

混合雑音が重畳した画像復元に関するハードウェア実装の研究

中村 幸弘[†] 橋本 健太郎[†] 木村 誠聡[†]

[†] 神奈川工科大学 情報学部 情報工学科 〒243-0209 神奈川県厚木市下荻野 1030

E-mail: {s055112, kimura}@cce.kanagawa-it.ac.jp

あらまし 撮像素子は画像を取得する際、レンズによるボケ、機器からの雑音の影響を受け、取得画像が劣化する。このため画像のボケと雑音の除去を行うためにフィルタによる処理が必要となる。また、これらの処理を実時間で行うためには、ハードウェアによる処理回路が必要となる。本稿ではそのうちのガウス性雑音とインパルス性雑音の混合雑音が重畳した画像における雑音除去のハードウェアの開発を行う。

開発するハードウェアはFPGAを用い画像の実時間処理を行うものであり、ファジー推論を用いた雑音除去フィルタと周辺処理の回路について言及するものである。本稿ではこれらのハードウェアを実装し、そのシミュレーション波形から仕様と合わせて問題が無いことを確認した。

キーワード 混合雑音, FPGA, ファジー推論

Mixed Noise Removal Hardware using FPGA

Yukihiro Nakamura[†] Kentaroh Hashimoto[†] Tomoaki Kimura[†]

[†] Department of Information Engineering, Kanagawa Institute of Technology

1030 Simoogino, Atsugi-shi, Kanagawa, 243-0292 Japan

E-mail: {s055112, kimura}@cce.kanagawa-it.ac.jp

Abstract An image is degraded by noise in the case of image acquisition. For this reason, filter process is needed in order to perform the remove of noise. And, real time processing of this process is needed hardware circuit. This paper states that hardware's development of image sensor's input and output We proposed image filter's real time processing hardware by using FPGA. The circuits of this filter are the noise removal circuit by using Fuzzy inference. In this paper, we explain this hardware by FPGA, and we verified from simulation waveform about validity of this embedded hardware.

Keyword Mixed Noise, FPGA, Fuzzy inference

1. まえがき

撮像素子は携帯電話やデジタルカメラなどで用いられており、その後の画像処理の部分は実時間での処理を行うことが要求されている。この撮像素子には特殊な用途を除けば CCD 撮像素子, CMOS 撮像素子の 2 種類が存在するが、画像を取得する際に暗電流や伝送経路の影響によりガウス性雑音やインパルス性雑音が重畳する問題がある [1,2]。また被写体を画像として取得する際に光学的に撮像素子の前にレンズが存在する。このレンズのために画像のボケによる劣化が存在する。すなわち、撮像素子によって取得した画像には、レンズによる画像の劣化、および、ガウス雑音やインパルス性雑音の重畳があるため、これらの問題を実時間にて処理する必要がある [2]。つまり、ボケの復元と雑音の除去を実時間で行える方法が望まれる。

画像のボケとガウス性雑音の除去の両方を行うため、一般的にはウィナーフィルタが用いられている [2]。このウィナーフィルタはレンズの点広がり関数 (PSF) と画像に重畳する雑音の状況によって最適なフィルタ係数を得るものの、除去できる雑音としてはガウス性雑音が対象となる。つまり、インパルス性雑音が重畳した画像から雑音を除去するためにはウィナーフィルタに代表される一般的な時不変線形フィルタでは不十分であることが知られている [2]。そこで、ウィナーフィルタの他にインパルス性雑音を除去するフィルタが必要となる。文献 [3] ではインパルス性雑音とガウス性雑音 (混合雑音) の両方の除去に対応するため、複数の情報をファジー推論によって関連付け、双方の雑音の除去を成し得ている。よって、文献 [3] のファジー推論を用いる方法とボケ復元の方法を組み合わせること

でレンズを通した画像の復元が可能と考えられる。そこで、本稿ではレンズによるボケの復元のためにウィナーフィルタとガウス性雑音とインパルス性雑音の除去が可能な文献[3]による方法の組み合わせを行う。しかしながら、これらの処理は処理点のみならず近傍領域の計算をも含むため、ソフトウェア処理では実時間の処理は困難である。そこで、画像のボケと混合雑音除去を実時間で行う方法が望まれる。

本稿では先に示した処理を実時間で行うためにウィナーフィルタとファジー推論を用いた雑音除去フィルタおよびその周辺の処理をハードウェア化することを提案し試みる。これらのハードウェアの規模は大きく、複数に分割して処理を進める必要がある。よってこの2つの処理のうち本稿において述べるのはファジー推論における処理と、その周辺の処理とする。つまり、本稿では「ファジー推論を用いた雑音除去フィルタ」と「周辺処理」に関してのみ述べるものとする。ファジー推論による処理では、文献[3]で示されている任意の2点間の画素の差分情報と距離情報の2つをファジー推論で関連付けた処理の実現を行う。そして、ファジー推論で用いるファジー集合をLUT(Look Up Table)化することで処理の高速化を試みる。また、各差分情報を求める回路では並列化させることで、ハードウェアの高速化を試み、提案する方法の実現化について具体的な回路の仕様を明らかにし、提案する方法の有効性を示す。

2. ファジー推論を用いた雑音除去法

文献[3]で示されている雑音除去フィルタは、任意の2点間の差分情報と、距離情報を関連付けることでガウス性雑音とインパルス性雑音で重畳された画像(混合雑音画像)の雑音除去を成し遂げている。差分情報と距離情報の2つの情報はファジー推論によって関連付けられており、差分が小さく、かつ処理点からの距離が短いときに処理点における平滑化処理に含まれるというものである。言い換えれば、インパルス性雑音が重畳している画素やエッジ等で変化が大きい画素は処理対象となる処理の平滑化に含まれれば、これらのことを式で示すと以下の式で表される。

$$y = \frac{\sum_k \sum_l W(\cdot) \cdot u(\cdot)}{\sum_k \sum_l u(\cdot)} \quad (1)$$

ここで、 $u(\cdot)$ は各ファジー集合の適合度であり、 $W(\cdot)$ はフィルタ荷重である。さらに $W(\cdot)$ は、各情報量 $E(\cdot)$ 、 $e(\cdot)$ 、 $d(\cdot)$ によって算出される。 $E(\cdot)$ は処理点と周辺画素との差分情報であり、次式によって求められる。

$$E(k,l) = \frac{1}{P^2} \left\{ \sum_{m=-Q}^Q \sum_{n=-Q}^Q e(k,l,m,n) \right\} \quad (2)$$

$e(\cdot)$ は、ある点と点との差分情報であり、次式によって求められる。

$$e(k,l,m,n) = |x(i+k, j+l) - x(i+m, j+n)| \quad (3)$$

$d(\cdot)$ は、距離情報であり、次式によって求められる。

$$d(\kappa, \lambda) = \sqrt{\kappa^2 + \lambda^2} \quad (4)$$

以上の情報を用いてこれらを以下のルールによって関連付ける。

if	E(0,0)	is	S
	e(k,l,0,0)	is	{S,M,L}
	d(?,?)	is	{N,NF,F}
then	w(k,l)	is	W(i=1,...,9)
if	E(0,0)	is	L
	E(k,l)	is	{S,M,L}
	d(?,?)	is	{N,NF,F}
then	w(k,l)	is	W(i=10,...,18)

式(5)の上のifの項は、処理点 $E(0,0)$ にインパルス性雑音が重畳していない時のルールである。この時は処理点を中心とした近傍領域内の画素と処理点との差分値を求める。もし、差分値が大きい時には、その近傍領域の画素は平滑化に用いないことを表している。式(5)の下の方のifの項は、処理点 $E(0,0)$ にインパルス性雑音が重畳していた場合のルールである。この時は近傍領域内の各画素同士の差分値を求め、その差分絶対和が一番小さい画素がこの近傍領域内のメジアン値である[3]。よって、この任意の2点間の差分値を求めることで対象となる画素の近傍領域内におけるインパルス性雑音の影響等を取り除くことが出来る。

各式により求められた情報量に対するファジー集合を図.1~3に示す。各情報量に対応する適合度 μ [$E(0,0)$]、 μ [$E(k,l)$]、 μ [$e(k,l,0,0)$]、 μ [$d(\lambda, \kappa)$]はこのファジー集合によって求められる。

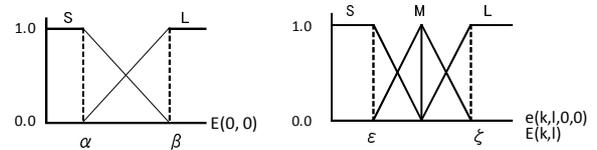


図.1 $E(0,0)$ のファジー集合

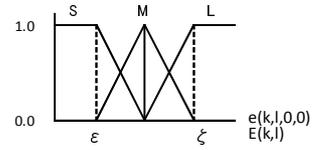


図.2 $e(k,l,0,0)$, (k,l) のファジー集合

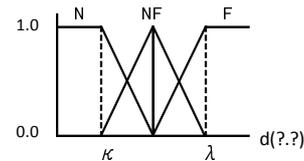


図.3 $d(\lambda, \kappa)$ のファジー集合

求められた適合度を用いて、各ルールに対する適合度を次式で表すmin演算で算出する。

$$\begin{aligned}\mu_1 &= \min\{\mu_S[E(0,0)], \mu_S[e()], \mu_N[d()]\} \\ \mu_2 &= \min\{\mu_S[E(0,0)], \mu_S[e()], \mu_{NF}[d()]\} \\ \mu_3 &= \min\{\mu_S[E(0,0)], \mu_S[e()], \mu_F[d()]\} \quad (6)\end{aligned}$$

...

$$\mu_{18} = \min\{\mu_L[E(0,0)], \mu_L[e()], \mu_F[d()]\}$$

各ルールに対する適合度とそのルールに対する後見部実数値を用い、次式で示す非ファジー化手続きにより式(1)におけるフィルタ係数 $W(k,l)$ が算出されることになる。

$$W(k,l) = \frac{\sum_{q=1}^n u_q \cdot w_q}{\sum_{q=1}^n u_q} \quad (7)$$

なお、表 1 に後件部実数値を示す。

表 1. ファジールールのルールテーブル

d(k,l)	E(0,0)					
	Small			Large		
	e(k,l,0,0)			E(k,l)		
	S	M	L	S	M	L
N	w1 1.00	w4 0.50	w7 0.25	w10 1.00	w13 0.50	w16 0.25
NF	w2 0.75	w5 0.50	w8 0.00	w11 0.75	w14 0.50	w17 0.00
F	w3 0.50	w6 0.25	w9 0.00	w12 0.50	w15 0.25	w18 0.00

文献[3]では以上の様に 2 つの情報を関連付けることで混合雑音の除去を成し得ている。

なお、文献[3]ではファジー集合や閾値のパラメータは定まっているが、本稿ではこれを特に定めず、ハードウェアとして可変可能としておくことで柔軟な対応がとれる様にする。

3. 実装するハードウェアの構成

ここでは、2 章で説明したファジー推論による雑音除去フィルタを実時間処理する為にハードウェア化することを試みる。ハードウェア化する為にハードウェアの内容がプログラマブルに書き換え可能な LSI である FPGA(Field Programmable Gate Array)を用いる。この時演算回路を効率化するために積和演算機能のハードウェアブロックが搭載された Xilinx 社製「Spartan3-3400-DSP」が実装された東京エレクトロニクス社製のボード「TB-3S-3400DSP-IMG」を使用する(図.4)。なお用いる言語は VHDL とする。

このボードにカメラ入力のインターフェースとディスプレイ出力の DVI インターフェースを付属するこ

とで、カメラ入力された画像を処理しディスプレイに表示するハードウェアが実現可能となる。

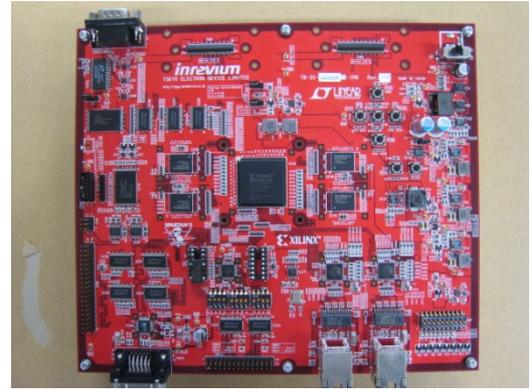


図.4 評価ボード

3.1. ファジー推論による雑音除去フィルタの実装

ここでは、ファジー推論における具体的な回路について述べる。図.5 はファジー推論部の全体の構成である。図.5 の構成はさらに差分情報を求めるブロック、各ファジー集合の適合度を求めるブロック、各適合度の MIN 演算のブロックらで構成される。

以下は各々のブロックについて説明を述べる。

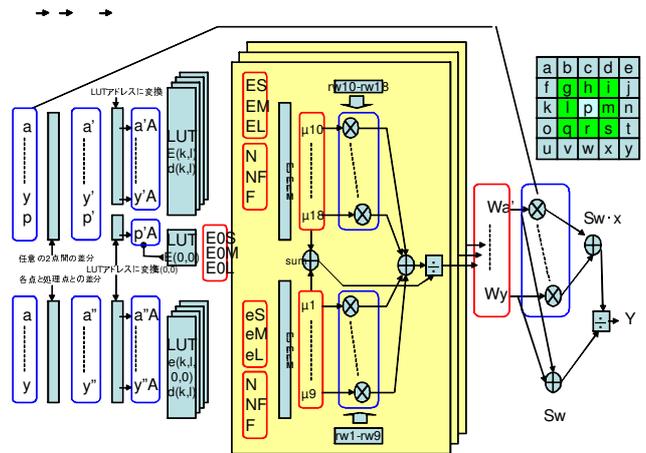


図.5 Fuzzy Filter の構成

3.1.1. 差分情報を求める回路

この回路は図.6 に示すように処理点と周辺画素との差分情報を求める部分と図.7 に示すように任意の 2 点間の差分情報を求める部分とに分かれる。前者は 2 章で述べたように処理点にインパルス性雑音の影響が無い場合であり、後者は処理点がインパルス性雑音である可能性が良い場合である。図.8, 図.9 は、

その具体的な回路である。図.6 は処理点とその近傍領域との差分情報であり、回路としては単なる絶対差分となっている。

図.7 は任意の 2 点間の差分情報であり、ある点と処理点近傍の 9 点の信号との絶対差分和である。この回

路も単純ではあるが、その回路規模は図.6 に示すものより大きくなる。

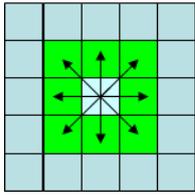


図 6. 処理点と周辺画素との差分情報

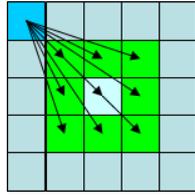


図 7. 任意の二点間の差分情報

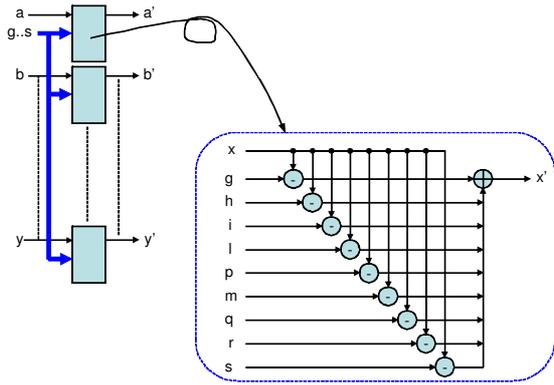


図 8. 処理点と周辺画素との差分値を求める具体的回路例

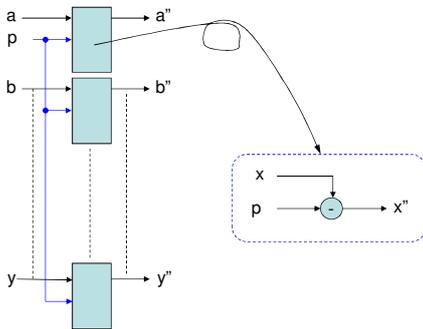


図 9. 任意の二点間の差分値を求める具体的回路例

3.1.2. 各ファジー集合の適合度を求める回路

適合度の値を順序回路にて構成すると非常に多くの遅延が発生すると考えられるため、各ファジー集合の適合度を、各差分情報を元に LUT(Look Up Table)方式を用いて求める方式とする。LUT は基本的にメモリであり、アドレスに対応したデータを出力する。LUT を用いる理由として、非常に高速で適合度を求めることが要求されるためである。つまり LUT の場合、メモリアクセスの遅延時間のみとなるため、高速になることは理解できよう。

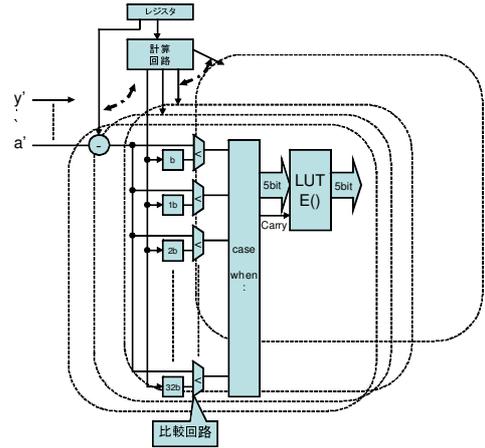


図 10. E() の LUT 回路

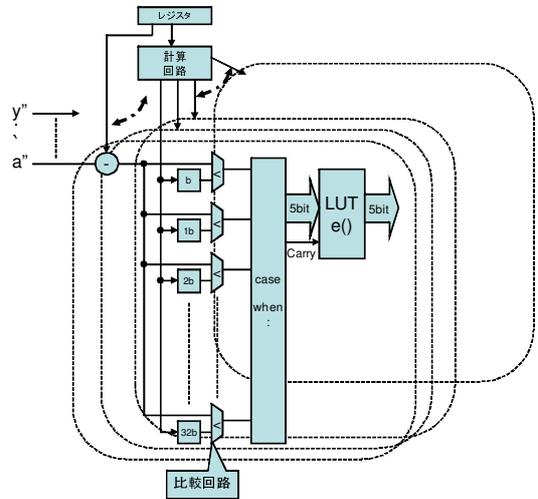


図 11. e() の LUT 回路

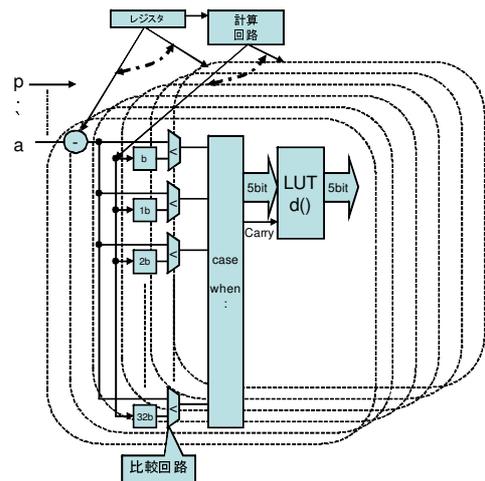


図 12. d() の LUT 回路

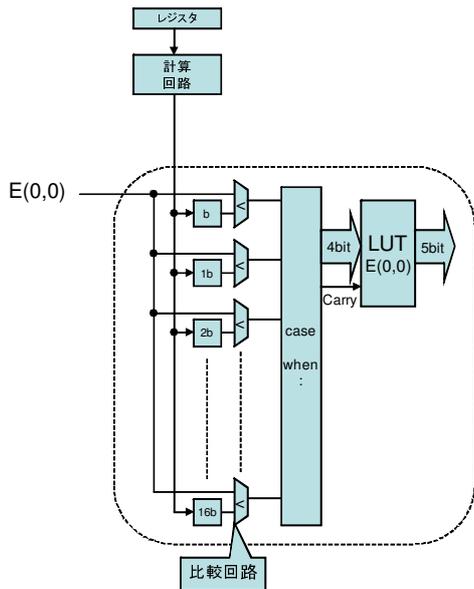


図 13. E(0,0)の LUT 回路

図.10~13 はその模式図である. LUT のアドレス部に差分情報, 又は距離情報を入力することで, その情報に対応した適合度を出力する. 具体的には各情報量 $E(k,l)$, $e(k,l,0,0)$, $d(k,\lambda)$ を各 LUT にそれぞれ入れることで適合度が求まる. 図.10~13 の各 LUT は, ファジー集合の各メンバシップ関数である. しかしながら, この LUT にメンバシップ関数の細かいデータを入力することはメモリの増大をまねく. よって, メモリの消費を抑えるためにメンバシップ関数を段階的に表現することを考える. 図.14~16 は段階的に表現したメンバシップ関数である. この精度は計算上 10%以内の誤差に抑えることができることが判明しており, シミュレーション上計算した適合度とほとんど変わりが無いことが分かっている.

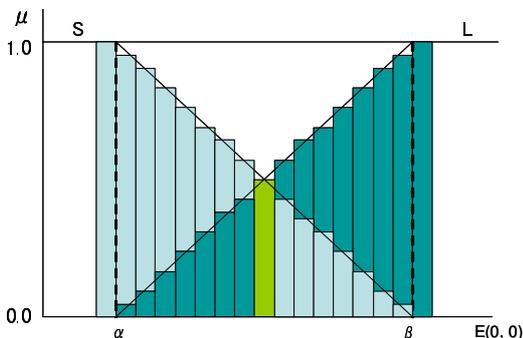


図.14 LUT 化したメンバシップ関数 E()

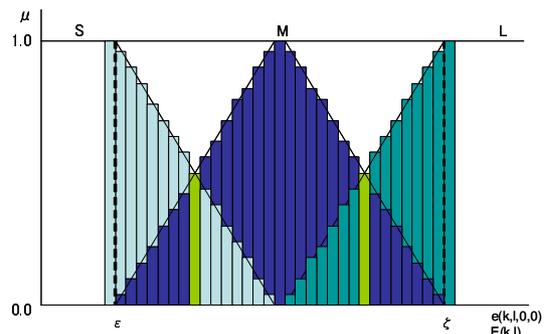


図.15 LUT 化したメンバシップ関数 e()

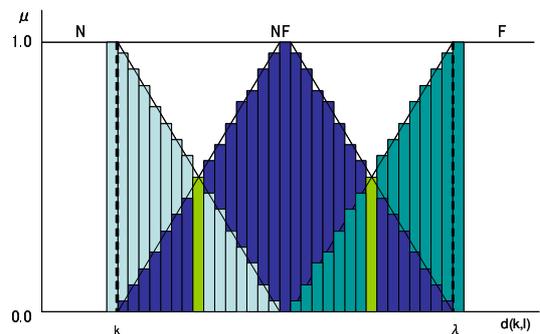


図.16 LUT 化したメンバシップ関数 d()

このメンバシップ関数を段階的に表現したのちに LUT 化を行う. 具体的に各メンバシップ関数を LUT 化したものの例として E() を図.17 に示す.

E(0,0)	S	L
α 以下	1.0	0.0
$\alpha + a$	0.9375	0.0625
$\alpha + 2a$	0.875	0.125
$\alpha + 3a$	0.8125	0.1875
$\alpha + 4a$	0.75	0.25
$\alpha + 5a$	0.6875	0.3125
$\alpha + 6a$	0.625	0.375
$\alpha + 7a$	0.5625	0.4375
$\alpha + 8a$	0.5	0.5
$\alpha + 9a$	0.4375	0.5625
$\alpha + 10a$	0.375	0.625
$\alpha + 11a$	0.3125	0.6875
$\alpha + 12a$	0.25	0.75
$\alpha + 13a$	0.1875	0.8125
$\alpha + 14a$	0.125	0.875
$\alpha + 15a$	0.0625	0.9375
$\alpha + 16a$ 以上	0.0	1.0

図.17 数値化したメンバシップ関数 E()

3.1.3. MIN 演算回路

各 LUT により求められた情報量を用いて, それぞれの MIN 回路に流して適合度を求める. 図.21 に MIN 演算回路のブロック図を示す. この回路は E(0,0), e(), d() から出力された適合度の中で一番小さい値を選択するものである. この一番小さい値を選択する回路の具体的な回路を図.22 に示す.

最初に $e()$ と $d()$ を比較し、小さい方の値を次の回路へ伝える。比較として補数を用いた加算回路、つまり $e() - d()$ の引き算回路を用いる。図.22 において Carry が立った時には、 $e()$ が $d()$ より小さいことを示し、その際には $e()$ のデータを次段に伝える。

次に、 $e()$ 又は、 $d()$ のデータと $E(0,0)$ を同様に比較する。その結果出力には一番小さいデータが出力されることになる。最終的に一つの画素の差分情報に対して 9 つの比較回路が必要となる。

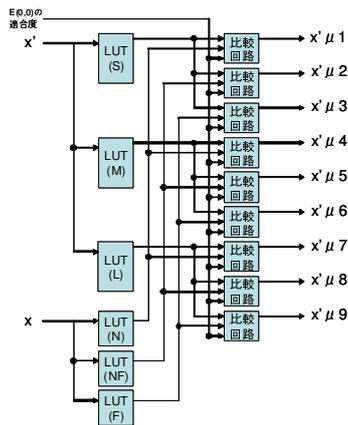


図.21 MIN 演算回路のブロック図

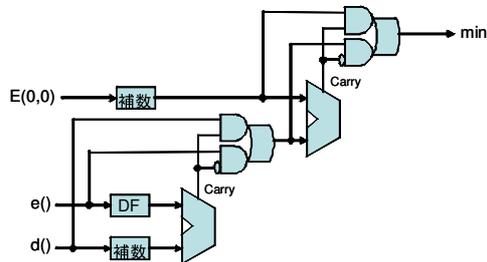


図.22 MIN 演算回路の具体例

3.1.4. ファジーフィルタの出力

3.3.1~3.3.3 までに述べたように各々のブロックにてファジー推論に必要なデータがそろうことになる。最終的には図.5 の最終段にあるように、表 1 に示す後見部実数値を各々の適合度と乗算し、その総和をとる。そして $W(k,l)$ にて正規化を行うことで処理における雑音除去の結果が得られることになる。

4. シミュレーション結果

表 2 は、Fuzzy Filter の差分を求める回路の論理合成結果であり、ボードに搭載されている Xilinx 社 FPGA 「Spartan3A-DSP」の全体中の消費%を示している。

表 2. 差分回路の論理合成結果

Slice数	4932	20%
FF数	1808	3%

図.20 に Fuzzy Filter の差分情報を求める回路の RTL 生成結果の一部抜粋したものを、図.21 にシミュレーション波形の結果を載せる。RTL 生成結果とシミュレーション結果により、差分情報を求める回路の確認ができた。

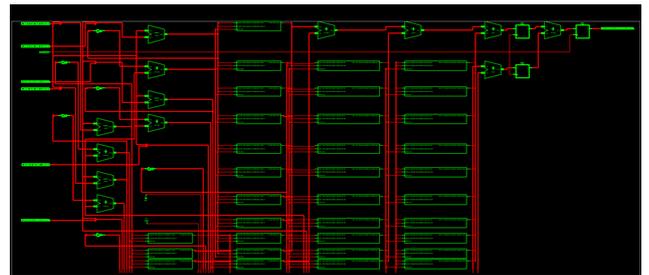


図.23 差分情報を求める回路の RTL 生成結果(抜粋)



図.24 差分情報を求める回路のシミュレーション結果

5. まとめ

本稿では、ボケと混合雑音で劣化画像の実時間による復元処理のハードウェアについて提案を行い、その仕様を明らかにした。提案する方法では種々の周辺回路とファジーフィルタの高速化について提案し、その実現方法について言及した。

参考文献

- [1]R.C.Gonzales, R.E.Woods, “Digital Image Processing”, Addison-Wesley Publishing, Massachusetts, 1993
- [2]A.Rosenfeld, “Digital Picture Processing”, Academic Press, Inc, 1976
- [3]木村誠聡, 田口亮, 濱田敬, 村田裕, “混合雑音重畳画像復元のためのファジーフィルタの提案”, 電子情報通信学会技術報告, DSP98-28, May., 1998.