコンパレータの出力、Cmp_out_{n-1}、Cmp_out_{n-1} はその両隣のコンパレータの出力。

Cmp_out_{n-1}	Cmp_out_n	Cmp_out_{n+1}	出力	出力の意味
1	0	1	1	極大点
0	0	＊	0	非極大点
＊	0	0	0	非極大点
＊	1	＊	0	-

表 2.1: ロジック部の真理値表

時間を追ってコンパレータの出力 Cmp_out を見ると、Cmp_out は高い電位を持つノードから順番に '0' になるので、極大の電位を持つノードであってもあるタイミングでは非極大点と判定される組み合わせを持つ。極大点であると判定した後に非極大点であると再判定されてしまっては意味が無いので一度出力した値は組合せが変わっても保持している必要がある。この問題は出力からのフィードバックを設けることで解決できる。以上のことから考案したロジック部の回路構成を図 2.9 に示す。

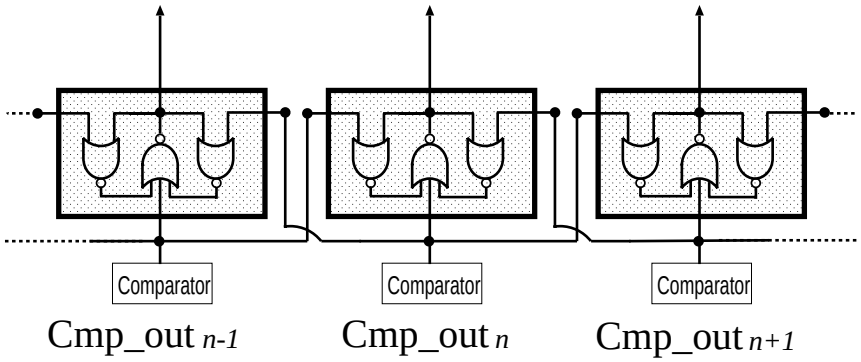

図 2.9: ロジック部の回路構成

2.6　ノコギリ波発生回路

今回考えたノコギリ波を発生させる回路を図 2.10 に示す。充電したコンデンサーから電流源を用いて一定量の電流（電荷）を引き抜くことで、ノコギリ波を作る。この部分は次の章の場所情報を抽出する回路との関連から、途中でノコギリ波が減少するのを止める機構が必要となった。そこで、電流源とコンデンサーとの間にスイッチ (図 2.10 の SW) を設け、これでノコギリ波を制御している。また、ノコギリ波が止まったときの電位を調べることで、おおよその領域の大きさを知ることができる。

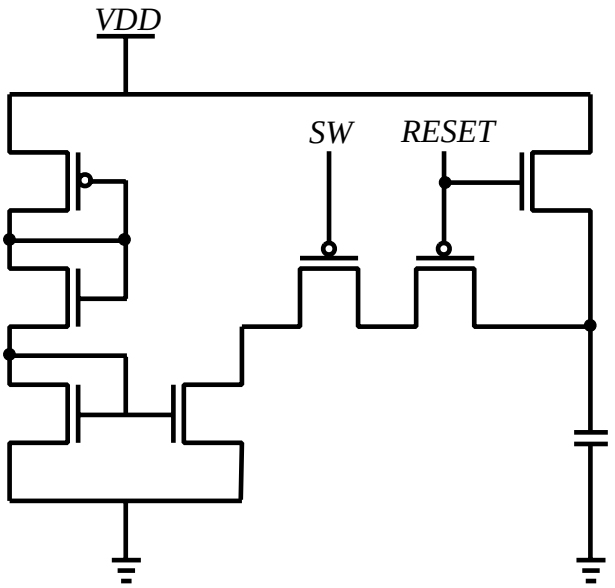

図 2.10: ノコギリ波発生回路

ノコギリ波の周期は全画素を比較するのにかかる時間、つまり画像中にある全ての領域の重心を検出するのにかかる時間、となるので回路の動作と仕様を決める重要な部分となる。ノコギリ波の周期を高くすればそれだけ回路の性能が高くなるわけだが、高すぎると回路の動作がこれに追い付かず、誤動作をすることがある。誤動作をする可能性は隣り合うノード間、例えば図 2.11 の A,B 間で、A のロジック部が極大の判定を行っている間にノコギリ波が減少していき、B の電位まで達してしまい、B のロジック部の出力が変化する場合である。したがって、この周期 (ノコギリ波の傾き) の限界をきめる ΔX,ΔY は今回の場合それぞれ、ロジック部の遅延、近隣ノードの電位差の最小値に対応する。

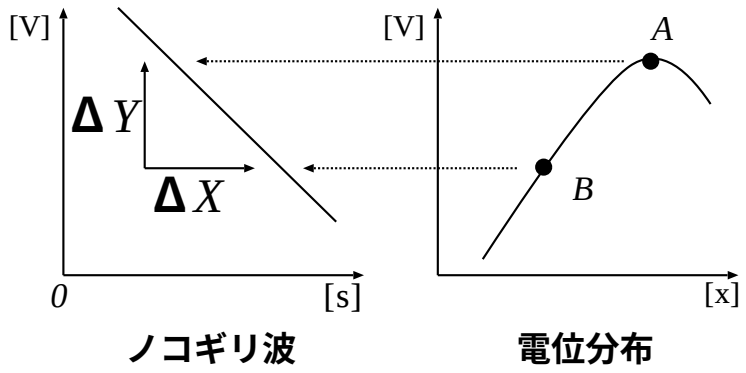

図 2.11: ノコギリ波の傾き

- ΔX について

2.5 節で考えたロジック部の論理回路において、出力が誤りとなる可能性は先ほども述べたように、図 2.11 において、A のロジック部が極大の判定を行っている間にノコギリ波が減少していき、B の電位まで達してしまい、B のロジック部の出力が変化する場合であるがロジック部の内部を詳しく見るとこの部分が誤動作を起こす可能性は正確には図 2.12 の

1. A の出力が決まる前に B から A に入力が来る場合
2. B の出力が決まる前に A から B に入力が来る場合

のどちらかである。したがって、A、B どちらか遅い方の動作速度にあわせてやればロジック部が誤動作をすることは無いことが分かる。以上を H-SPICE でのシミュレーションによって調べたところ、遅い方 (B) の動作速度が約 600[ps] であることが分かった。

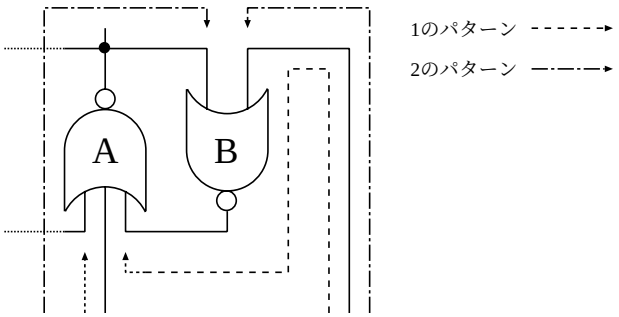

図 2.12: 誤りの可能性

- ΔY について

最も近隣ノードの電位差が無くなるのは、H-SPICE によるシミュレーションから、全てのノードのフォトダイオードが反応しているときであることがわかった。このときの極大点付近での近隣ノードの電位差はおよそ 0.1[mV] であった。

以上より、計算上 30[μs] の周期でこの回路の正確な動作が見込めるが、ある程度の余裕を持たせてノコギリ波の周期は 50[μs] と定めた。

2.7　重心検出のシミュレーション

この章で述べた回路を画素数が 15×15 として、H-SPICE による回路シミュレーションを行った。図 2.13,2.14,2.15 にその時の電位分布と重心として検出された画素を示す。全体の画素数が少ないため、入力する領域の一辺の長さが偶数になる場合はやや曖昧になるが、どの領域もほぼ重心と思われる画素が検出された。これまで述べてきた回路構成では、全画素の処理は完全に並列であるため、この回路の動作速度 50[μs] という周期は、理論上画素数に依存しない。したがって、従来の逐次処理と比べ十分高速に重心の検出を行うことができていると考えられる。

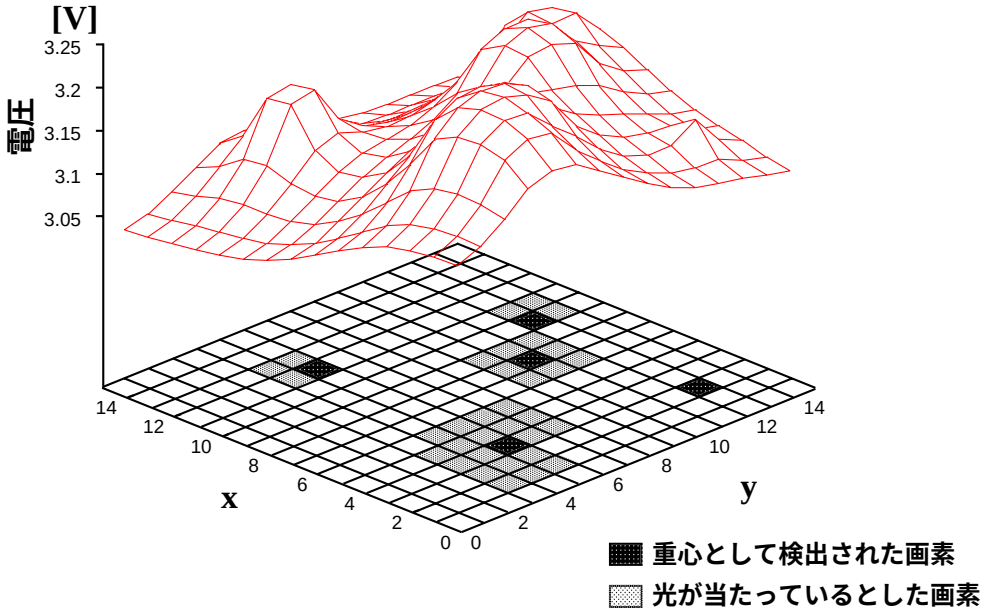

図 2.13: 電位の分布と重心検出 1

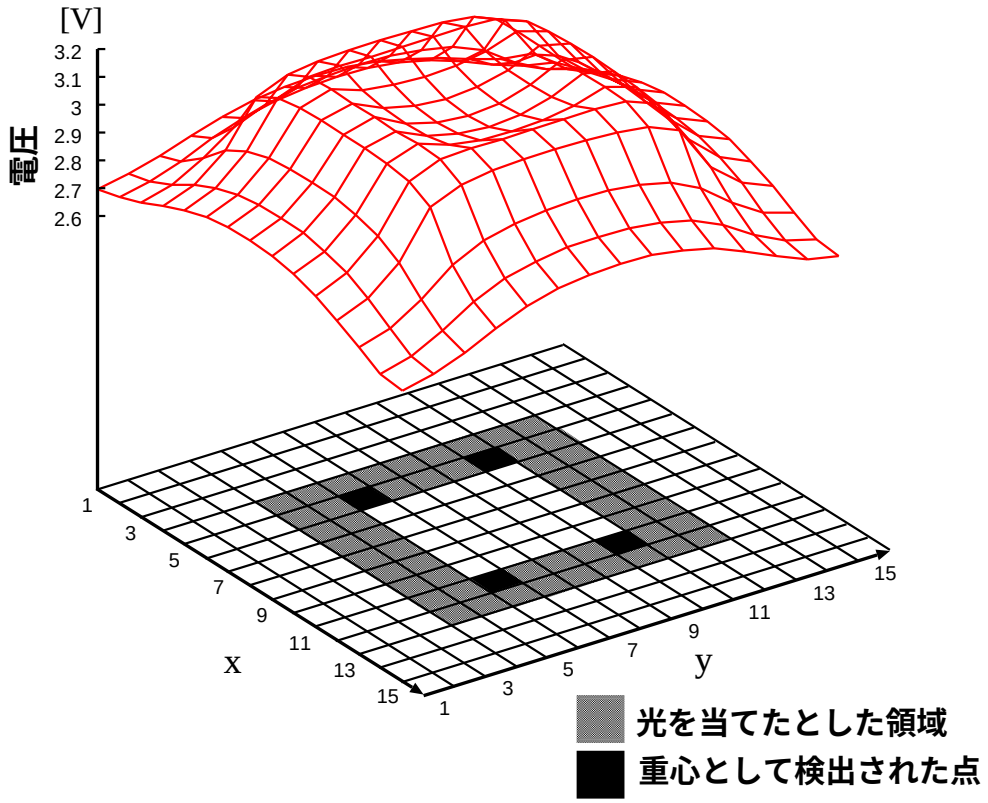

図 2.14: 電位の分布と重心検出 2

ドーナツ型の輪の重心は、この回路ではどのように解釈されるかというシミュレーション。結果は輪の中心ではなく、図のような場所に現れた。この回路の動作原理では『輪』の中心は検出できない。

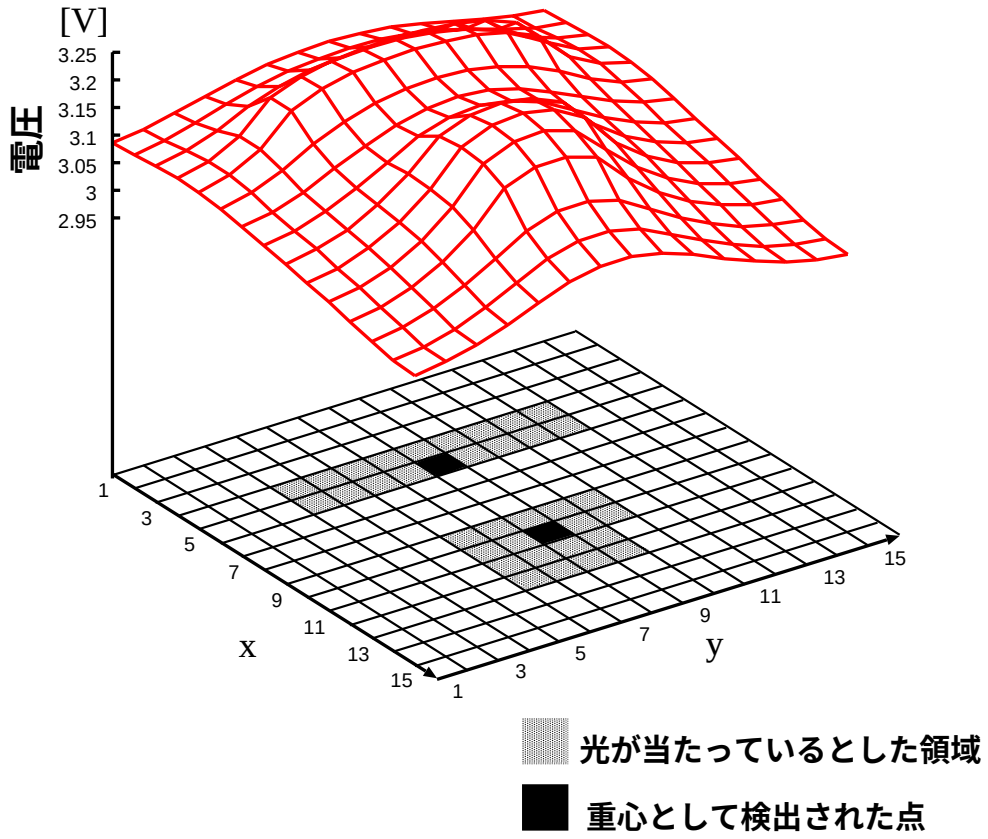

図 2.15: 電位の分布と重心検出 3

面積が同じで、形の違うものがどの程度の電位の差が生じるかのシミュレーション。結果は、棒状のものの重心点の電位が 3.2324[V] 正方形に近い形をした方が 3.2405[V] であった。次の章で述べる、出力回路と組み合わせたところこのふたつの点は同じステップで出力された。図 2.13 のシミュレーションでは面積の違う領域に対しては異なるステップで検出されたので、このシミュレーションから、同じステップでの検出はほぼ同じ面積と考えてもいいだろう。また、どの程度の電位がどの程度の面積に対応するかをあらかじめ調べることで、具体的な面積も分かるはずである。

第3章　座標情報の抽出

3.1　出力部

2章で述べた回路構成から、感知した領域の重心点を図3.1に示すように‘1’,‘0’の信号として平面上に出力することができた。

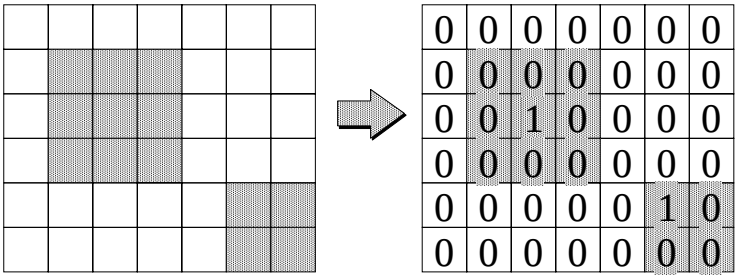

図 3.1: 重心検出の結果

しかし、これだけではただ‘1’,‘0’の配列が平面上にできただけで座標の情報が得られていない。したがって、回路の最終段で重心が検出されたことを示す‘1’を出力している画素の座標情報を得るためにエンコードが必要となる。これより以下にエンコードをどのように行ったかを説明する。

3.2　エンコードする際の問題点

これまで述べてきた重心検出回路の並列性を活かすために座標情報を得る際にも高速な処理が望まれる。まず、座標情報を高速に、できるならば重心が検出されたことを示す信号が出力されると同時に (ワンステップで) 得ることができないかと考え、画素毎にユニークな情報を持たせてそれを出力させるということを考えた。しかし、この場合画素が増えるに従い必要となる情報量が増し、そこで問題が発生することが分かった。例えば、画素毎にユニークな情報として電位を持たせるとして、利用できる電圧の範囲が 0[V]〜5[V] とした場合、100000 の画素を想定すると判別するべき電圧の値は $5\times\ 10^{-6}$[V] であるので、画素が増えるに従いこの判別が難しくなるのは容易に想像できる。さらに画素数が増え判別するべき電圧のレベルがもっと細かくなれば情報が雑音に埋もれる可能性も出てくるだろう。また、同時に複数個の出力を出そうとするならば画素数分だけ出力ピンが必要となる。これはたとえ 100 画素で考えても非現実的である。したがって、この手法を用いるのは難しいことが分かった。

そこで、このほかの手法としてエンコーダを用いることを考えた。エンコーダを使うのに、図 3.2 のように、各画素を各列毎に縦、横の OR をとるのものを考えた。しかし、こ

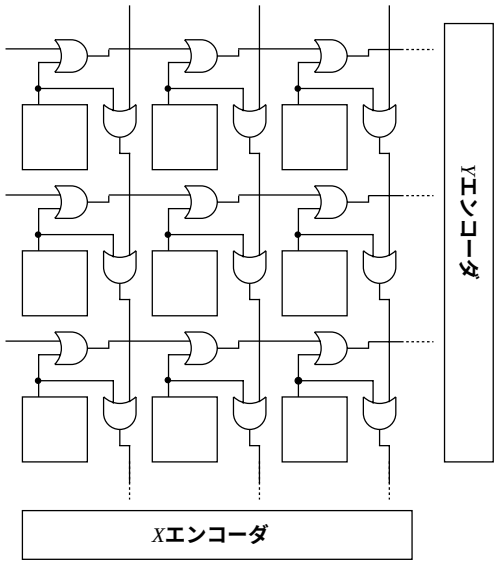

図 3.2: エンコーダを使うための配線

の構成だと複数個の出力があるとエンコーダに一度に複数個の入力が入ってしまう。エンコーダは入力が複数有る場合、一般には正しくエンコードできない。もし仮にできたとしても、各画素毎の個別の情報を載せることができないので例えば図 3.3 のように、

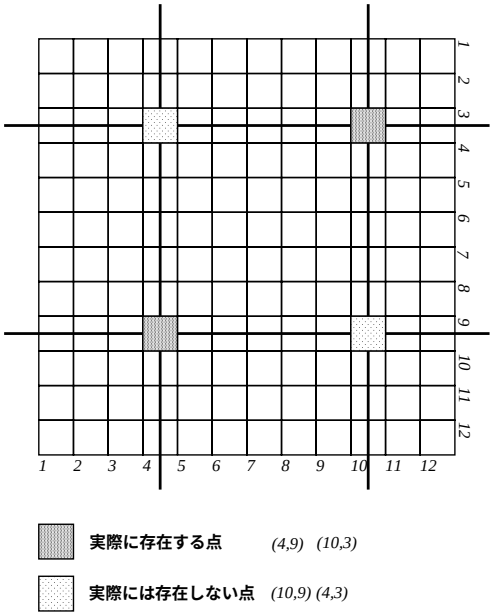

図 3.3: この配線の問題点

(4,9),(10,3) の 2 点の出力があった場合、影となる (10,9),(4,3) の部分も見えてしまい、2 点しかないはずがあたかも、4 点有るかのように見えてしまう。したがって、今回の回路のように同時に複数個の点が入力として考えられる場合には、この手法では正しい座標情報を得ることができない。以上、さまざまな手法を検討した結果、複数個の点から正しい座標情報を得るには、逐次処理を行うことが必要だろうという結論に至った。

しかし、全画素に対して逐次処理を行うと、ここで大きな時間が割かれてしまい 2 章で述べた回路の高速な並列処理が活かされなくなるので、全画素ではなく、処理をするべき画素のみを対象として、逐次処理を行うことを考えた。こうすれば、画素数が増えても全処理時間は検出された重心の個数と処理時間の積で済む。次の節でこの処理を行う回路構成と動作を説明する。

3.3 回路構成と動作

エンコーダは複数個の入力が同時にある時正しい出力が得られない。そこで、逐次処理を用いて順番に 1 つづつエンコーダに入力を与えることを考えた。この逐次処理は先ほども説明した通り、重心として検出された点のみが処理の対象となる。実際の回路構成は図 3.4 のようになる。

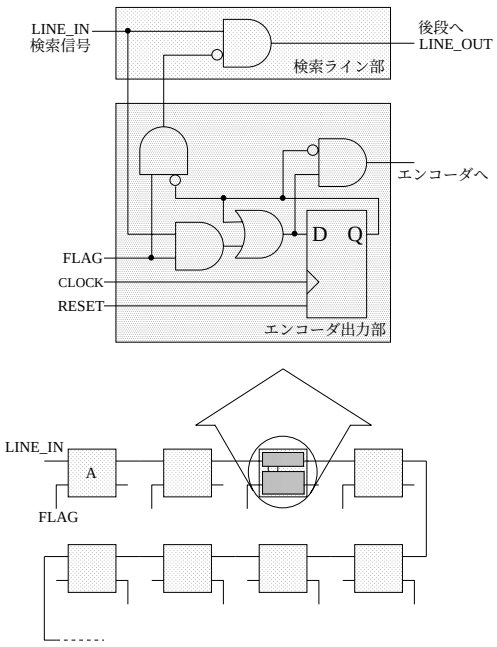

図 3.4: 出力回路の構成

まず全画素の出力回路を直列に接続し、全ての LINE_IN を ‘0’ にセットする。検索の開始は直列につないだ出力回路の一番端となる図 3.4 の A の LINE_IN に検索信号として ‘1’ を入力する。すると、図 3.5 にしめすように FLAG 端子が ‘0’ となっている場所では検索信号は素通りし、‘1’ となっている場所では 1 クロック分だけエンコーダ出力部へ寄り道をする構造となっている。ここでいう FLAG 信号は重心が検出されたことを示す信号で、極大の電位を持つ画素のロジック部の出力を指す。検索信号が寄り道をした画素では、エンコーダに出力が渡される。このようにすれば、検索信号 ‘1’ が届いていない画素はエンコーダからは見えない状態になり、FLAG 端子が ‘1’ の画素のみ順番にエンコーダへ出力を渡すことができる。また、一度エンコーダに出力を渡した画素は図 3.4 のような構造を取ることで一クロック分だけ出力を渡した後はエンコーダからは見えない状態になる。

全画素を検索信号が伝わるので、正確には全画素を逐次検査しているわけであるが、素通りする（AND のゲート遅延）時は 約 180[ps]/1 画素で通り過ぎるので 100×100 の画素が

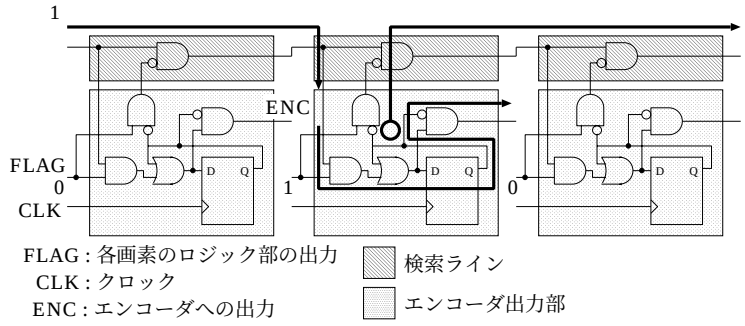

図 3.5: 出力回路の動作

あったとしてもこの素通りする時間はたかだか 10000×180^{-12}=1.8 [ns] であるから、50[μs] のノコギリ波の周期から見ると無視できるオーダであると考えて良い。また、寄り道をする時間、つまりクロックの周期はノコギリ波の周期より十分短くかつ、エンコーダが確実に動作する時間と座標情報出力回路が誤動作を起こさないようにするため 200[ns] とした。

3.4 座標情報出力回路の制御

この回路が正確な動作をするためには、座標情報出力回路が FLAG 信号の検索をしている時に、新たに FLAG 信号が立たない事が条件である。新たに FLAG 信号が立たないようにするには、2.6 節でも少し述べたようにノコギリ波の電位の減少を止めることで実現できる。したがってあとは検索信号を走らせるタイミングと FLAG 信号の立つタイミングの整合を取ればいい。FLAG 信号の立つタイミングはランダムでありこれを制御をするのはナンセンスであるので、検索信号を走らせるタイミングをよく考え整合をとることとした。

1. 検索信号を走らせるタイミング
   これは FLAG 信号が立った時である。これを調べるのに図 3.6 のような回路で FLAG 信号と出力回路の D-FF の Q の出力をまとめ、これを全画素 OR をとることにした。この信号を今後 OR_ALL 信号と呼ぶことにする。OR_ALL 信号はどこか 1 箇所でも FLAG 信号が立った時には '1' となり、全ての FLAG 信号が出力回路によって検索されると '0' になる。したがって、この信号に合わせて出力回路の検索信号を走らせてやればいい。

2. LINE_IN をリセットするタイミング
   次に全画素 FLAG 信号の検索と出力が終了した時点で、ノコギリ波を再び下げ始め次の重心点を検出しなくてはいけない。この準備としてまた、全画素の LINE_IN と

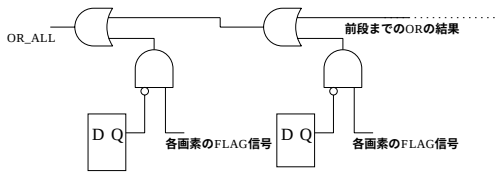

図 3.6: OR_ALL 信号を得るための回路

LINE_OUT を '0' に RESET しなくてはいけないのだが、このタイミングは直列につないだ LINE_OUT の一番最後の出力が '1' となった時である。この LINE_IN の一番最後の出力を今後 TAIL 信号と呼ぶ。TAIL 信号が '1' となったら、検索ラインを RESET する信号として検索信号の代わりに '0' を入力する。

3. ノコギリ波を再び下げ始めるタイミング
   最後に、次の極大点を検出するために再びノコギリ波を下げ始める。この確認は、直列につないだ LINE_IN の一番最後の出力が '0' であることを調べればよい。

1. の部分で OR_ALL 信号が立った直後に検索信号を走らせると回路が誤動作を起こすことが有る。これは、極大となっている電位が非常に近い場合にコンパレータが少し遅れて反応をはじめる時に起こる。図 3.7 これは、コンパレータのセトリングタイムが影響しているため、コンパレータの特性を変えてみても改善ができなかった。したがって FLAG 信号が立ってからしばらく待つ（1 クロック分）ことでこれを解決した。

以上の制御回路を Verilog-HDL でのトップダウンにより設計した。図 3.8 にこの回路の状態遷移を示す。正しい出力を得るために最低 3 クロック必要となった。また、ノコギリ波を発生させる回路はチップには組み込まないという前提で設計したので、この制御回路もチップには組み込まず外付けの回路とし、チップには組み込まないことにした。

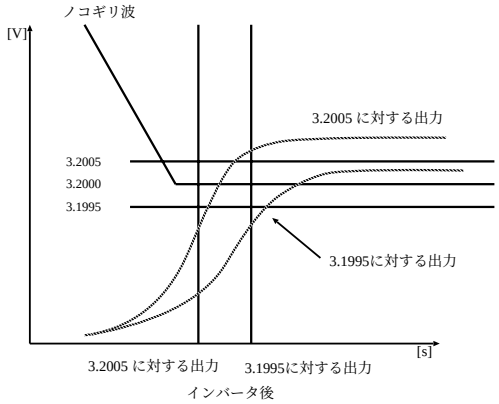

図 3.7: コンパレータのセトリングタイム

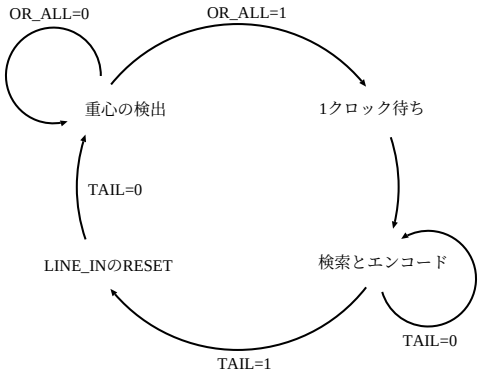

図 3.8: 制御回路の状態遷移図

3.5　出力回路の改良

全画素を直列につなぐことで逐次検索を実現したのであるが、素通りする時間が 180[ps] と大変短い時間ではあるが、やはり画素数に比例して増えていく。そこで、図 3.9 のような接続をしてこれを改善した。

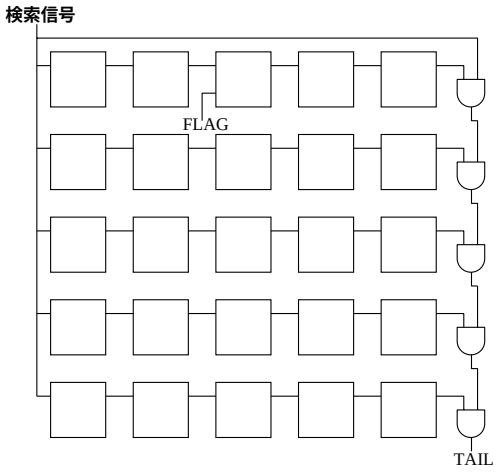

図 3.9: 出力回路の改善

このように配線することで、検索を並列に行うことができ FLAG 信号の立っている一列目の出力を行っている間に、他の 4 列の検索は終了している。したがって、直列につなぐよりも検索が早く終わり、画素数が増えても、全画素数ではなく、一辺の画素数に比例して処理時間が増えるので直列に接続したものに比べ、処理時間が短縮された。

3.6　回路全体のシミュレーション結果と動作の検証

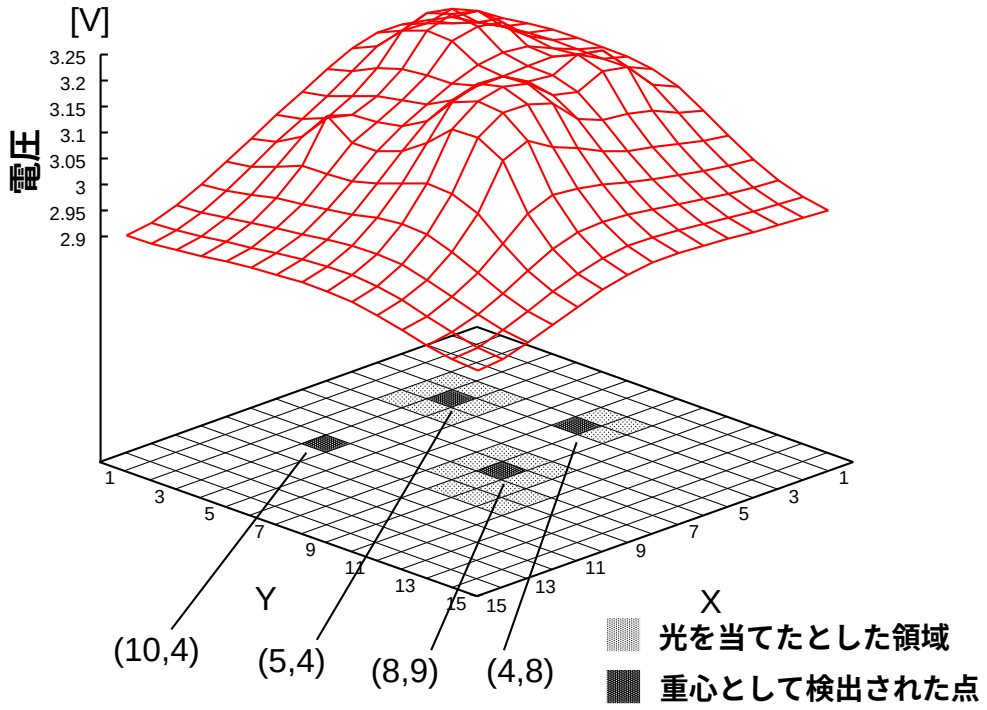

図 3.10: 回路全体のシミュレーション

2 章で述べた重心検出回路と 3 章で述べた出力回路をあわせ H-SPICE での回路シミュレーションを行った。その結果を図 3.10 に示す。重心の検出については正しく行われていることを 2 章の最後で述べたので、ここでは省略する。このシミュレーションにおける検出時間は、まず、重心検出にかかる時間、つまりノコギリ波の周期が 50[μs]。次に、出力回路による重心点の検索時間は、点 (5,4) と点 (8,9) における電位が非常に近い値となったので同ステップで検出されこのときにかかる時間が約 1.262[μs]。点 (4,8) と点 (10,4) は別ステップで検出されこのときかかる時間がそれぞれ約 0.928[μs] と約 0.958[μs] であった。したがって、エンコードにかかる時間を考えなければこの回路の図 3.10 のような入力に対する処理時間は 53.148[μs] であった。この結果から一つの点を検出するのにかかる時間が約 1[μs] であり n 個の検出を行う場合、この回路のおよその処理時間 t は

$$t = 50 + n \qquad [\mu s]$$

である。また、感知した領域の面積が非常に近い場合にはこれより速い検索時間が期待できる。これにより、現在ロボットビジョンの分野で必要とされる処理時間 1[ms] を十分に

上回り、高速に動作していることが確認できた。

3.7　チップ試作

以上の回路をノコギリ波発生回路と、それを制御する回路を除いた構成で、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 ROHM CMOS 0.6[μm]、3 層金属配線のプロセスで回路のレイアウトを行った。チップサイズは 4.5[mm] で、その 1 画素のレイアウトと全体のレイアウトを図 3.11 および図 3.12 に示す。画素数は 23×23 個、トランジスタ数は 69,575 個で、開口率は 5.7%である。また、受光部以外の拡散層で光電流が生じないように、受光部以外を 3 層目の金属でシールドを施している。来年度にこのレイアウトをもとにチップを試作する予定である。

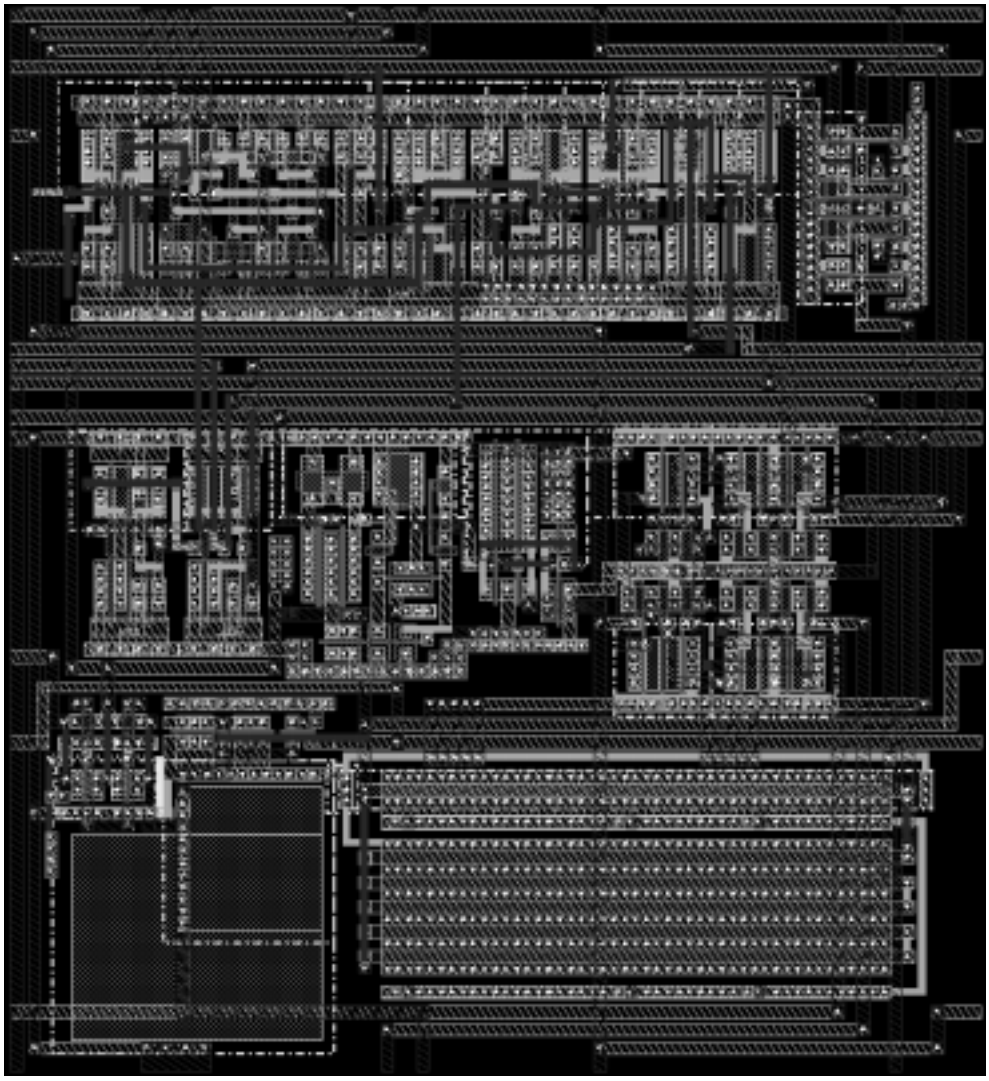

図 3.11: 1 画素分のレイアウト

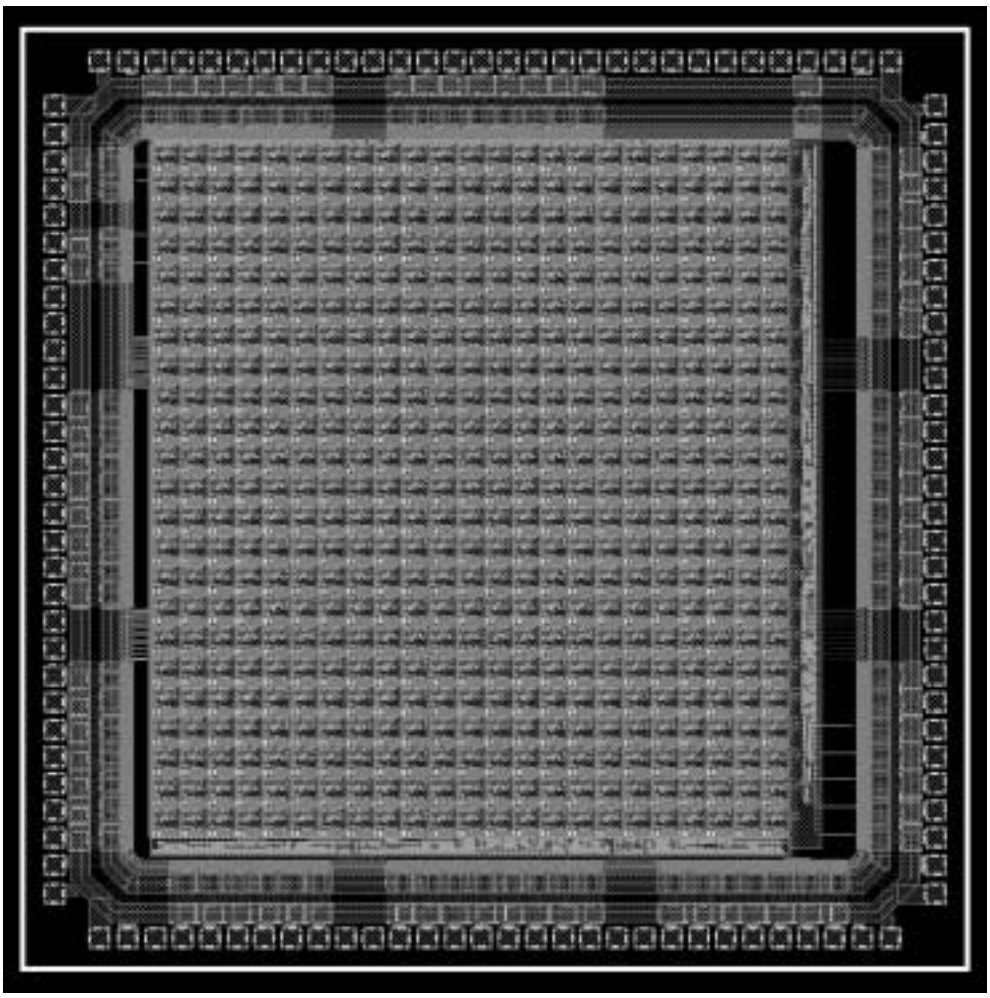

図 3.12: 全体のレイアウト（画素数 23x23 個）

第4章　まとめ

スマートセンサーという手法を用い、領域の重心検出と大きさ検出という部分に焦点を絞り、それを実現するアルゴリズムと回路構成の検討を行ってきたわけであるが、2 章、3 章で述べた回路構成により、回路に入力されたある領域の重心の座標情報と領域のだいたいの大きさが分かり、その処理時間はおよそ $50[\mu\mathrm{s}]$ と非常に高速である回路を作ることが出来た。この回路の応用として考えられるロボットビジョンの分野で現在必要とされる回路の性能は、画素数がおよそ 100×100 で、処理をするためにかかる時間が約 $1[\mathrm{ms}]$ である。今回提案した回路は検出する重心の個数にもよるがおよそ、ノコギリ波の周期である $50[\mu\mathrm{s}]$ で処理を行うことができ、これは必要とされる $1[\mathrm{ms}]$ と比べれば、十分に高速に処理が行えているといってよい。画素数については、現在の主流となっている $0.35[\mu\mathrm{m}]$ のプロセスで $9[\mathrm{mm}]$ ダイのチップを用いれば約 100×100 の画素数を実現できる。したがって、必要とされるスペックは持っているといえる。実際のロボットビジョンでの応用としては、検出した重心点の電位の値からおおよその領域の面積が分かることから、検出の対象となる物体の形がだいたい分かっていれば、対象の位置情報と大きさが分かるので、物体認識として応用の範囲は広いといえる。

関連図書

[1] A.Gruss *et al.*, "A VLSI Smart Sensor for Fast Range Imaging," *Proc.IEEE Int. Conf. on Intelligent Robots and Systems*,1992.

[2] C.Mead. "Analog VLSI and Neural Systems" 1989

[3] 藤井 信生 "集積回路時代のアナログ電子回路" 昭晃堂 1984

[4] 林 晴比古 "新 C 言語入門" ソフトバンク 1991

[5] 宮田 武雄 "速解 論理回路" コロナ社 1987

[6] 尾崎 弘 他 "電子回路 アナログ編" 共立出版社 1989

[7] C.Mead "Phototransduction by continuous-time,adaptive,logarithmic photoreceptor circuits" Rapport technique,California Institute of Technology,Computation and Neural Systems Program,CNS Memorandum30,Pasadena,CA 91125,1994

謝辞

本研究を行うにあたり、多くの方々に御助言、ご協力をいただきました。この場をお借りしまして感謝の意を表したいと思います。様々な面で御助言、ご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、生活面、その他様々な場面におきまして、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いただ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ご指導いただい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 の操作指導をしていただ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今年 1 年大変お世話になりました共同研究者の高瀬信二さん、渡辺晃さんに深く感謝いたします。1 年間ではありましたが、お世話をしていただきました M2 の小川明弘さん、中橋憲彦さん、早川史人さんに感謝いたします。研究室を盛り上げ、また、様々な助言をして下さった M1 の今井豊さん、数馬晋吾さん、藤田隼人さん、水野浩樹さんに感謝いたします。設計室の御掃除と整理整頓をして頂いた電磁波通信工学研究室 M1 の村上崇さんに感謝致します。同じ学年で一緒に研究を行い、楽しい生活空間を作っていただいた高松直樹君、中村公亮君、大門慎治君、佐々木勝光君、笠井稔彦君、蓮達弘君、水木誠君、辻川隆俊君に感謝いたします。大学での VLSI 設計を可能にしていただいた大規模集積システム設計教育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ご冥福をお祈りいたします。