

データ点数・語長を変えたWSI FFTプロセッサの冗長化設計

Redundancy Design of WSI FFT Processors Varying Number of Data Points and Word Length

○金沢正治, 苫米地宣裕

○Shyouji Kanazawa, Nobuhiro Tomabechi

八戸工業大学

Hachinohe Institute of Technology

キーワード: 冗長化 (redundancy), WSI, FFT, 高速 (high speed),
システム構成 (system design)

連絡先: 〒031 青森県八戸市大字妙字大開88-1 八戸工業大学 電気工学科
苫米地研究室 金沢正治, Tel. (0178)25-3111, Fax. (0178)24-1430

1. はじめに

現在、画像信号のデジタル化と共に、画像信号をリアルタイムで処理可能なFFTプロセッサの実現が期待されている。そのような高速FFTプロセッサは、全ての演算をハードウェアで構成する必要があり、WSI規模の集積度を必要とする。本研究者らは、このWSI規模のFFTを、冗長化手法(欠陥救済法)¹⁾の導入によって、歩留りの低下なしに構成する方法を検討してきた。本稿では、データ点数と、演算精度を表すbit数(語長)を変化させた、9種類のFFTの冗長化について報告している。特に、冗長化設計において、冗長な配線と切り替えスイッチを含めた、冗長なチップ面積の合計が最小となるように条件を求めている。

2. 非冗長FFTプロセッサの設計

2.1 FFTの仕様の設定

FFTの仕様は表1の様に設定する。

表-1 FFTプロセッサの仕様

次元	: 1次元
サンプリング周波数 f_s	: 10MHz
データ点数	N: 512~2048点
リアルタイム性	: リアルタイム動作
語長	D: 12~20bit 固定小数点
基数	r: 2
間引き方式	: 時間間引き方式
素子	: CMOS素子

ここで、データ点数は512点、1024点、2048点、語長は12bit、16bit、20bitとし、計9種類のFFTについて冗

長化設計を行う。表1の仕様を満足するには全てのバタフライをハードウェアで構成する必要がある。バタフライ演算は複素入力A、B及び回転子 W^k (ここでWはデータ点数Nに関する複素指数関数であり、kは整数である)に対して $A' = A + W^k B$ と $B' = A - W^k B$ の出力を得るものであり、この演算の実現には加算器を6個、乗算器を4個必要とする。データ点数 $N = 2^m$ 点のFFTは、 $m \times N/2$ 個のバタフライを必要とする。仮に、 $N = 1024$ 点では、段数が10段のバタフライ列(これをパイプラインの列と定義する)が512列必要となる。従って、システムはバタフライを図1の様に配列したものになる。

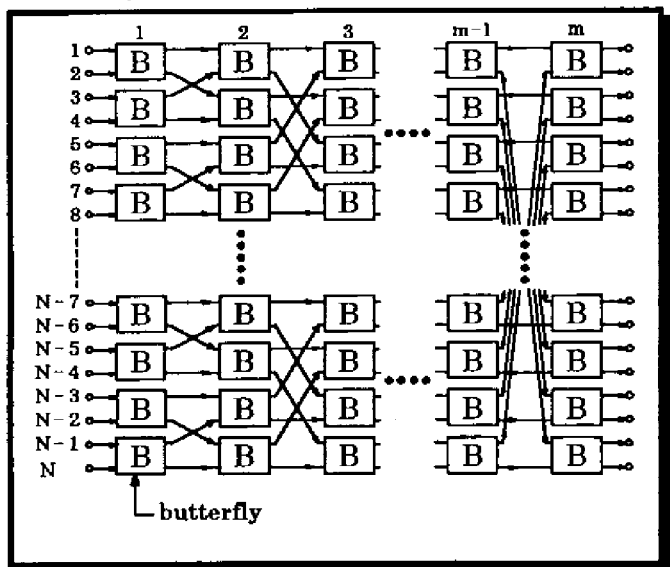


図-1 FFTの信号流れ図

Fig.1 Signal flow graph of FFT

2.2 FFTプロセッサの問題と対策

全ての演算をハードウェアで実現するFFTプロセッサは次のような問題がある。

1. バタフライ間の相互配線のチップ面積が、バタフライ自身のチップ面積に比較して無視できない大きさとなる。
2. バタフライの配列が、縦方向と横方向で数が極端に異なり、非対象構造となる。仮に、データ点数が1024点のFFTでは、縦は512個に対して横が10個

という構成になり、もしバタフライ1個のレイアウトが正方形ならば、システム全体は50対1という極端な長方形となり、制作が困難となる。

“1”の問題の対策として、本稿ではバタフライ間の相互配線における信号伝送を8bit直列伝送とすることになっている。信号の直並列変換は、バタフライの入出力段に送信・受信のバッファレジスタを取り付けて行う。これはFFTをパイプラインで処理する際に必要なレジスタの機能と併用できるため、この回路の付加によるチップ面積の増加は少ない。バッファレジスタの動作クロックはFFTの動作クロックの8倍の80MHzとなり、処理時間はFFTのパイプライン処理にマッチしており、信号の直列伝送によるFFT処理時間の増加は無視できる。

“2”の問題の対策としては、バタフライのレイアウトを出来るだけ横長に設計することで解決している。この方法は、同時にバタフライ間の相互配線のチップ面積を少なくすることにも寄与している。

2.3 非冗長FFTプロセッサの設計

非冗長なFFTプロセッサの論理設計、回路設計、および、レイアウト設計を行い、各回路の構成とチップ面積を明らかにしている。

FFTは、バタフライの段ごとにパイプライン方式で処理を行う。これにより、システムの動作速度はバタフライ1個の動作速度と等しくなる。バタフライの構成は図2の様になる。

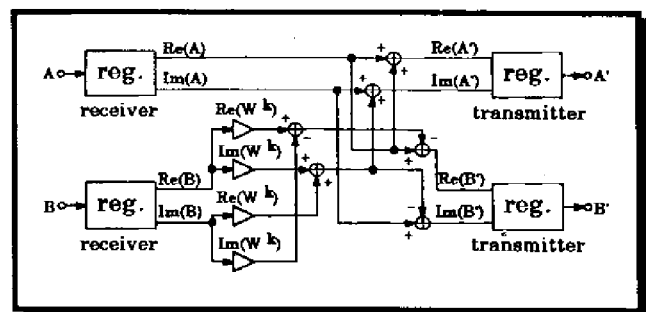


図-2 バタフライの構成

Fig.2 Structure of a butterfly

図示したように、バタフライは、送信シフトレジスタ2個、受信シフトレジスタ2個、加算器6個、および乗算器4個により構成されている。加算器は4bitを単位とするキャリー先見方式、乗算器はWallace Tree方式を採用している。4bitキャリー先見加算器からバタフライまでのレイアウト設計結果を図3に示す。

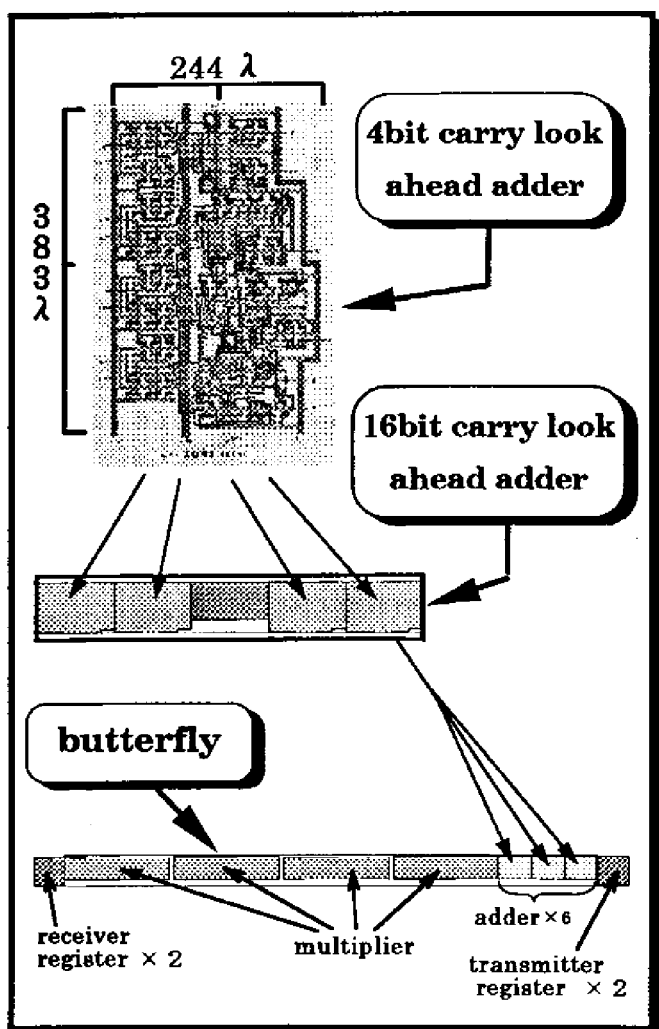


図-3 バタフライのレイアウト
Fig. 3 Layout of a butterfly

図3において、素子はCMOSを用いており、レイアウト設計規則は、Mead-Conway²⁾の規則をCMOSに拡張して³⁾用いている。また、λは基準寸法単位を表している。レイアウト設計結果をもとに割り出された、FFTの各構成要素のチップ面積を表2に示す。表2には記されていないが、この時のバタフライの縦横比は、語長が12bit時は、約1:23、

16bitでは約1:25、さらに20bitでは約1:26となり、どれも横長に設計されている。

表-2 各回路のチップ面積
(単位: λ²)

回路 \ 語長	12	16	20
受信シフトレジスタ	8.18×10^5	1.36×10^6	2.07×10^6
送信シフトレジスタ	8.18×10^5	1.36×10^6	2.07×10^6
加算器	9.81×10^5	1.53×10^6	2.23×10^6
乗算器	6.60×10^6	1.17×10^7	1.80×10^7
バタフライ	3.74×10^7	6.44×10^7	9.83×10^7

表2の結果を基にデータ点数、語長を変えた9種類の非冗長FFTのチップ面積を求めたものを表3に示す。

表-3 非冗長FFTのチップ面積
(単位: λ²)

データ点数 \ 語長	512	1024	2048
12	9.02×10^{10}	2.08×10^{11}	4.86×10^{11}
16	1.55×10^{11}	3.56×10^{11}	8.32×10^{11}
20	2.37×10^{11}	5.44×10^{11}	1.27×10^{12}

表3に記されている非冗長FFTの縦横比は、どの場合でも1:1~1:3の範囲内であり、非対象構造が解消されている。

2. 4 冗長FFT設計の指標

FFTの歩留りを計算するにあたり、基準となるチップを設定する。

ある時点の技術水準で、歩留りがある基準値:Y₀となるチップ面積を基準チップ面積と呼び、A₀で表す。本稿ではY₀=0.5ととる。基準チップ面積を有するLSIを基準LSIと呼ぶ。現在の技術水準では、基準LSIはダイナミックRAMで8Mbit程度であると考えられる。そのチップ面積を前述の設計ルールを

用いて求めると、約 $3 \times 10^9 \lambda^2$ となる。よって本稿では $A_0 = 3 \times 10^9 \lambda^2$ と設定する。ある対象となるシステムの歩留りが Y_0 に等しいという条件下で、システムのチップ面積が基準 L S I の何倍となるかという倍率 F は“チップ拡大率”⁴⁾ と呼ばれ、冗長化設計の指標となる。 $F \geq 100$ が W S I 規模を表すと考えられる。本稿で考察する F F T プロセッサのチップ面積は、表 3 よりデータ点数 5 1 2 点の 3 種類の F F T 及びデータ点数 1 0 2 4 点、語長 1 2 b i t の F F T 以外はすべて $F \geq 100$ となり、これらの F F T は W S I 規模になることになる。

3. F F T プロセッサの冗長化設計

3.1 冗長化 F F T プロセッサの構成法

本稿で提案するシステムの冗長化の方法を図 4 に示している。

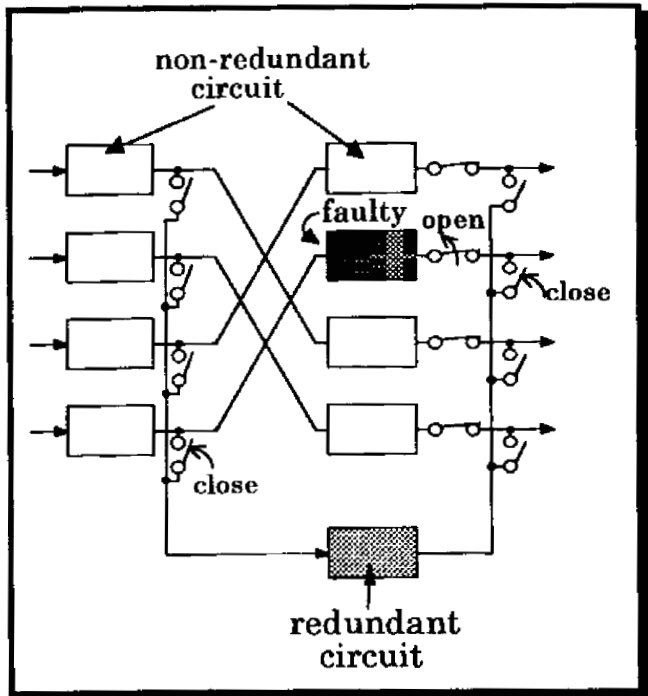


図-4 冗長化システムの構成法

Fig.4 Structure of the redundant system

まず、システムを等しい機能を有する基本回路に分割し、さらに基本回路と等しい機能を有する予備の回路（冗長な回路）を用意する。不良となった基本回路がある場合は、冗長な回路

と置き換える。回路の置き換えには基本回路の入出力信号線をスイッチで切り替えて行う。その際、バタフライ間の相互配線の不良に対処するため、相互配線はその配線を入力とする基本回路に含めて取り扱う。このため入力信号線の切り替えスイッチは、前段の基本回路の出力点の直後に設置する。冗長な回路の乗算器の係数及び切り替えスイッチの制御は、レーザーによってヒューズを切断する方法^{5) 6)}を用いる。入力信号線の切り替えスイッチはヒューズ⁶⁾とパストランジスタで構成される。出力信号線の切り替えスイッチは2個を1組として、ヒューズ1個、パストランジスタ2個、及びインバータ1個で構成される。

冗長化を行った後の F F T プロセッサの全体構成を、基本回路を単位とする一般的な形で表したものを図 5 に示している。

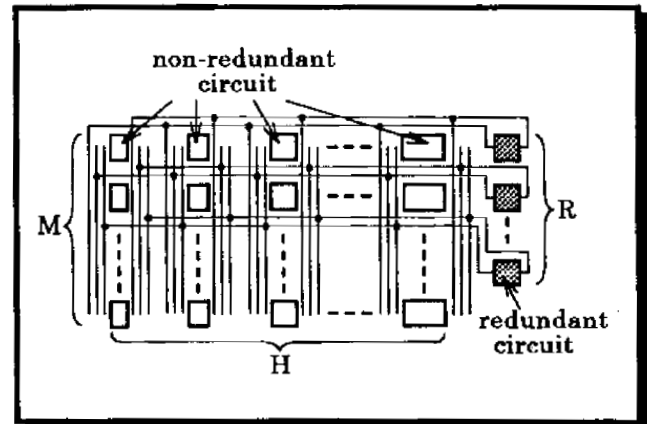


図-5 冗長化 F F T プロセッサの全体構成
Fig.5 Overall structure of the redundant FFT processor

図において冗長な回路は、各々がシステム内の任意の回路と置き換えられるよう結線されている。また、基本回路は相互配線を含むので、その寸法は横方向に行く程大きくなる。

F F T プロセッサの基本回路としてはバタフライを設定するのが最も冗長なチップ面積が小さく、効率が良くなることをすでに確認している。従って、本稿では基本回路としてバタフライを用いることにしている。

3. 2 冗長化FFTの歩留り計算式

本稿では、次の仮定と条件のもとに歩留り計算を行っている。

仮定1：欠陥分布はランダム⁷⁾とする。

仮定2：電源配線、クロック配線の切断やショートのような広範囲の障害を生ずる欠陥は生じないとする。

条件1：冗長な配線と切り替えスイッチは、冗長な回路に所属するとして取り扱う。

仮定1の下での非冗長なLSIチップの歩留りYは次のように与えられる⁸⁾。

$$Y = e^{-D_0 A} \quad (1)$$

ただし、 D_0 は平均欠陥密度を、 A はチップ面積を表している。まず、 $\text{Pr}[\]$ は確率を、 H は基本回路の横方向の数、 M は基本回路の縦方向の数、 R は冗長な基本回路の数、 A_i は横方向 i 番目の基本回路のチップ面積、 A_p は冗長な基本回路のチップ面積、 ${}_M C_n$ は M 個の中から n 個を選択する組み合わせの数を表すことにする。このとき、冗長化を行ったFFTプロセッサの歩留り Y_R は、次のようになる⁹⁾。ただし、

$\sum_{n_1+n_2+\dots+n_H+n_p \leq R}$ は、 $n_1+n_2+\dots+n_H+n_p \leq R$ となる、

$n_1, n_2, \dots, n_H, n_p$ の全ての組み合わせについて加算を行うことを表している。さらに、 $\min(M, n)$ は変数 M, n の内で小さい方の値を出力する演算を表している。

$$Y_R = \sum_{n_1+n_2+\dots+n_H+n_p \leq R} \text{Pr}[$$

1 番目の行の n_1 個の回路が不良で $M-n_1$ 個の回路が正常、かつ、

2 番目の行の n_2 個の回路が不良で $M-n_2$ 個の回路が正常、かつ、

⋮

H 番目の行の n_H 個の回路が不良で $M-n_H$ 個の回路が正常、かつ、

冗長な回路の n_p 個の回路が不良で $M-n_p$ 個の回路が正常]

$$= \sum_{n_1+n_2+\dots+n_H+n_p \leq R} \{ {}_M C_{n_1} (1-Y_0^{A_1/A_0})^{n_1} Y_0^{(M-n_1)A_1/A_0} \times \\ {}_M C_{n_2} (1-Y_0^{A_2/A_0})^{n_2} Y_0^{(M-n_2)A_2/A_0} \times \\ \vdots \\ {}_M C_{n_H} (1-Y_0^{A_H/A_0})^{n_H} Y_0^{(M-n_H)A_H/A_0} \times \\ {}_M C_{n_p} (1-Y_0^{A_p/A_0})^{n_p} Y_0^{(M-n_p)A_p/A_0} \} \\ = \sum_{n_1=0}^{\min(M,R)} \sum_{n_2=0}^{\min(M,R-n_1)} \dots \sum_{n_H=0}^{\min(M,R-n_1-\dots-n_{H-1})} \sum_{n_p=0}^{R-n_1-\dots-n_H} \\ \{ {}_M C_{n_1} (1-Y_0^{A_1/A_0})^{n_1} Y_0^{(M-n_1)A_1/A_0} \times \\ {}_M C_{n_2} (1-Y_0^{A_2/A_0})^{n_2} Y_0^{(M-n_2)A_2/A_0} \times \\ \vdots \\ {}_M C_{n_H} (1-Y_0^{A_H/A_0})^{n_H} Y_0^{(M-n_H)A_H/A_0} \times \\ {}_M C_{n_p} (1-Y_0^{A_p/A_0})^{n_p} Y_0^{(M-n_p)A_p/A_0} \} \quad (2)$$

上式において、 $\min(M, n)$ は、 $R > M$ の場合に、 $n_1, n_2, \dots, n_H \leq M$ とするための処置である。常に $R \leq M$ が成り立つならば、式(2)は次のように簡単化出来る。

$$Y_R = \sum_{n_1=0}^R \sum_{n_2=0}^{R-n_1} \sum_{n_3=0}^{R-n_1-n_2} \dots \sum_{n_H=0}^{R-n_1-n_2-\dots-n_{H-1}} \sum_{n_p=0}^{R-n_1-n_2-\dots-n_H} \\ \{ {}_M C_{n_1} (1-Y_0^{A_1/A_0})^{n_1} Y_0^{(M-n_1)A_1/A_0} \times \\ {}_M C_{n_2} (1-Y_0^{A_2/A_0})^{n_2} Y_0^{(M-n_2)A_2/A_0} \times \\ \vdots \\ {}_M C_{n_H} (1-Y_0^{A_H/A_0})^{n_H} Y_0^{(M-n_H)A_H/A_0} \times \\ {}_M C_{n_p} (1-Y_0^{A_p/A_0})^{n_p} Y_0^{(M-n_p)A_p/A_0} \} \quad (3)$$

式(2)または式(3)の計算では、 M, H, R の値が大きくなると、計算時間が非現実的な値となる。この場合は、いくつかの行を1つにまとめて、そこに含まれる基本回路のチップ面積は皆等しい(平均のチップ面積を有する)と

して計算している。この場合は、行のまとめ方を適切に行うと、近似計算を行っても十分な精度が得られることが確認されている。

4. 冗長化FFTプロセッサの設計

4.1 設計手順

本稿では、次の条件の下に最適なシステム構成の冗長化FFTプロセッサを設計している。

条件2：最適なシステム構成とは冗長なチップ面積が最小となる構成をいう。

条件3：冗長化後の歩留り Y_R は基準LSIの歩留り Y_0 に等しくとる。

設計手順は次のようになる。

手順1：式(2)または式(3)を用いて $Y_R = Y_0$ を与える冗長な基本回路(バタフライの個数 R を求め、冗長なチップ面積 $A_R = RA_P$ を計算する。

手順2：手順1を繰り返し、 A_R が最小となる R の値を求める。 R が定まれば、 H 、 M 、 A_i ($i=1 \sim H$) など、システム構成を示すパラメータが一意に定まる。

FFTプロセッサを冗長化する場合、システム全体を一体として冗長化を行うと、1個の冗長な回路が、システム内の全ての基本回路に対して結線する必要があり、一般に冗長な配線のチップ面積の合計値が非常に大きくなる。条件2を満たすには、この冗長な配線のチップ面積を出来る限り小さくする必要がある。そこで、システム全体を、いくつかのサブシステムに分割し、サブシステムごとに冗長化を行う方法を検討している。この場合、冗長な回路は特定のサブシステムに所属し、そのサブシステム内の回路とだけ置き換えが可能とする。この方法では冗長な回路の個数、および冗長な配線の本数がいくらか増加するが、冗長な配線1本の長さが大きく減少するので、冗長なチップ面積の合計は減少する可能性がある。ただし、縦方向の1行のバタフライを、いくつかのサブシステムに分割しようとする、バタフライ間の相互配

線に起因する制約が生ずる。すなわち、バタフライ演算の最後段のバタフライ1行では、バタフライの相互配線は、行の上半分の位置にあるバタフライと下半分の位置にあるバタフライの間で結線されるので、この1行はサブシステム分割をすることが出来ない。同様に後ろから2行目については、2個に分割できるが3個以上に分割することは出来ない。以下、一般に、後ろから n 列目は、 $2^n - 1$ 以上に分割することは出来ない。このようなサブシステム分割を行った場合の設計は、手順1を以下のように修正する。まず、サブシステム分割数を B 、 i 番目のサブシステム部分の冗長化後の歩留りを Y_i 、 i 番目のサブシステムの非冗長なチップ面積の合計値を A_{i1} 、 i 番目のサブシステムで必要な、冗長な配線、切り替えスイッチを含めた冗長なバタフライ1個分のチップ面積を A_{P1} とする。このとき、条件3より、次式が成り立つ必要がある。

$$Y_1 Y_2 \cdots Y_{B-1} Y_B = Y_0 \quad (4)$$

各 Y_i の選択には自由度が存在するが、ここでは、 Y_i は A_{i1} によって決まる次のような値に取ることにする。

$$A_i = A_{i1} + A_{i2} + \cdots + A_{iB-1} + A_{iB}$$
$$Y_i = Y_0^{A_{i1}/A_i} \quad (i=1 \sim B) \quad (5)$$

手順1を、次の手順に置き換える。

手順1'_1：1個のサブシステムについて、式(2)または式(3)を用いて $Y_R = Y_i$ となる冗長な基本回路の個数 R を求め、これを R_i とおく。

手順1'_2：手順2'_1を $i=1 \sim B$ の間繰り返す。

手順1'_3：冗長なチップ面積 $A_R = R_1 A_{P1} + R_2 A_{P2} + \cdots + R_B A_{PB}$ とする。

4.2 冗長化設計結果

表4に冗長化設計結果を示している。表には、システムを一体としたときと、最適なサブシ

テム分割を行ったときの、冗長なチップ面積の内訳、すなわち、冗長なバタフライの面積、冗長な配線の面積、および切り替えスイッチのチップ面積を示している。また、同表の冗長な面積の増加率は、「冗長な面積の合計/非冗長な

FFTの面積」を表している。また、サブシステム分割の例として、データ点数1024点、演算精度16bitのFFTの最適なサブシステム分割の詳細を図6に示している。

表-4 冗長化設計結果

データ 点数	話長	サブシ ステム 分割数	冗長なバ タフライ の数	冗長なバタ フライの 面積[λ^2]	冗長な配線 の面積 [λ^2]	スイッチの 面積 [λ^2]	冗長な面積 の合計 [λ^2]	冗長な面 積の増加 率[%]
512	12	1	22	8.24×10^8	6.23×10^9	3.21×10^8	7.37×10^9	8.2
		7	35	1.31×10^9	1.57×10^9	8.11×10^7	2.96×10^9	3.3
	16	1	40	2.58×10^9	1.89×10^{10}	7.79×10^8	2.22×10^{10}	14.3
		15	82	5.28×10^9	2.78×10^9	1.15×10^8	8.17×10^9	5.3
	20	1	65	6.39×10^9	4.66×10^{10}	1.58×10^9	5.45×10^{10}	23.0
		15	109	1.07×10^{10}	5.66×10^9	1.92×10^8	1.66×10^{10}	7.0
1024	12	1	56	2.09×10^9	3.52×10^{10}	1.82×10^9	3.91×10^{10}	18.8
		15	94	3.52×10^9	4.34×10^9	2.24×10^8	8.08×10^9	3.9
	16	1	106	6.83×10^9	1.11×10^{11}	4.59×10^9	1.22×10^{11}	34.3
		31	202	1.30×10^{10}	8.07×10^9	3.33×10^8	2.14×10^{10}	6.0
	20	1	185	1.82×10^{10}	2.94×10^{11}	1.00×10^{10}	3.23×10^{11}	59.4
		31	263	2.59×10^{10}	1.62×10^{10}	5.52×10^8	4.26×10^{10}	7.8
2048	12	1	157	5.87×10^9	2.17×10^{11}	1.12×10^{10}	2.34×10^{11}	48.1
		63	320	1.20×10^{10}	9.81×10^9	5.06×10^8	2.23×10^{10}	4.6
	16	1	337	2.17×10^{10}	7.77×10^{11}	3.21×10^{10}	8.31×10^{11}	99.9
		63	495	3.19×10^{10}	2.58×10^{10}	1.06×10^9	5.87×10^{10}	7.1
	20	1	683	6.71×10^{10}	2.39×10^{12}	8.12×10^{10}	2.54×10^{12}	200.0
		63	649	6.38×10^{10}	5.38×10^{10}	1.83×10^9	1.19×10^{11}	9.4

16	8	4	sub-system 2	sub-system 1
17				
18	9	5	sub-system 3	
19				
20	10	6	sub-system 3	
21				
22	11	7	sub-system 3	
23				
24	12	7	sub-system 3	
25				
26	13	7	sub-system 3	
27				
28	14	7	sub-system 3	
29				
30	15	7	sub-system 3	
31				

1~6th, 7th, 8th, 9th, 10th stage

図-6 FFTプロセッサの最適サブシステム分割(1028点16bit)

Fig.6 Optimum subsystem dividing of the FFT processor (1028points, 16bits)

5. まとめ

本稿では、画像信号をリアルタイムで処理可能なWSI規模のFFTプロセッサを、データ点数と語長を変化させて冗長化設計を行った結果を報告した。その際、バタフライ間の相互配線、冗長な配線、及び切り替えスイッチのチップ面積を重視した冗長化を行っている。通常、システムの冗長化の検討には、相互配線の面積、特に冗長な相互配線の面積は無視されることが多い。

冗長化設計の結果、WSI規模のチップ面積を有するFFTプロセッサは、冗長化によるチップ面積の増加が、最大でも非冗長なチップ面積の9.4%あれば、基準チップ面積の歩留り50%で構成できることが明らかとなった。

なお、本稿では欠陥分布はランダムと仮定して設計を行っているが、LSIの欠陥は集中的に発生する傾向があり、このような場合については今後検討する予定である。

謝辞

ご指導頂いた、八戸工業大学松坂知行教授に感謝致します。

参考文献

- 1) Swartzlander E. E. : Wafer Scale Integration, Kluwer Academic Publishers(1989).
- 2) Mead, C, Conway, L : Introduction to VLSI systems, Addison-Wesley Publishing Company Inc(1980).
- 3) 松山泰男, 富沢孝: VLSI設計入門, 共立出版社(1983).
- 4) 苔米地宣裕: 冗長化WSIシステムの歩留り評価指標, 信学論(D-I), J78-D-I No. 3, 383/392(1995-03).
- 5) Raffel J. I. : Laser linking for defect avoidance and customization, Proc. SPIE, Vol. 774, 93/100(1987).
- 6) 池田正幸, 藤岡知夫他: レーザプロセス技術ハンドブック, 朝倉書店(1992).
- 7) Stapper C. H. : Defect Density Distribution for LSI Yield Calculation, IEEE Trans., ED-20 No. 7, 655/657(1973).
- 8) 柳井久義, 永田譲: 集積回路工学(2), コロナ社(1979).
- 9) 苔米地宣裕: ブロック構造VLSIの冗長化による歩留り向上とその冗長化設計法: 信学論(D-I), J79-D-I No. 5, 289/298(1996).