冗長2進数系に基づく演算回路の速度とチップ面積

苫米地　宣　裕*

Consideration on the Operation Speed and the Chip Size of the Arithmetic Circuits Based on the Redundant Binary Number System

Nobuhiro TOMABECHI

Abstract

In this paper, the arithmetic circuits based on the redundant binary number system are compared with ones based on the usual binary number system from the view points of operation speed and chip size. Following results are obtained, where N denotes the length of digits and the ratio shows the value of the redundant binary arithmetic circuits divided by that of the usual binary arithmetic circuits. (1) The ratio of chip size is about 4 times and is independent on the type of circuits and the value of N. (2) The ratio of operation speed of the adder is N times. (3) The ratio of operation speed of the multiplier is N times when pipelining operation is available, and is $3N/\log_2(N)$ times when pipelining operation is unavailable.

Key words: redundant binary number system/arithmetic circuit/speed/chip size/comparison

1. まえがき

冗長2進数系 [1] は，多桁の加算（乗算も部分積の加算となる）において，桁上げが桁の長さによらずただ1回しか生じないという特長を有しており，暗号演算のように桁数が極端に多くしかも高速性が要求される演算に適している。一方，冗長2進数系には，通常の2進数系に比較して回路の規模が大きくなる，あるいはLSI化したときのチップ面積が大きくなるという問題点も存する。このため，冗長2進数系を実際のシステムに応用したという報告はあまり多くないようである。

本稿では，冗長2進数系と通常の2進数系の比較を，演算速度と回路のチップ面積の観点から行った。その結果，次のような知見が得られた。桁数を N と表すと，冗長2進演算回路は通常の2進演算回路に比較して，チップ面積が回路の種類や N の値によらず約4倍となる。演算速度は，① 加算回路については N 倍，② 乗算回路については，パイプライン処理が可能なときは N 倍，パイプライン処理ができないときは $3N/\log_2(N)$ 倍となる。

2. 冗長2進数系の定義と演算

冗長2進数系は，1桁の値を1，0，−1の3つの値で表す数体系である。この表現をとることにより，加算において桁上げが複数の桁にわたって次々に伝播することを防ぐことができる。すなわち，各桁の加算を行うとき，一つ下位の桁からの桁上げが発生する可能性があるか否かを調べ，桁上げが発生する可能性がある場合には，その桁の値を0，または−1とすることによって，連続桁上げの発生を防ぐことができる。

本稿では，冗長2進数を x^* のように記号＊を付けて表す。以下，簡単のため x^* は正の整数に限定して論ずる。1桁の冗長2進数 x^* は2ビットで表される。これを，$x^*=(x^+, x^-)$ と表記し，かつ，$(1, 0)=1$，$(0, 0)=0$，$(0, 1)=-1$ と対応づける。

冗長2進数 x^*, y^* の加算は，表1に従って行われる[2]。すなわち，

① 桁 (i)，$(i-1)$ で，表1に基づいて中間和 Z_i^* と桁上げ C^*_i を求める。

② 中間和 Z_i^* と下位からの桁上げ C^*_{i-1} を加算し和出力 S^*_i とする。

表1の構成より，② の演算では，桁上げが連続しないことが分かる。従って，多桁の加算は，桁数によらず，1

表1　冗長2進加算

被加数 x^*_i	加数 y^*_i	1つ下位の桁 x^*_{i-1}, y^*_{i-1}	桁上げ C^*_i	中間和 Z^*_i
1	1	—	1	0
1	0	両方とも非負	1	−1
0	1	少なくとも一方負	0	1
0	0	—	0	0
1	−1			
−1	1			
0	−1	両方とも非負	0	−1
−1	0	少なくとも一方負	−1	1
−1	−1	—	−1	0

平成12年12月21日受理

*八戸工業大学　システム情報工学科　教授

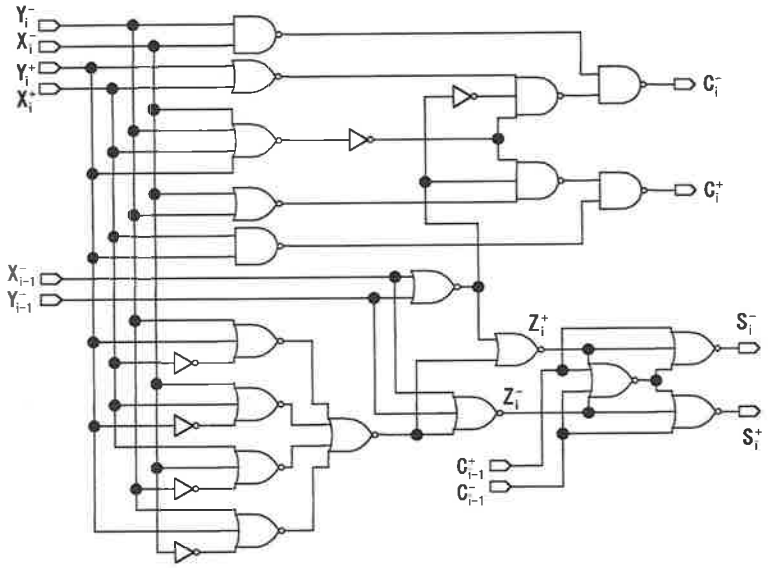

図1　冗長2進加算器の回路

桁の加算と同一の演算時間で終了する。

冗長2進数 x^*, y^* の乗算は，通常の符号付き乗算の要領で行われる。すなわち，1桁の乗算は表2のように行われる。多桁の乗算は，各桁ごとに符号付き乗算を行って部分積を求め，この部分積の加算を行って求める。このとき，加算では前述のように連続桁上げが発生しないので，乗算の演算時間は次のようになる。

N 桁の乗算時間＝1桁の乗算時間＋部分積加算の段数×1桁の加算時間

表2　冗長2進乗算

被乗数 x^*	乗数 y^*	積 P^*
−1	−1	1
−1	0	0
−1	1	−1
0	−1	0
0	0	0
0	1	0
1	−1	−1
1	0	0
1	1	1

3.　冗長2進加算回路・乗算回路の構成

3.1　冗長2進加算回路

1桁の冗長2進加算器（以下，冗長2進FAという）の入力を $x^*_i=(x_i^+, x_i^-)$ および $y^*_i=(y_i^+, y_i^-)$，一つ前の桁の入力を $x^*_{i-1}=(x^+_{i-1}, x^-_{i-1})$ および $y^*_{i-1}=(y^+_{i-1}, y^-_{i-1})$，中間和を $Z^*_i=(Z^+_i, Z^-_i)$，桁上げ出力を $C^*_i=(C^+_i, C^-_i)$，一つ前の桁の桁上げ出力を $C^*_{i-1}=(C^+_{i-1}, C^-_{i-1})$，和出力を $S^*_i=(S^+_i, S^-_i)$ と表す。表1より Z^*_i, C^*_i, S^*_i はそれぞれ次のように求められる[3]，[4]。ただし，＋はOR演算を表している。

$$U=x^+_i\overline{y^+_i}\,\overline{y^-_i}+\overline{x^+_i}\,\overline{x^-_i}y^+_i+\overline{x^+_i}\,\overline{x^-_i}y^-_i+x^-_i\overline{y^+_i}\,\overline{y^-_i}$$
$$Z^+_i=(x^-_{i-1}+y^-_{i-1})U$$
$$Z^-_i=\overline{(x^-_{i-1}+y^-_{i-1})}\,U$$
$$W=x^+_i+x^-_i+y^+_i+y^-_i$$
$$C^+_i=\overline{x^-_i}\,\overline{y^-_i}\,W\overline{x^-_{i-1}}\,\overline{y^-_{i-1}}+x^+_iy^+_i$$
$$C^-_i=\overline{x^+_i}\,\overline{y^+_i}\,W\overline{x^-_{i-1}\,y^-_{i-1}}+x^-_iy^-_i$$
$$S^+_i=\overline{Z^-_i}\,\overline{C^-_{i-1}}(Z^+_i+Z^-_i+C^+_{i-1}+C^-_{i-1})$$
$$S^-_i=\overline{Z^+_i}\,\overline{C^+_{i-1}}(Z^+_i+Z^-_i+C^+_{i-1}+C^-_{i-1})$$

上式に基づいて構成した冗長2進FAの回路を図1に示している。ただし，図は x^*_i, y^*_i から S^*_i に至る最長経路のゲートの段数ができるだけ少なくなるよう式の変形を行って求めている。図1の回路は，FPGAを用いて構成し，正常に動作することを確認している。

回路の演算速度を決める要因にはゲート・ディレイと配線ディレイがあるが，配線ディレイは製造プロセスに依存するので一概に決め難い。本稿では，ゲート・ディレイのみ考慮することとし，演算速度を決定する最長経路のゲートの段数（以下，「最長経路ゲート段数」という）で演算速度の比較を行うこととする。図1より，次の結果が得られる。

［結果1］　冗長2進FAの最長経路ゲート段数は6となる。

なお，N桁の冗長2進加算回路の演算時間は冗長2進FA1個の演算時間に等しい。

3.2 冗長2進乗算回路

1桁の冗長2進乗算器の入力を $x^*=(x^+, x^-)$ および $y^*=(y^+, y^-)$，積出力を $P^*=(P^+, P^-)$ と表す。P^* は次のようになる。ただし，＋はOR演算を表している。

$$P^+=x^+y^+ + x^-y^-$$
$$P^-=x^+y^- + x^-y^+$$

上式に基づいて，冗長2進乗算器の回路はNANDゲート2段，合計ゲート数6個で構成することができる。よって次が言える。

［結果2］ 1桁の冗長2進乗算器の最長経路ゲート段数は2となる。

N 桁の冗長2進乗算回路は，$N\times N$ 個の1桁冗長2進乗算器と「N 個の部分積加算回路」で構成される。このとき，「N 個の部分積加算回路」の接続をどのように行うかが問題となる。

(1) パイプライン処理が可能なとき

この場合は，配列乗算回路，すなわち部分積を逐次加算する構成が採用できる。このとき，演算時間は1桁の加算時間，すなわち冗長2進FA1個の演算時間と等しくなる。

［結果3］ N 桁の冗長2進乗算回路の最長経路ゲート段数は，パイプライン処理が可能なとき（配列乗算回路のとき）は6となる。

ただし，パイプライン内におけるデータの滞留時間は $N\times$（1桁の冗長2進乗算器の演算時間＋冗長2進FAの演算時間）となる。

(2) パイプライン処理ができないとき

配列乗算回路を用いると，部分積の加算が N 段となり，冗長2進数系の特長が十分に生かされない。冗長2進数系の特長を最大限に生かすには加算回路の2進木構造を用いる。この場合，部分積の加算の段数は $\log_2(N)$ となる。よって次が得られる。

［結果4］ N 桁の冗長2進乗算回路の最長経路ゲート段数は，パイプライン処理ができないとき（2進木構造のとき）は $(2+6\log_2(N))$ となる。

4. 冗長2進加算回路・乗算回路のレイアウト

レイアウト設計規則は λ ルール[5]を用いる。ただし，2層配線を用いる。図2に，冗長2進FAのレイアウトを

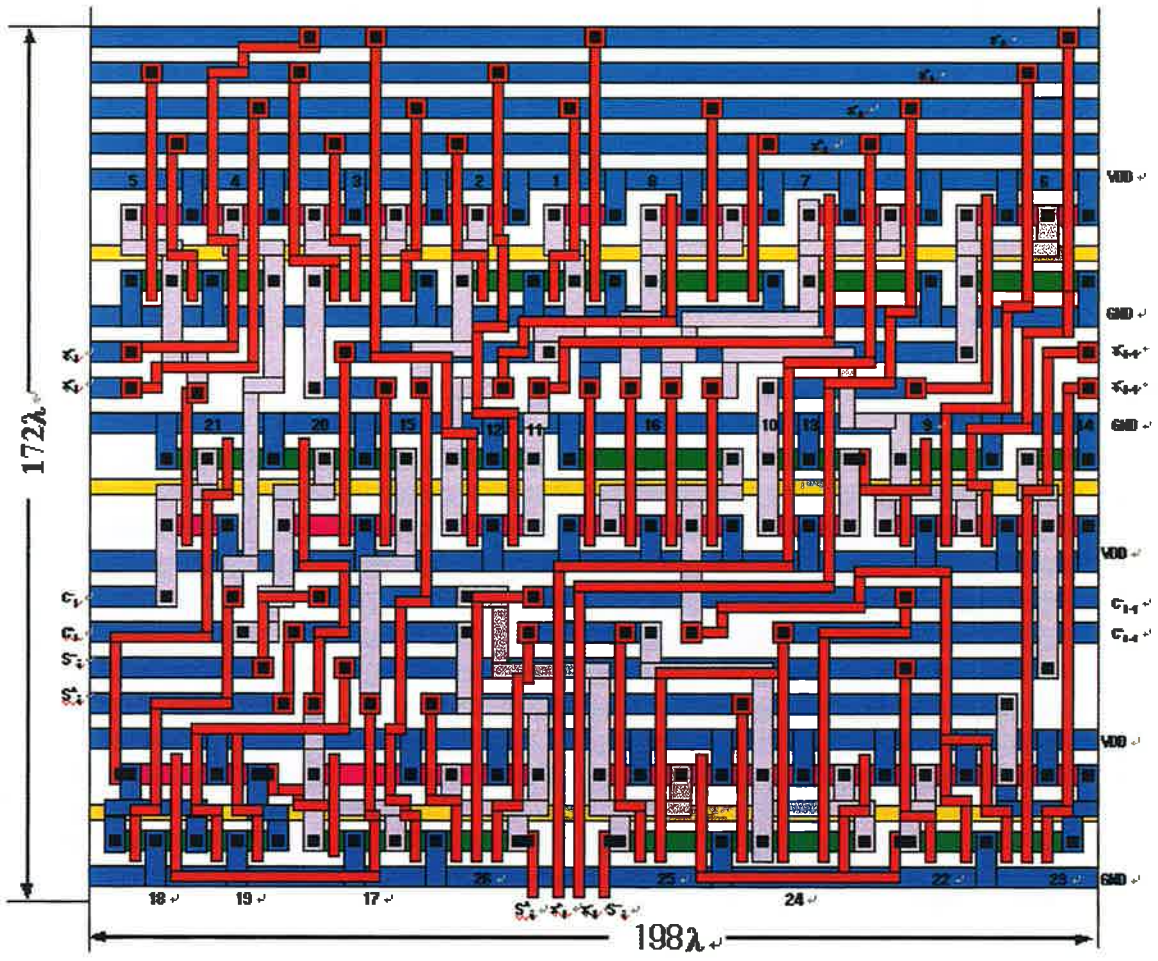

図2 冗長2進FAのレイアウト

示している。ここで，λは基準寸法を表している。なお，図はレイアウトが見易いようにポリシリコン層を最上位に描いている。図2より次の結果が得られる。

［結果5］　冗長2進FAのチップ面積は$(172\lambda\times198\lambda)$となる。

N桁の冗長2進加算回路のチップ面積は，(N×冗長2進FAの面積）となる。

図3は1桁の冗長2進乗算器1個のレイアウト，図4は冗長2進乗算器2個1組のレイアウトを示している。図3，図4は，図2に示した冗長2進FAに接続されるように横幅を合わせている。

配列乗算回路によるN桁の冗長2進乗算回路のレイアウトは，(1桁の冗長2進乗算器1個＋冗長2進FA)を$(N\times N)$個配列したものとなる。ただし，1桁ずつずれた平行四辺形の配置となる。レイアウトを規則的なものとするため，ここでは，次のように計算する。

配列乗算回路のチップ面積＝(1桁の冗長2進乗算器＋冗長2進FA)の面積$\times N\times 2N$

このとき，次が得られる。

［結果6］　配列乗算回路によるN桁の冗長2進乗算回路のチップ面積は，$2N^2(234\lambda\times198\lambda)$となる。

2進木構造によるN桁の冗長2進乗算回路のレイアウトを，図5に示している。図において，乗算器は2個1組を用いている。2進木構造と配列乗算回路との大きな違いは，2進木構造ではビット方向配線領域が必要となることである。配線の数は，1桁2ビットなので，$2\log_2(N)$本となる。図5より次の結果が得られる。

［結果7］　2進木構造によるN桁の冗長2進乗算回路のチップ面積は，$2N^2(215\lambda\times(198+14\log_2(N))\lambda)$となる。

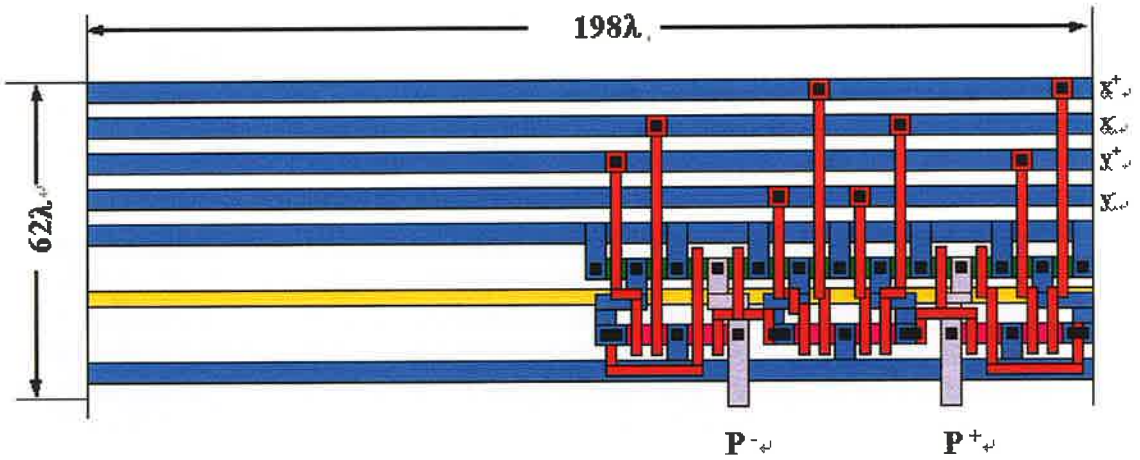

図3　冗長2進乗算器1個のレイアウト

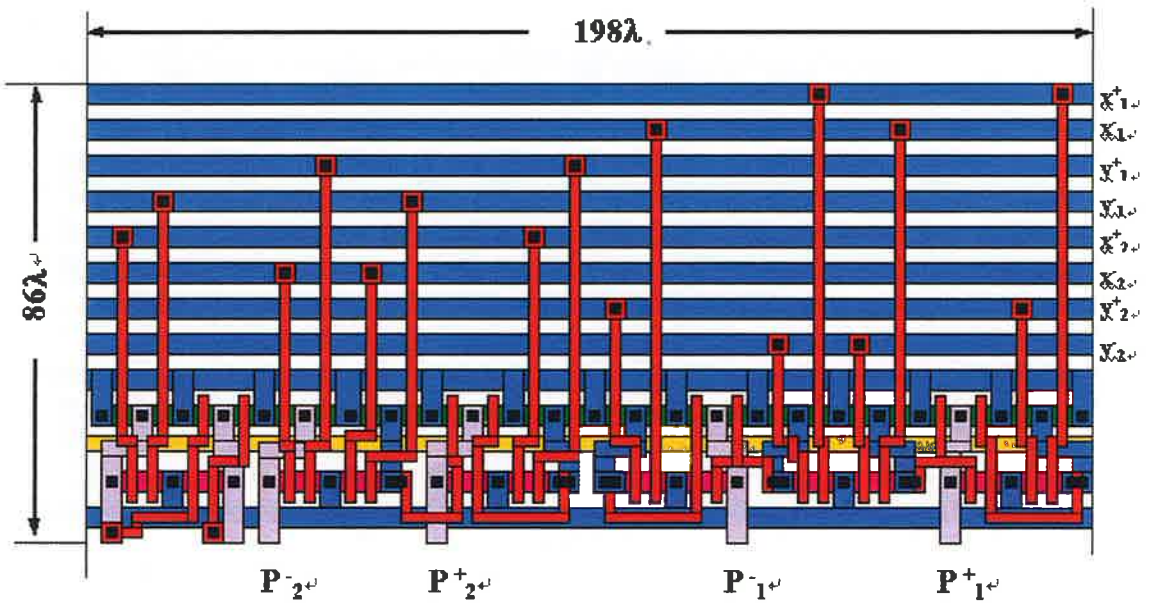

図4　冗長2進乗算器2個1組のレイアウト

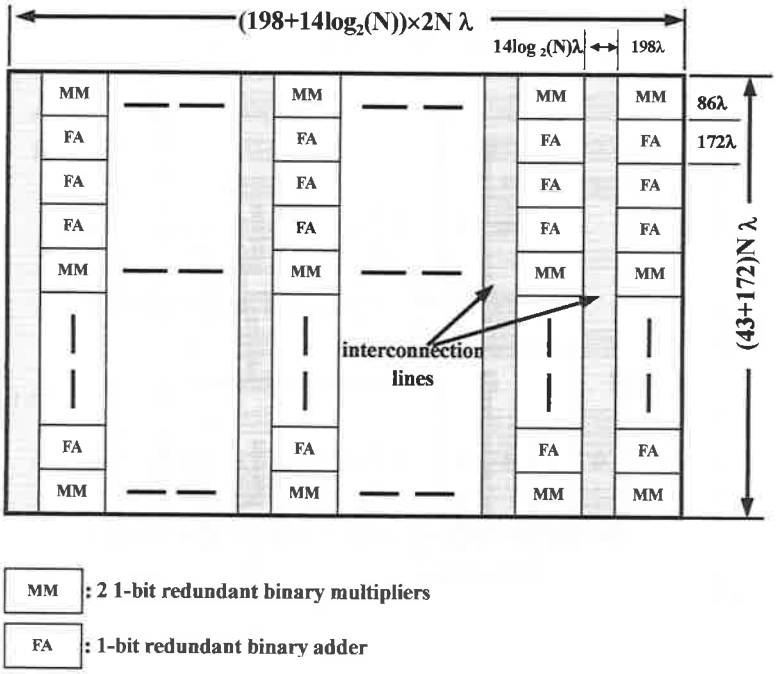

図5　2進木構造を用いた冗長2進乗算回路のレイアウト

5. 通常の2進演算回路との比較

5.1 演算速度の比較

通常の2進数系に基づく1桁の加算器(すなわち,FA)は，演算速度とチップ面積の異なる種々の回路構成がとり得る。ここでは，比較的チップ面積の小さい図6の構成をとる。本回路の最長経路ゲート段数は6となる（和出力に対しては6,桁上げ出力に対しては5となるが,分かり易くするため6ととる)。

通常の2進数系に基づく N 桁の加算回路も,演算速度とチップ面積の異なる種々の構成，リップル・キャリー，キャリー・スキップ，キャリー・ルックアヘッド，その他がとり得るが，ここではチップ面積の最も小さなリップルキャリアダーをとる。本加算回路の最長経路ゲート段数は（N×FAのゲート段数）となる。よって次が得られる。

［結果8］　冗長2進加算回路と通常の2進加算回路の演算速度比は,1桁の場合は1,N 桁の場合は N となる。

通常の2進数系に基づく1桁の乗算器はAND回路1個で構成できる。そのゲート段数は2となる。

通常の2進数系に基づく N 桁の乗算回路としては,配列乗算回路をとることとする。この乗算回路の最長経路ゲート段数は，パイプライン処理が可能なときは（N×FAのゲート段数$=6N$）となり，パイプライン処理ができないときは（1桁の乗算器のゲート段数$+3N$×FAのゲート段数$=2+18N$）となる。よって次が得られる。

［結果9］　冗長2進乗算回路と通常の2進乗算回路の演算速度比は,1桁の場合は1となる。N桁の場合は，パイプライン処理が可能な場合はN，パイプライン処理ができないときは$(1+9N)/(1+3\log_2(N))$となる。

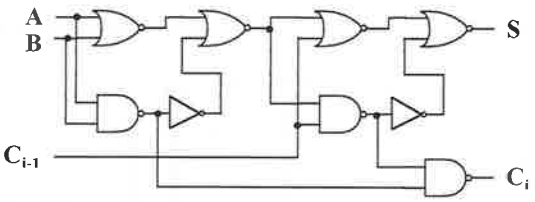

図6　FAの回路

5.2 チップ面積の比較

図7は通常のFAのレイアウトを示している。図より，FAのチップ面積は$(89\lambda\times98\lambda)$となる。通常の2進数系に基づく N 桁の加算回路のチップ面積は，（N×FAのチップ面積）となる。よって次が得られる。

［結果10］　冗長2進加算回路と通常の2進加算回路のチップ面積比は，N の値によらず，$(172\times198)/(89\times98)=3.9$ となる。

図8は，通常の2進数系に基づく乗算器1個を付属したFA(以下，乗算器付FAという)のレイアウトを示している。本回路のチップ面積は$(103\lambda\times112\lambda)$となる。通常の2進数系に基づく N 桁の乗算回路のチップ面積は，配列乗算回路をとるとすると，($2N^2$×乗算器付FAの面積）となる。よって次が得られる。

［結果11］　冗長2進乗算回路と通常の2進乗算回路のチップ面積比は,パイプライン処理が可能なときは,N の値によらず，$(234\times198)/(103\times112)=4.0$ となり，パイプライン処理ができないときは215×(198

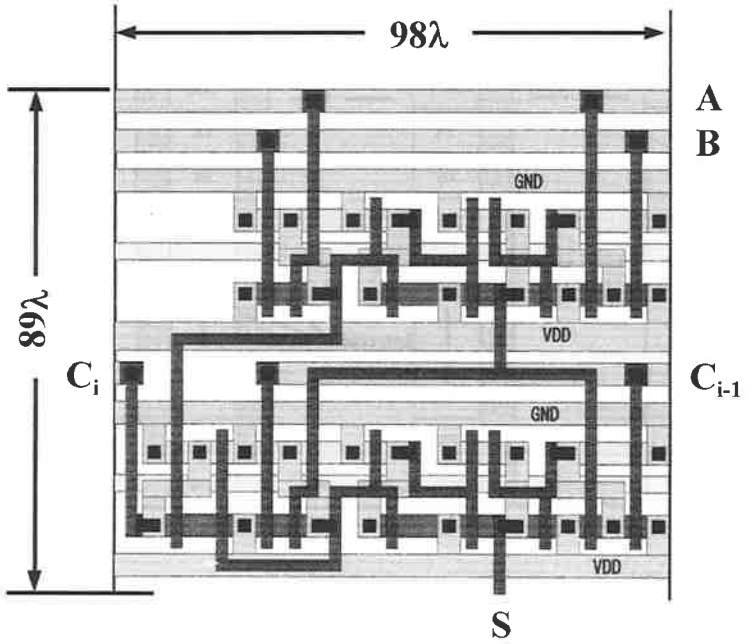

図7　FA のレイアウト

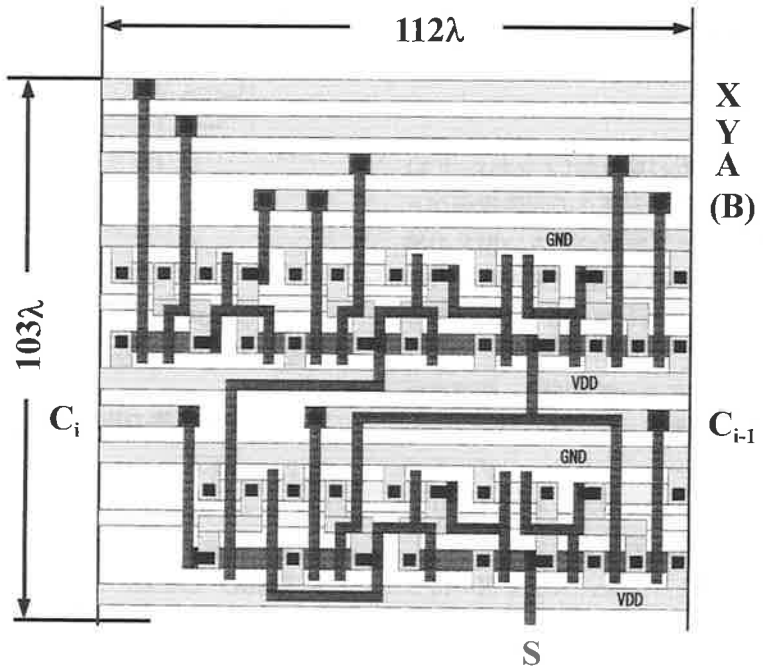

図8　乗算器付 FA のレイアウト

$+14\log_2(N))/(103\times112)$ となる。

5.3 考　　察

1桁の回路に関するデータは積算の基礎として必要ではあるが，比較には N 桁の回路が重要となる。よって，本稿で得られた結果は次のようにまとめることができる。

桁数を N と表すと冗長2進数系は通常の2進数系に比較して，

①　加算回路については，演算速度は N 倍，チップ面積は3.9倍となる。

②　乗算回路について，パイプライン処理が可能なときは，演算速度は N 倍，チップ面積は4.0倍となる。

③　乗算回路について，パイプライン処理ができないときは，演算速度は $(1+9N)/(1+3\log_2(N))$ 倍，チップ面積は $215\times(198+14\log_2(N))/(103\times112)=3.7\times(1+\log_2(N)/14)$ 倍となる。

N の値を変数とした比較を表3に示している。1例として，暗号処理で用いられる $N=1{,}024$ をとると，乗算回

表3　冗長2進演算回路と通常の2進演算回路の比較

	演算速度の比率（冗長2進/2進）			チップ面積の比率（冗長2進/2進）		
	加算回路	乗算回路 PL可能	乗算回路 PL不可	加算回路	乗算回路 PL可能	乗算回路 PL不可
N=16	16	16	11	3.9	4.0	4.8
N=64	64	64	30	3.9	4.0	5.3
N=256	256	256	92	3.9	4.0	5.8
N=1024	1024	1024	297	3.9	4.0	6.3

注：PL可能：パイプライン処理可能な場合を表す。
　　PL不可：パイプライン処理不可の場合を表す。

路の演算速度は，パイプライン処理が可能なときは1,024，パイプライン処理ができないときは297となる。冗長2進数系の効果が非常に大きいことが分かる。また，パイプライン処理が可能なときの方が冗長2進数系の特長が一層発揮できることが分かる。

本稿ではゲート・ディレイだけを比較したが，演算速度を決める要因にはゲート・ディレイと配線ディレイがある。パイプライン処理が可能なときは，信号は隣接した回路に伝達されるので配線ディレイは小さい。しかし，パイプライン処理ができなくて2進木構造を用いるときは，配線が長くなり配線ディレイが大きな問題となる。また，チップ面積が大きいほど配線ディレイが大きくなる。この配線ディレイの観点からも，パイプライン処理が可能なときの方が冗長2進数系の特長が発揮できることが分かる。

チップ面積は，乗算回路のパイプライン不可の場合でかつNの値が大きい場合を除けば，加算回路，乗算回路の別なく，またNの値によらず，おおよそ4倍になる。冗長2進演算回路は，ゲート数が多くなることだけでなく，1桁を2ビットで表すことがチップ面積を大きくする要因となる。すなわち，通常の2進演算に比較して配線面積が少なくとも2倍になる。従って，冗長2進演算回路では多層配線（3層配線）が有効となる。更に，冗長2進数系（より一般的に言えばSD数系）は多値論理と組み合わせ，1本の信号線で多値の情報を伝送する方法をとればその特長を一層発揮できると考えられる［6］。

なお，演算速度とチップ面積を統合した評価指標として，「演算時間とチップ面積の積」が用いられることもあるが，本稿の場合は，この指標の値がNによって大きく変化するのであまり有効とは言えない。

得られた結果を，更に要約すると次のようになる。

［結論］　冗長2進演算回路は通常の2進演算回路に比較して，チップ面積が回路の種類やNの値によらず約4倍となる。演算速度は，① 加算回路についてはN倍，② 乗算回路については，パイプライン処理が可能なときはN倍，パイプライン処理ができないときは$3N/\log_2(N)$倍となる。

6. むすび

本稿では，冗長2進数系と通常の2進数系の比較を，演算速度と回路のチップ面積の観点から行い，次のような結果を得た。桁数をNと表すと，冗長2進演算回路は通常の2進演算回路に比較して，チップ面積が回路の種類やNの値によらず約4倍となる。演算速度は，① 加算回路についてはN倍，② 乗算回路については，パイプライン処理が可能なときはN倍，パイプライン処理ができないときは$3N/\log_2(N)$倍となる。

なお，演算速度を決める要因にはゲート・ディレイと配線ディレイがあるが，本稿ではゲート・ディレイだけの比較を行った。今後，配線ディレイも含めた比較を検討する予定である。また，本稿では，冗長2進加算器の構成法として最長経路のゲート段数が6となる回路構成を示したが，より小さなゲート段数の回路構成について今後検討する予定である。

本研究は，平成12年度文部省科学研究費（基盤研究C(2)）の補助を受けたことを付記する。

参考文献

[1]　高木，安浦，矢島，“冗長2進加算木を用いたVLSI向き高速乗算器，”信学論D，vol. J66-D，No. 6，pp. 683-690，1983-6.
[2]　富田，村上，“計算機システム工学，”昭晃堂，1988。
[3]　苫米地宣裕，伊藤輝樹，“冗長2進剰余テーブルを用いた高速RSA暗号プロセッサの構成法，”信学技報，Vol. FIIS-99，No. 44，pp. 1-10，1999-1.
[4]　伊藤輝樹，“八戸工業大学大学院修士論文：冗長2進数系に基づく高速RSA暗号プロセッサの構成法，”2000-3.
[5]　国枝博昭，“集積回路設計入門，”コロナ社，1996.
[6]　亀山，魏，樋口，“Signed-Digit数多値演算回路に基づくRSA暗号処理プロセッサの構成，”信学論D，Vol. J71-D，No. 12，pp. 2659-2668，1988-12.