論　文

けた上げ保存加算器で構成された部分積加算部をもつ乗算器のテスト

鬼頭　信貴[†]　　高木　直史[†]

Testability of Multipliers with a Partial Product Compressor Consisting of Carry Save Adders

Nobutaka KITO[†] and Naofumi TAKAGI[†]

あらまし　並列乗算器の主要な構成回路である部分積加算部のテストについて考える．部分積加算部のけた上げ保存加算器の段数を L とすると，部分積加算部が，若干の回路の付加により，単一セル機能故障の仮定において L に比例するパターン数でテストできることを示す．まず，任意の構成の部分積加算部が部分積生成部を含めてたかだか $6L+5$ 個のパターンでテストできることを示し，更に，実用的な構成の部分積加算部がたかだか $2L+9$ 個のパターンでテストできることを示す．これにより，N ビット Wallace 乗算器の部分積加算部が $O(\log N)$ 個のパターンでテスト可能であることが明らかになった．

キーワード　テスト容易化設計，単一セル機能故障，乗算器，VLSI

1. まえがき

VLSI 製造技術及び設計技術の進展により，VLSI チップ上に集積される回路がますます大規模化し，そのテストパターン設計に要するコストが増大している．テストパターン設計のコストを低減するために，VLSI を構成する回路について，テストに必要なパターン数など，テストに関する性質を明らかにすることが重要である．本論文では，けた上げ保存加算器(注1)(CSA) で構成された部分積加算部をもつ並列乗算器について，オペランドサイズとテストに必要なパターン数との間の関係を示す．VLSI の開発においては，要求に応じて様々なオペランドサイズの乗算器が必要であり，オペランドサイズとテストに必要なパターン数との間の関係を明らかにすることは重要である．

並列乗算器は一般に部分積生成部，部分積加算部，最終加算部の三つの部分で構成される．部分積生成部では被乗数と乗数から部分積を生成する．部分積加算部では CSA を用いてけた上げ伝搬なしで部分積を加え合わせ，けた上げ保存形で加算結果を得る．最終加算部はけた上げ伝搬加算器からなり，部分積加算部で得られるけた上げ保存形のけた上げと中間和を加え合わせて最終結果を出力する．

乗算器には部分積加算部の構造により，配列型乗算器，Wallace 乗算器 [1]，4-2 加算木を用いた乗算器 [2] などがある．配列型乗算器では直列に接続した CSA を部分積加算部として用いる．N ビット Wallace 乗算器では，まず N 個の部分積を $\lfloor \frac{N}{3} \rfloor$ 個の CSA を用いて三つずつ並列に加え合わせ，およそ $\frac{2N}{3}$ 個の中間結果を得る．次に，これらの中間結果をおよそ $\frac{2N}{9}$ 個の CSA を用いて三つずつ並列に加え合わせる．このように中間結果を三つずつ並列に加え合わせる計算を繰り返す．4-2 加算木を用いた乗算器では，CSA 2 段からなる 4-2 加算器を二分木状に接続した 4-2 加算木を部分積加算部として用いる．部分積加算部の入出力間の CSA の段数の最大数をレベル数 L とすると，N ビット配列型乗算器では $L=N-2$，Wallace 乗算器では $L \approx \log_{\frac{3}{2}} N$，4-2 加算木を用いた乗算器では $L=2\lceil \log_2(\frac{N}{2}) \rceil$ となる．L が小さいほど高速に部分積加算ができ，高速な乗算器となる．L が最小になるのは Wallace 乗算器であり，高速性が求められる場合に用いられる．

文献 [3]〜[6] などでは，配列型乗算器の C テスト可能な設計を示している．算術演算回路が C テスト可

[†] 名古屋大学大学院情報科学研究科，名古屋市
Department of Information Engineering, Nagoya University, Nagoya-shi, 464–8603 Japan

(注1)：本論文では「けた上げ保留加算器」を「けた上げ保存加算器」とする．

能であるとは，回路がそのオペランドのサイズに依存せず定数個のパターンでテストできることである．文献 [7], [8] では，4-2 加算木の CSA のレベル数 L に比例するパターン数でテスト可能な設計を示している．文献 [9], [10] では，4-2 加算木などの C テスト可能な設計を示している．これまでの研究では，Wallace 木など，4-2 加算木よりレベル数が小さな部分積加算部について，オペランドのサイズとテストに必要なパターン数との間の関係が示されていなかった．

本論文では，単一セル機能故障の仮定において，任意の構造の部分積加算部が，テストのための若干の回路の追加で，L に比例する数のパターンでテストできることを示す．このことから，Wallace 乗算器が $O(\log N)$ 個のパターンでテストできることが明らかになる．まず，任意の構造の部分積加算部が部分積生成部を含めてたかだか $6L+5$ 個のパターンでテストできることを示す．更に，三つの入力が別々の CSA に接続するような CSA をもつ，実用的ではない部分積加算部を除くと，たかだか $2L+9$ 個のパターンでテストできることを示す．

本論文では，図 1 の破線部で示す部分積生成部と部分積加算部のテストを考える．パターンは乗算器の入力とテスト用の 1 ビットの外部入力 e から入力し，最終加算器の出力を観測する．最終加算器として用いられるけた上げ伝搬加算器については，文献 [11] などでテスト容易な構成が提案されている．図 1 のように部分積加算部と最終加算部の間にマルチプレクサを挿入することで，部分積加算部と切り離してテスト可能な構成にできる．

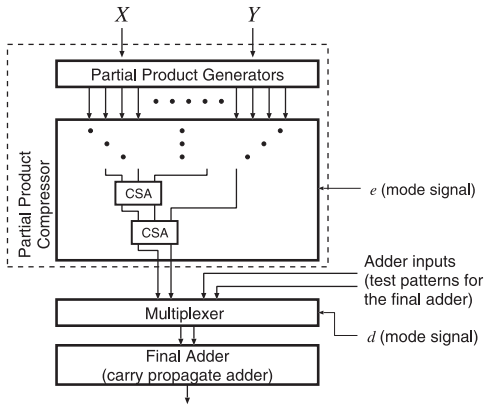

図 1　テスト容易な乗算器の全体構成
Fig. 1　A design of a multiplier.

本論文では，**2.** で準備として並列乗算器と本論文で用いる故障モデルの説明を行う．**3.** では任意の部分積加算部が部分積生成部を含めてたかだか $6L+5$ 個のパターンで，**4.** では実用的な構成の部分積加算部が部分積生成部を含めてたかだか $2L+9$ 個のパターンでテストできることを，パターン集合の構成法を示すことで明らかにする．

2. 準　　備

2.1 並列乗算器

本論文では，符号なし 2 進並列乗算器を対象とする．被乗数を $X=(x_{N-1}\dots x_0)_2$，乗数を $Y=(y_{N-1}\dots y_0)_2$ とする．一般的な並列乗算器は部分積生成部，部分積加算部，最終加算部の三つの部分から構成される．部分積生成部は，被乗数と乗数から部分積 $P_j=X\cdot y_j\cdot 2^j \quad (0\le j<N)$ を生成する．部分積加算部は，けた上げ保存加算器 (CSA) を用いてけた上げ伝搬なしで部分積を加え合わせ，けた上げ保存形で結果を得る．最終加算部は，部分積加算部で得られるけた上げ保存形のけた上げと中間和を足し合わせ，最終結果を出力する．部分積加算部の例を図 2 (a) に示す．

CSA は図 3 (a) のように全加算器 (FA) で構成される．CSA は三つの 2 進数を足し合わせ，加算結果はけた上げ保存形でけた上げと中間和の二つの 2 進数で出力する．出力は両端を除いて各けたで 2 ビットずつ得られる．

CSA を構成する各 FA について，図 3 (a) のように入力端子に u，v，w と名前付けし，出力端子に c，s と名前付けする．また，CSA の入力にも図 3 (b) に示すように U，V，W と名前付けし，出力に C，S と名前付けする．CSA の入出力 U，V，W，C，S はそれぞれ CSA の中のすべての FA の対応する端子 u，v，w，c，s からなる．部分積加算部の入力は，部分積 $P_0,\dots,P_{N-1}$ である．部分積加算部の各 CSA は，区別のために番号付けを行う．図 2 (a) では CSA に 1 から 10 までの番号を付けている．図 2 (b) は，図 2 (a) の CSA2，CSA3，CSA4 の FA の接続関係を示す．CSA3 の FA の c，s 出力は CSA2 の FA の u，v 入力に，CSA4 の FA の c 出力は CSA2 の FA の w 入力に接続しているので，図 2 (a) で CSA3 の C，S は CSA2 の U，V に，CSA4 の C は CSA2 の W に接続している．

部分積加算部に図 2 のように出力側をレベル 1 として，入力に向かってレベル付けする．各 CSA の属す

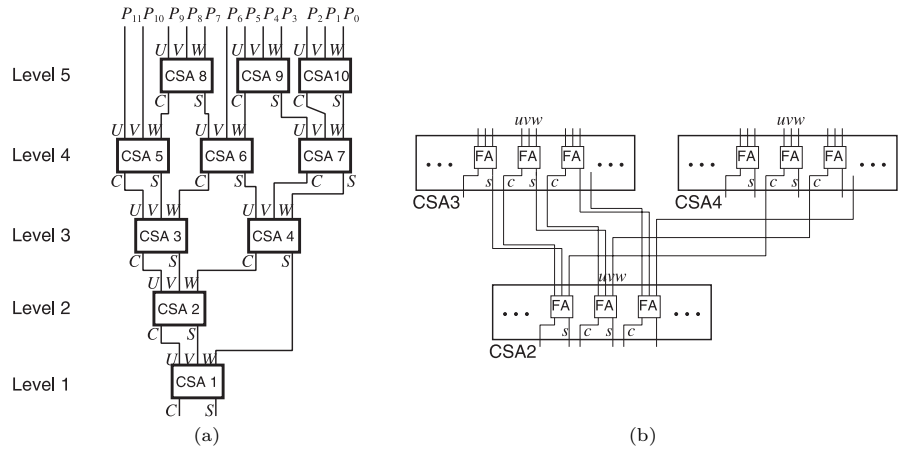

図 2　部分積加算部の例
Fig. 2　An example of a partial product compressor.

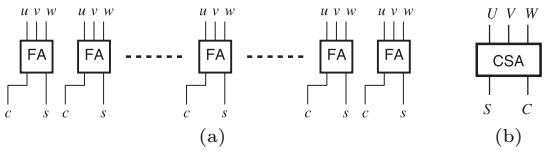

図 3　けた上げ保存加算器
Fig. 3　A carry save adder.

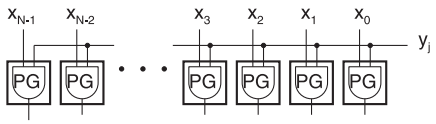

図 4　部分積生成回路
Fig. 4　A partial product generator.

るレベルは，その CSA の出力に接続する CSA の属するレベルの最大値より一つ大きな値とする．部分積加算部の最大のレベルをレベル数 L とする．図 2 では，$L = 5$ である．

2.2　故障モデル

本論文では，単一セル機能故障を仮定する．FA などをセルとして扱い，これらのセルを用いて回路を構成する．単一セル機能故障とは，次の条件を満たす故障である．

- テスト対象回路で故障するのはたかだか一つのセルのみである．
- 故障したセルの出力は現在の入力のみで決まる．
- 故障したセルでは，正常なセルと出力値が異なるような入力が存在する．
- セルに出力端子が複数ある場合には，複数に誤りを出力することがある．例えば，FA の場合 s，c 端子の両方に誤りを出力することがある．

単一セル機能故障を検出するため，次の二つの条件を満たすようにテスト集合を設計する．

- テスト集合の各パターンを外部入力より入力することで，テスト対象回路を構成するすべてのセルの各々に全入力パターンを入力できる．つまり，各セルについて，セルが m 入力のとき 2^m 通りのパターンをすべて入力できる．
- テスト対象回路内のセル故障の影響が外部出力まで伝搬し，外部出力で観察できる．

単一セル機能故障を仮定することにより，各セルのゲートレベルでの実装に依存しないテスト集合を構成できる．単一セル機能故障は，算術演算回路のテストに関する研究で広く用いられている [3]〜[5], [11].

3.　任意の構成の部分積加算部に対するテスト

本論文では，テストにおいて部分積加算部の各入力には部分積として $(00\ldots0)_2$ か $(11\ldots1)_2$ のどちらかのビットパターンを入力する．$(00\ldots0)_2$ と $(11\ldots1)_2$ のビットパターンをそれぞれを **0**，**1** で表す．

各部分積を生成する部分積生成回路を図 4 に示す．部分積生成回路は N 個の部分積生成セル（PG セル）で構成する．各 PG セルは AND ゲートで構成できる．被乗数を $X = (11\ldots1)_2$ とすると，部分積加算部へ入力される部分積 P_j のビットパターンは y_j が 0 か 1 かにより $(00\ldots0)_2$ か $(11\ldots1)_2$ となり，部分積加算

部の各入力へ **0**，**1** を容易に入力できる．

部分積加算部の各入力を **0** か **1** にすると，部分積加算部の各 CSA の各入力と各出力もすべて **0** か **1** となる．以後，CSA の入力パターンを CSA 内の FA の入力ビットを太字で記したもので表現する．例えば，CSA の中の FA に入力端子 u，v，w においてそれぞれ 0，1，1 が入力されているとき，この CSA の入力パターンを **011** で表す．

部分積はビット幅が等しいものの，それぞれの部分積の最下位ビットの位置が異なるため，CSA の両端において入力ビットが二つや一つになるビット位置がある．部分積加算部の内部の CSA についても両端に入力ビットが三つにならないビット位置がある．これらのビット位置についてテストのために入力ビットを補完し，入力ビットパターンを CSA の他のビット位置と同じにする．このとき，CSA のこれらのビット位置でのビット加算にも FA を用いる．

入力ビットの補完は，CSA 内部の FA の入力ビットの配線を分岐させ，入力ビットが不足するビット位置に接続することで実現する．このとき，テスト時のみビットの補完を行うように，入力ビットの補完の配線に補完ビット制御セル（CB セル）を挿入する．CB セルを用いて入力ビットの補完をした CSA の例を図 5 に示す．外部入力信号 e を導入し，各 CB セルに接続することで，補完の制御を行う．通常時には $e = 0$ とし，部分積加算部のテスト時には $e = 1$ とする．CB セルは AND ゲートで実現できる．配線の分岐位置は，通常時 $(e = 0)$ に $X = Y = (11\ldots1)_2$ のとき，分岐元の配線の信号値が 1 となるビット位置をシミュレーションなどで求め，その中から配線長などを考慮して適切なものを選ぶ．

部分積加算部のテストのために，部分積加算部の各 CSA に **000** から **111** までの八つのパターンのすべてを入力できるテスト集合の設計法を考える．

部分積加算部のすべての入力に **0** を入力すると，すべての CSA に **000** が入力される．同様に部分積加算部の全入力に **1** を入力すると，すべての CSA に **111** が入力される．したがって，二つのパターンですべての CSA に **000** と **111** が入力できる．

あるレベル l のすべての CSA に一斉にパターン p $(p \in \{\mathbf{001}, \mathbf{010}, \mathbf{011}, \mathbf{100}, \mathbf{101}, \mathbf{110}\})$ を入力することを考える．CSA は FA で構成され，FA が実現する関数は全射である．そのため，CSA の出力を決めると，その出力になるような CSA の入力が少なくとも一つは存在する．したがって，レベル l に属するすべての CSA に p を入力するための，レベル $l+1$ の CSA への入力パターンを必ず求められる．この性質を再帰的に用いると，レベル l に属するすべての CSA に一斉に p を入力する部分積加算部への入力パターンが得られる．

図 6 にパターンの設計例を示す．図では，レベル 3 に属するすべての CSA (CSA3，CSA4) に一斉に **001** を入力する入力パターンを構成する場合を示している．まず，CSA3 と CSA4 に **001** を入力するための，レベル 4 に属する CSA (CSA5，CSA6，CSA7) へのパターンを一つ求める．CSA5，CSA6，CSA7 へのパターンとして **000**，**101**，**001** や **000**，**110**，**010** などいくつかのパターンがあり，どれを選んでもよい．図では CSA5，CSA6，CSA7 へそれぞれ **000**，**101**，**001** を入力している．次に，レベル 5 に属する CSA8，CSA9，CSA10 に対するパターンを一つ求める．ここでは，それぞれに対してパターン **000**，**101**，**000** を用いる．このようにして，レベル 3 に属するすべての CSA に一斉に **001** を入力する部分積加算部への入力パターン $(P_{11}, \ldots, P_0) = (\mathbf{0},\mathbf{0},\mathbf{0},\mathbf{0},\mathbf{1},\mathbf{0},\mathbf{1},\mathbf{1},\mathbf{0},\mathbf{0},\mathbf{0},\mathbf{1})$ が得られる．このパターンを部分積加算部に入力するには被乗数と乗数を $X = (111111111111)_2$，

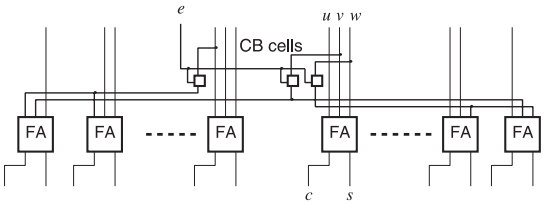

図 5　入力ビットの補完をした CSA の例
Fig. 5　An example of a CSA with padding input bits.

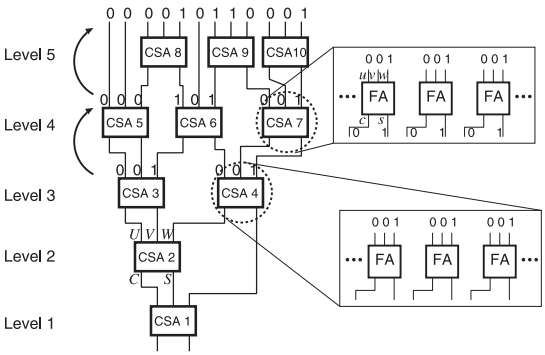

図 6　部分積加算部への入力パターン構成例
Fig. 6　An example of pattern generation.

$Y = (000010110001)_2$ とすればよい．

このように入力パターンを構成すると，一つの入力パターンで一つのレベルのすべての CSA に一斉に同じパターンを入力できる．部分積加算部のすべての CSA に六つのパターンを入力するために，各レベルに六つの入力パターンを生成すると，合計 $6L$ 個の入力パターンとなる．したがって，$6L+2$ 個のパターンで部分積加算部内のすべての FA にすべてのパターンを入力できる．この手法でパターンを構成すると，同じ入力パターンが重複して得られることがあるので，重複する場合は一つを残して他を除去する．

次に，PG セルと CB セルについて考える．これまでに構成したテストパターンでは，X の各ビットは 1 なので，PG セルの x_i 入力は常に 1 である．また，$e = 1$ なので，CB セルの e 入力は常に 1 となる．PG セルと CB セルのすべてのパターンを入力するために，$e = 0$ として，$X = Y = (00\ldots0)_2$ と $X = (00\ldots0)_2$，$Y = (11\ldots1)_2$ と $X = Y = (11\ldots1)_2$ の三つのテストパターンを追加する．一つ目のパターンですべての PG セルと CB セルに 00 を入力できる．二つ目のパターンですべての PG セルに残りのパターンを入力できる．三つ目のパターンですべての CB セルに残りのパターンを入力できる．

故障した FA が重み 2^k のビット位置にあるとき，故障は $\pm 2^k$ または $\pm 2 \cdot 2^k$ または $\pm 3 \cdot 2^k$ の数値の誤りになる．同様に，PG セルや CB セルの故障も数値の誤りとなる．したがって，故障の影響は数値の誤りとして部分積加算部を伝搬する．CB セルにより CSA 内の複数の FA に故障の影響が伝搬することがある．このとき，CB セルは入力ビットが不足した箇所にビットを補完するため，故障の影響はビット補完前の入力の最上位ビットよりも大きいか，あるいは，最下位ビットよりも小さなビット位置に伝搬する．このため，数値の誤りが打ち消され 0 になることはない．したがって，演算結果は故障がない場合の値とは異なる値となり，最終加算器の出力で観測することができる．

以上より，部分積加算部は部分積生成部を含めてたかだか $6L+5$ 個のパターンでテストできる．

4. 実用的な構成の部分積加算部に対する少パターンによるテスト

4.1 CSA 間接続に制約を加えた部分積加算部

部分積加算部において，CSA の三つの入力に別々の CSA を接続すると配線が複雑になるため，実用的な乗算器ではそのような接続を行うことはないと考えられる．そこで，本節では，各 CSA の三つの入力に接続する CSA が最大二つである部分積加算部を対象に考える．本節の制約を加えても，Wallace 木などの一般的な部分積加算部を構成できる．

以降，部分積加算部を構成する CSA を次の三つのタイプに分類する．

- タイプ 1：三つの入力がすべて部分積加算部の入力である CSA
- タイプ 2：他の一つの CSA の出力 S，C の両方が入力する CSA
- タイプ 3：タイプ 1 とタイプ 2 以外の CSA

図 7 において，CSA8，CSA9，CSA10 はタイプ 1，CSA1，CSA2，CSA3，CSA4，CSA7 はタイプ 2，CSA5，CSA6 がタイプ 3 である．

テスト集合の構成の説明を容易とするため，以下を仮定する．CSA の入力の対称性から，これらの仮定をしても一般性を失わない．

- 他の一つの CSA の出力 S，C の両方を入力するタイプ 2 の CSA では，これらの S，C はそれぞれ U，V に接続されている．
- タイプ 3 の CSA では，CSA の入力の一つが部分積加算部の入力であるときは，これは U に入力され，入力の二つが部分積加算部の入力であるときは，これらは U，V に接続されている．

タイプ 3 の CSA において，入力 U は必ず部分積加算部の入力である．図 7 は図 2 の CSA の入力の入換え（端子名の付替え）により得られる．

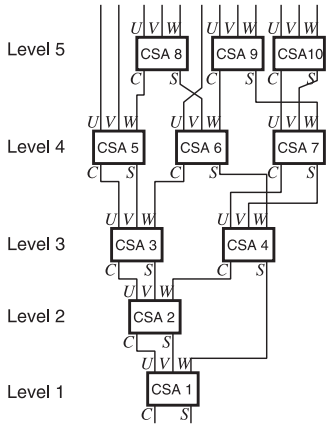

図 7 CSA 間接続に制約を加えた部分積加算部の例
Fig. 7 An example of a partial product compressor under the constraint.

4.2 入力パターン集合の構成法

部分積加算部のテストのための入力パターン集合の構成アルゴリズムを図 8 に示す．アルゴリズムで構成した各入力パターンは，**3.** で示した方法により，部分積生成部を通して部分積加算部に入力できる．

1: P_0 から P_{N-1} が全て **0** の入力パターンと全て **1** の入力パターンを構成し，各 CSA g について $R(g)$ を $\{\mathbf{001}, \mathbf{010}, \mathbf{011}, \mathbf{100}, \mathbf{101}, \mathbf{110}\}$ とする
2: レベル L からレベル 1 へ各レベルに順に注目し，各レベルの CSA に以下の操作を行い 1 つの入力パターンを構成する．
（1）タイプ 1 の各 CSA について，U, V, W に入力する部分積をそれぞれ **1**, **0**, **0** とする
（2）W が部分積加算部の入力となっているタイプ 2 の各 CSA について，W に入力する部分積を **0** とする
（3）V が部分積加算部の入力となっているタイプ 3 の各 CSA について，V に入力する部分積を **0** とする
（4）タイプ 3 の各 CSA について，U に入力する部分積を，V,W の入力が **0**,**0** か **0**,**1** のとき **1**，それ以外のとき **0** とする
得られた入力パターンの反転の入力パターンも構成し，各 CSA g について，$R(g)$ から 2 つの入力パターンにより入力されるパターンを除く
3: レベル L の全ての CSA に一斉に **010** を入力する入力パターンと，その反転の入力パターンの 2 つを構成し，レベル L の各 CSA g について，$R(g)$ から **010** と **101** を除く
4: **for** $l = L$ to 2 **do**
レベル l と $l-1$ の CSA へ入力するパターンを次の手順で決め，**3.** で述べた構成法で入力パターンを 1 つ構成する．
（1） レベル $l-1$ のタイプ 1 の各 CSA g に入力するパターンを，**001** とする
（2） パターンが決まっていないレベル $l-1$ の CSA g を一つ選択し，タイプ 2 なら $R(g)\backslash\{\mathbf{001}, \mathbf{110}\}$ の中の一方，タイプ 3 なら **001** を入力するパターンとする
（3） パターンが決まっていないレベル l の各 CSA g について，接続先のレベル $l-1$ の CSA のパターンが決まっていたら，矛盾しない出力となる入力を $R(g)$ より選び，入力するパターンとする
（4） パターンが決まっていないレベル $l-1$ の各 CSA g について，接続するレベル l の CSA のパターンが決まっていたら，CSA g がタイプ 2 なら $R(g)\backslash\{\mathbf{001}, \mathbf{110}\}$ から，タイプ 3 なら $\{\mathbf{001}, \mathbf{110}\}$ から，矛盾しないパターンを選び，入力するパターンとする
（5） (4) で新たにパターンが決まれば (3) へ戻る．(4) で新たにパターンが決まらず，レベル $l-1$ にパターンが決まっていない CSA があれば (2) へ戻る．
得られた入力パターンの反転の入力パターンも構成し，レベル l, レベル $l-1$ の各 CSA g について，$R(g)$ から決定したパターンとその反転のパターンを除く
end for
5: レベル 1 の CSA g に $R(g)$ の各パターンを入力するための入力パターンを，**3.** で述べた構成法を用いて構成する

図 8 入力パターン集合の構成手順
Fig. 8 An algorithm for pattern set generation.

アルゴリズムでは，各 CSA g に対して，まだ構成していない入力パターンの集合を表す $R(g)$ を用いる．

ステップ 1 では，すべての CSA に **000** と **111** を入力するための二つの入力パターンを構成する．ステップ 1 により，すべての CSA について $R(g)$ は $\{\mathbf{001}, \mathbf{010}, \mathbf{011}, \mathbf{100}, \mathbf{101}, \mathbf{110}\}$ となる．

ステップ 2 では，部分積加算部の入力に直接接続する CSA に着目し，これらの入力を定めることにより，各 CSA に **100**，**010**，**011**，**101** のいずれかを入力する部分積加算部への入力パターンをまず一つ構成する．そして，構成した部分積加算部への入力パターンについて，各部分積の **0** と **1** を入れ換えた反転の入力パターンも構成する．したがって，ステップ 2 では合計二つのパターンを構成する．FA が実現する関数は自己双対関数であり，FA の入力ビットパターンが反転すると出力ビットパターンも反転するため，部分積加算部の入力パターンを反転すると各 CSA に入力するパターンも反転する．

図 9 (a) にステップ 2 で構成した入力パターンを示す．図 9 (a) では，まずレベル 5 の CSA について，(1) より各 CSA の U，V，W に入力する部分積を **1**，**0**，**0** と決めている．次にレベル 4 の CSA について，(3) より CSA5 の V に入力する部分積を **0**，(4) より CSA5，CSA6 の U に入力する部分積をそれぞれ **1**，**0** とし，入力パターンを構成している．その反転の入力パターンが図 9 (b) である．

ステップ 2 で構成する二つの入力パターンにより，タイプ 1 の CSA には **100** とその反転の **011** を入力できる．タイプ 2 とタイプ 3 の CSA には，**100**，**010**，

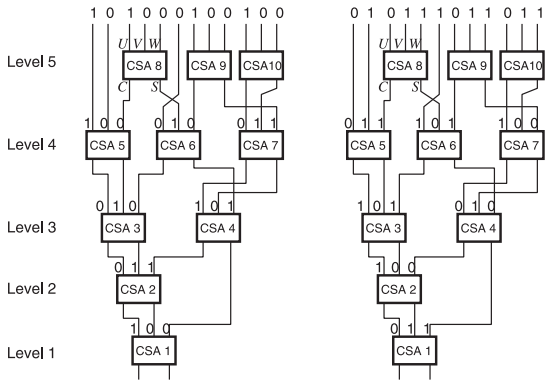

図 9 ステップ 2 による入力パターンの構成例
Fig. 9 An example of pattern generation by step 2.

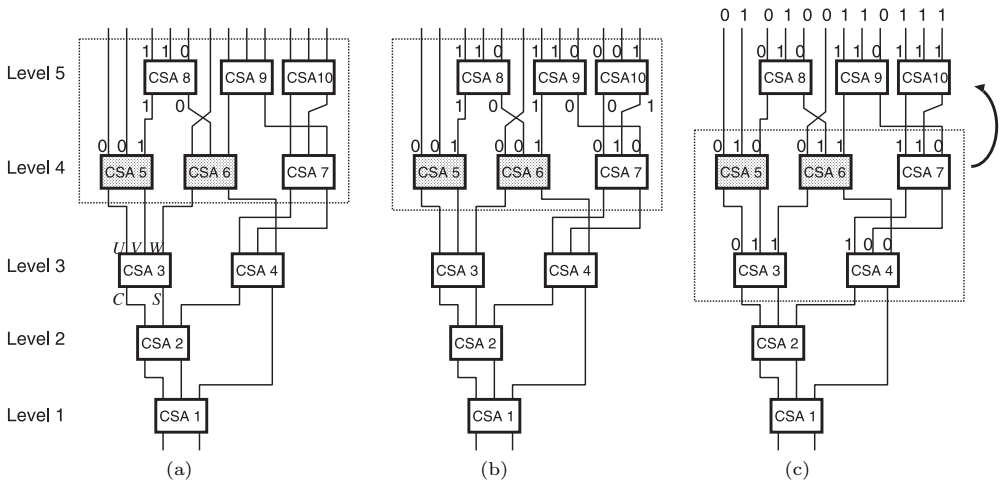

図 10　ステップ 5 による入力パターンの構成例（(a)(b) レベル 5 と 4 へのパターンの構成，(c) レベル 4 と 3 へのパターンの構成）
Fig. 10　Examples of pattern generation by step 5.

101，**011** のいずれかと，その反転のパターンの計二つを入力できる．したがって，各 CSA の $R(g)$ は $\{\mathbf{001}, \mathbf{010}, \mathbf{101}, \mathbf{110}\}$ か $\{\mathbf{001}, \mathbf{011}, \mathbf{100}, \mathbf{110}\}$ になる．レベル L の CSA はすべてタイプ 1 であるため，$R(g)$ は前者となる．

ステップ 3 ではレベル L の CSA に **101** を入力する入力パターンと，その反転の入力パターンを構成する．レベル L の CSA の $R(g)$ は $\{\mathbf{001}, \mathbf{110}\}$ になる．

ステップ 4 では，$l = L$ から 2 まで順に，隣接するレベル l と $l-1$ に注目してパターンを二つずつ生成していく．このとき，レベル l の各 CSA で R は空に，レベル $l-1$ の各 CSA で R の要素数が 2 となる．

図 10 (a)，(b) では，$l = 5$，つまりレベル 5 と 4 の CSA に注目した際のパターンの構成を示す．タイプ 3 の CSA を網かけで示している．図 10 (a) では，(2) で CSA5 へのパターンを **001** に決めて，(3) で接続元の CSA8 の入力パターンを決めた状態を示している．CSA5 の W の入力は **1** なので，CSA8 の $R(8) = \{\mathbf{001}, \mathbf{110}\}$ から，CSA8 のパターンを **110** と決めている．図 10 (b) は，図 10 (a) から更に (4) で CSA6，(3) で CSA9，(4) で CSA7，(3) で CSA10 と CSA のパターンを順に決めた後の状態を示している．

図 10 (c) では，$l = 4$，つまりレベル 4 と 3 の CSA に注目した際のパターンの構成を示す．レベル 3 の CSA3，CSA4 の $R(g)$ はどちらも $\{\mathbf{001}, \mathbf{011}, \mathbf{100}, \mathbf{110}\}$，レベル 4 の CSA5，CSA6，CSA7 の $R(g)$ はそれぞれ，$\{\mathbf{010}, \mathbf{101}\}$，$\{\mathbf{011}, \mathbf{100}\}$，$\{\mathbf{001}, \mathbf{110}\}$ である．このとき，レベル 3 の CSA3 に入力するパターンを (2) で **011** に決めると，CSA5，CSA6 のパターンが決まり，更に CSA4，CSA7 の順にパターンが決まる．最後に，これらのパターンが入力されるように部分積加算部の入力パターンを求めている．

ステップ 4 ではレベル 1 の CSA に $R(g)$ の要素が二つ残る．ステップ 5 では **3.** で示したパターン構成法を用い，それぞれに対してパターンを構成する．

アルゴリズムでは，ステップ 1 からステップ 3 までで各 2 個，ステップ 4 で $2(L-1)$ 個，ステップ 5 で 2 個のパターンを生成する．したがって，部分積加算部のテストに合計で $2L+6$ 個のパターンを生成する．ステップ 4 で構成するパターンに重複があれば一つを残して他を除去する．部分積生成部を含めた部分積加算部のテストに必要なパターン数は，**3.** で示した部分積生成部と補完ビット制御セルのテストのための 3 パターンを加えた，たかだか $2L+9$ 個のパターンである．

4.3　入力パターン集合の最小性

部分積加算部を **0** と **1** で構成したパターンでテストするとき，部分積生成部と併せて **4.2** で示した $2L+9$ 個のパターンすべてがテストに必要になる部分積加算部が存在することを示す．

各 CSA の三つの入力に接続する CSA が最大二つで，タイプ 1 とタイプ 2 の CSA のみが含まれ，更に，レベル l $(1 \leq l < L)$ の CSA が必ずレベル $l+1$ の CSA と接続された部分積加算部を考える．図 11 にそ

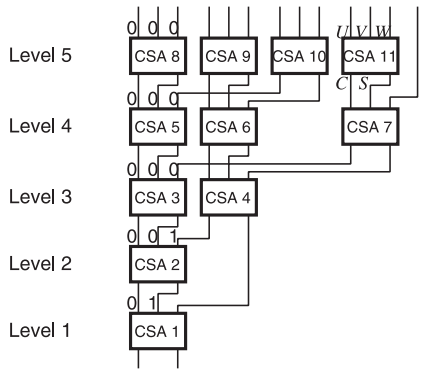

図 11　CSA への **001** の入力の例
Fig. 11　An example of an input pattern to feed **001** to a CSA.

のような部分積加算部の例を示す．

このような部分積加算部では，レベル 1 の CSA の U，V 入力は，レベル 2 のある一つの CSA の C，S 出力と接続する．このレベル 2 の CSA の U，V 入力に接続する，レベル 3 のある一つの CSA があり，同様に繰り返すことでレベル L の CSA まで到達する．図 11 では，レベル 1 からレベル 5 にわたって，CSA1，CSA2，CSA3，CSA5，CSA8 の CSA で U，V 入力と S，C 出力が接続している．

図中の CSA1，CSA2，CSA3，CSA5，CSA8 のような U，V 入力と S，C 出力が連続して接続する CSA 列の，レベル 1 からレベル $L-1$ までの各 CSA に対して **001** を入力することを考える．列内のレベル l $(1 \leq l < L)$ の CSA に **001** を入力するには，列内のレベル $l+1$ から L までの CSA の入力はすべて **000** とする必要がある．図 11 では，例として列内のレベル 2 の CSA2 に **001** を入力する場合の各 CSA の入力を示している．レベル 3 からレベル 5 までの CSA3，CSA5，CSA8 の各入力はすべて **000** となる．このため，列内の複数の CSA に同時に **001** を入力することはできない．したがって，列内の各 CSA に **001** を入力するために $L-1$ 個のパターンが必要である．同様に，**100** を入力するために $L-1$ 個のパターンが必要になる．

この $2(L-1)$ 個のパターンでは列内のレベル L の CSA には **000** と **111** しか入力できず，レベル 1 の CSA には **000** と **111** を入力できない．レベル L の CSA に残りのパターンを入力する 6 個のパターンと，レベル 1 の CSA に **000** と **111** を入力する 2 個のパターンが必要となる．したがって，合計して $2L+6(=2(L-1)+6+2)$ 個のパターンが必ず必要である．このことから，上述の部分積加算部に **0**，**1** を入力してテストするには，少なくとも $2L+6$ 個の入力パターンが必要である．

以上より，部分積加算部と部分積生成部のテストに $2L+9$ 個のパターンが必要となる．

5. むすび

並列乗算器の主要な構成回路である部分積加算部について，テストに必要なパターン数の上界を示した．部分積加算部の CSA の段数を L とすると，部分積加算部を，若干の回路を付加することで，L に比例するパターン数でテストできることを示した．まず，任意の構成の部分積加算部が部分積生成部を含めてたかだか $6L+5$ 個のパターンでテストできることを示し，更に，実用的な構成の部分積加算部を，部分積生成部を含めてたかだか $2L+9$ 個のパターンでテストできることを示した．これにより，4-2 加算木より CSA の段数が小さな乗算器について，部分積加算部が L に比例するパターン数でテスト可能であることが初めて明らかになった．この結果から，Wallace 乗算器が $O(\log N)$ 個のパターンでテストできることが示された．

CSA の段数が 4-2 加算木より小さな乗算器について，C テスト可能であるかどうかは判明しておらず，更なる研究が必要である．

謝辞　本研究は科研費 (20300016) の助成を受けたものである．本研究を進めるにあたり御討論頂いた名古屋大学の高木一義准教授に心より感謝致します．

文　献

[1] C.S. Wallace, “A suggestion for a fast multiplier,” IEEE Trans. Electron. Comput., vol.EC-13, no.1, pp.14–17, Feb. 1964.

[2] J. Vuillemin, “A very fast multiplication algorithm for VLSI implementation,” Integration the VLSI J., vol.1, pp.39–52, 1983.

[3] J. Shen and F. Ferguson, “The design of easily testable VLSI array multipliers,” IEEE Trans. Comput., vol.33, no.6, pp.554–560, June 1984.

[4] A. Chatterjee and J. Abraham, “Test generation, design-for-testability and built-in self-test for arithmetic units based on graph labeling,” J. Electron. Test. Theory Appl., vol.2, pp.351–372, Nov. 1991.

[5] D. Gizopoulos, D. Nikolos, A. Paschalis, and C. Halatsis, “C-testable modified-booth multipliers,” J. Electron. Test. Theory Appl., vol.8, pp.241–260, June 1996.

[6] K.O. Boateng, H. Takahashi, and Y. Takamatsu,

"Design of C-testable modified-booth multipliers," IEICE Trans. Inf. & Syst., vol.E83-D, no.10, pp.1868–1878, Oct. 2000.
[7] B. Becker, "An easily testable optimal-time VLSI-multiplier," Acta Informatica, vol.24, pp.363–380, 1987.
[8] P. Zeng, Z. Mao, Y. Ye, and Y. Deng, "Test pattern generation for column compression multiplier," Proc. Seventh Asian Test Symposium, pp.500–503, 1998.
[9] 鬼頭信貴, 花井健輔, 高木直史, "4-2 加算木を用いたテスト容易な乗算器," 信学技報, VLD2006-140, ICD2006-231, March 2007.
[10] 鬼頭信貴, 高木直史, "種々の部分積加算構造をもつテスト容易な乗算器の設計手法," 信学論（D）, vol.J91-D, no.10, pp.2478–2486, Oct. 2008.
[11] D. Gizopoulos, M. Psarakis, A. Paschalis, and Y. Zorian, "Easily testable cellular carry lookahead adders," J. Electron. Test. Theory Appl., vol.19, pp.285–298, June 2003.

（平成 20 年 10 月 6 日受付，21 年 2 月 21 日再受付）

鬼頭　信貴　（学生員）

平 16 名大・工・電気電子情報工卒，平 18 同大大学院情報科学研究科博士前期課程了．現在，同大学院情報科学研究科博士後期課程在学中．テスト容易化設計，算術演算回路等の研究に従事．

高木　直史　（正員）

昭 56 京大・工・情報工卒，昭 58 同大大学院修士課程了，昭 63 京大工博．昭 59 京大・工・情報工助手，平 3 同助教授，平 6 名大・工・情報工助教授，平 10 同教授，平 15 名大・情報科学・教授．算術演算回路，ハードウェアアルゴリズム，論理設計等の研究に従事．平 7 日本 IBM 科学賞，情報処理学会坂井記念特別賞，平 17 科学技術分野の文部科学大臣表彰等受賞．