信号間の相互影響を考慮したバッファ挿入によるクロストークノイズ削減の一手法

定兼利行[†] 寺井正幸[††] 堀場康孝[†††]

本論文はクロストークノイズが生じた信号間の相互影響を考慮したバッファ挿入によるノイズ最適化問題を取り扱い，効率良い手法を提案する．従来の手法は victim 信号と aggressor 信号に対しそれぞれ個別にそれらのノイズの振幅を改善するのに対して，提案手法はタイミング制約の下で victim と aggressor 間の相互影響を同時に考慮したバッファ挿入による改善を行うために，独自の影響度関数 R を導入しこれを最適化するアルゴリズムを提案する．提案手法を実際の LSI 回路に適用しバッファ挿入位置と挿入数という点からその有効性を確認した．

Buffer Insertion for Crosstalk Noise Optimization

Toshiyuki Sadakane,[†] Masayuki Terai[††] and Yasutaka Horiba[†††]

Crosstalk noise avoidance is an increasingly critical phase in deep submicron LSI design. We propose an efficient buffer insertion method for noise and delay optimization. Our method applies buffer insertion to a victim net on which the noise is greater than the tolerable noise margin, so that the resulting noise pulses on the aggressor nets as well as those on the victim net may be less than the margin. Our experiments on the actual LSI circuits show that our method produces better results compared with the existing methods.

1. はじめに

LSI の配置配線後のレイアウト結果において，信号間の結合容量によって生じるクロストークノイズの電圧変動値（グリッチ（glitch）と呼ぶ）が最大許容値を超えた場合に，電気的誤動作が生じる．LSI 製造技術の微細化とともに配線寄生容量における配線間結合容量の占める割合が高くなり，そのグリッチのエラーの発生が顕著になっている．タイミング違反やクロストークノイズの改善手法として配線幅の調整（wire sizing）やバッファ挿入[1)~4)] とゲートサイジング[5)] が提案されている．このうち長い配線に起因するタイミング違反やノイズに対してはバッファ挿入が有効であると考えられる．Ginneken[3)] は Elmore 遅延モデル[6)] の下で挿入するバッファの最適な配置を見つけるダイナミックプログラミングのアルゴリズムを提案している．Lillis ら[4)] や Chen ら[2)] はこのアルゴリズムを拡張しバッファ挿入と配線幅調整をともに行うアルゴリズムを提案し，挿入バッファ数の削減を行った．Alpert ら[1)] はこれらのタイミング最適化を行うアルゴリズムにクロストークノイズの制約を課するという機能拡張を提案した．しかし，文献 1) の手法は後で示すようにバッファ挿入によって新たに許容値を超えるクロストークノイズが発生するという重大な欠点がある．

本論文は信号間の相互影響を考慮したバッファ挿入によるクロストークノイズ最適化の問題を取り扱う．**図 1** において上側の信号が aggressor（影響を与える側の信号）で下側の信号が victim（影響を受ける側の信号）である．2 つの信号間には結合容量が存在するので，aggressor の信号変化が victim 上のノイズのパルス電圧（グリッチ）を誘導する．このグリッチの振幅が与えられた最大許容値を超えている場合，図 1 (b) のように victim へバッファを挿入する．その結果できた 2 つの信号線に結合容量は分割され，挿入したバッファの入力端子と元の victim のシンク（sink）端子上に最大許容値以下の振幅のグリッチが現れる．こうしてクロストークノイズのエラーは解決される．着目す

† 株式会社ルネサステクノロジ製品技術本部
LSI Product Technology Unit, Renesas Technology Corporation
†† 大阪学院大学情報学部
Faculty of Informatics, Osaka Gakuin University
††† 関西大学大学院工学研究科
Graduate School of Engineering, Kansai University

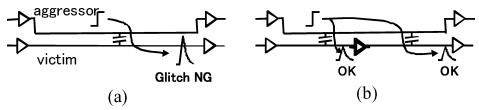


図 1 victim のクロストークノイズ：(a) 挿入前，(b) 挿入後
Fig. 1 Crosstalk noise of victim: (a) before buffering, (b) after buffering.

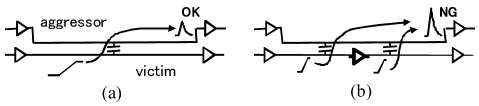


図 2 victim へのバッファ挿入の aggressor への影響：(a) 挿入前，(b) 挿入後
Fig. 2 Influence on aggressor from buffer insertion to victim: (a) before buffering, (b) after buffering.

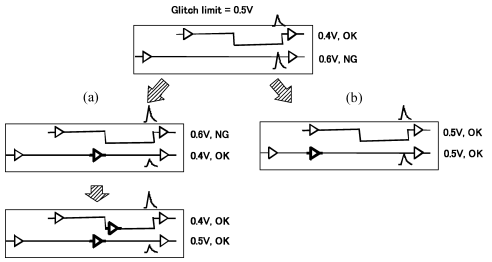


図 3 信号相互間の影響を考慮したバッファ挿入：(a) 考慮なし（従来手法），(b) 考慮あり（提案手法）
Fig. 3 Buffer insertion considering interaction between victim and aggressor: (a) without consideration, (b) with consideration.

る信号に許容値以上のクロストークノイズが発生した場合に，従来の手法[1] は victim と aggressor に対しそれぞれ個別にそれらのグリッチを改善する問題を扱うのに対して，本論文が扱う問題はその定式化が独自のものである．すなわち，提案手法はタイミング制約の下で 1 つの victim とその全 aggressor 信号群間の相互影響を同時に考慮したバッファ挿入によるグリッチの最適化を行う点が新規な点である．この信号間の相互影響を考慮したバッファ挿入を実現するために独自の影響度関数 R を導入し，これを最適化するアルゴリズムを提案する．目的関数 R としては，Devgan のグリッチ計算モデル[7] に基づく aggressor の全シンクノードの（victim の信号変化によって発生する）グリッチ評価値の総和としている．この目的関数の計算およびそのダイナミックプログラミングによる最適化手法の計算複雑度を問題の規模の多項式時間に抑える手法を考察した．

提案手法の利点はグリッチ違反修正の連鎖の発生を抑えることである．**図 2** (a) は victim の信号変化によって aggressor にもグリッチが誘導されることを示している．バッファ挿入後の victim の信号変化によって aggressor に許容値を超えるグリッチが現れることを図 2 (b) が示している．このグリッチを抑えるために，**図 3** (a) のように aggressor へのバッファ挿入を行う．図 1，図 2，図 3 に示すバッファ挿入によるグリッチエラーの連鎖，すなわち，victim のグリッチエラー発生，victim へバッファ挿入，その結果による新たに aggressor でグリッチエラー発生，aggressor へのバッファ挿入，…という連鎖を発生させることが従来手法の欠点であり，提案手法によりバッファがより適切な位置に挿入され，図 3 (b) のように連鎖の発生が抑えられ，挿入するバッファ数が減ることが期待できる．提案手法を実際の LSI 回路から抽出したクリティカルな小規模回路，500 K ゲートの大規模回路に適用し，その有効性を示す．

2. 問題の定式化

本章では信号伝播遅延とクロストークグリッチの計算モデルを説明した後，問題の定式化を行う．

2.1 準　　備

各信号ネットに対して配線を分岐点で分割しさらに指定長以下の線分に分割した後，**図 4** のような集中 RC 回路でモデル化する．

配線木をグラフ $T = (V, E)$ で表現する．ここで，ノードの集合 $V = \{v_0\} \cup SI \cup IN$ とし，E はノード間の $n-1$ 本 ($n = |V|$) の枝集合で枝 $e = (u, v) \in E$ は配線線分に対応し，信号はノード u からノード v へ伝播する．また v_0 は T のソースノード（根），SI は T のシンクノード（葉）の集合そして IN は T の中間ノードの集合とする．挿入すべきバッファの種類の集合を $B = \{b_1, b_2, \ldots, b_m\}$ とする．中間ノードの集合 IN の各ノードをバッファ挿入可能点とする．T へのバッファ挿入結果は写像 $P : IN \to B \cup \{b_0\}$ で表現できる．ここで b_0 はバッファ挿入なしを示す．

2.2 信号伝播遅延モデル

文献 1)，3) と同様に配線遅延は Elmore 遅延モデル[6] を用い，ゲート遅延はトランジスタをオン抵抗と見なす線形モデルを用いる．シンク $s \in SI$ に対して C_s を入力端子容量，$R_{\rm S}$ と $K_{\rm S}$ をそれぞれゲートのオン抵抗と真性遅延（intrinsic delay）とする．また C_e と R_e をそれぞれ枝 e の寄生容量と抵抗とする．$v \in V$ を根とする T の極大部分木を T_v と表すと T_v の負荷容量 $C(T_v)$ は以下の式で表される．

$$C(T_v) = \sum_{s \in SI(T_v)} C_s + \sum_{e \in E(T_v)} C_e \qquad (1)$$

ここで $SI(T_v)$ は部分木 T_v のシンクの集合，$E(T_v)$ は T_v の枝の集合である．枝 $e = (u, v)$ の Elmore モ

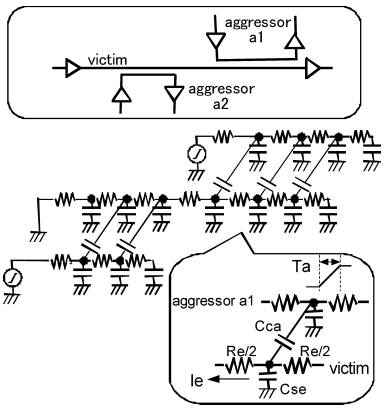


図 4 問題の定式化
Fig. 4 Problem formulation.

デルでの配線遅延は

$$DLY(e) = R_e(C_e/2 + C(T_v)) \quad (2)$$

となる．ソースノード v_0 のゲート遅延は

$$DLY(v_0) = K_S + R_S C(T_{v_0}) \quad (3)$$

と表せる．したがって T のソース v_0 を駆動するゲートの入力端子からシンク $s \in SI$ までの信号伝播遅延時間は，

$$DLY(v_0, s) = DLY(v_0) + \sum_{e \in path(v_0,s)} DLY(e) \quad (4)$$

と表せる．ここで $path(v_0, s)$ はノード v_0 からノード s へ向かう道を表す．各シンク $s \in SI$ には信号の要求到着時刻（required time）$Q(s)$ が与えられ，回路が正しく動作するためには $DLY(v_0, s) \leq Q(s)$ が成り立つことが必要である．この制約をタイミング制約と呼ぶ．

また，ノード v の要求到達時刻 $Q(v)$ を，$Q(v) = min_s\{Q(s) - DLY(v, s) | s \in SI\}$（$DLY(v, s)$ は v から s までの Elmore 遅延）と定義する．

2.3 クロストークグリッチモデル

クロストークグリッチの計算は，Devgan のグリッチモデル[7] を用いる．victim のグリッチは victim の配線抵抗，駆動ゲートのオン抵抗，aggressor との間の結合容量，aggressor の信号変化の遷移時間（slew time）に基づき計算する．図 4 (b) に示す victim の集中 RC 回路モデルを考える．victim の配線木 T 上の枝 $e \in E(T)$ の aggressor によって誘導される電流 I_e は式 (5) で示される．

$$I_e = C_e \sum_{a \in A} (C_{Ca}/C_{Se})(\mathrm{Vdd}/T_a) \quad (5)$$

ここで，A は枝 e と結合容量を持つ aggressor ノードの集合，C_{Ca} はノード a と victim 間の結合容量で，C_{Se} は枝 e に対する配線と基板との寄生容量（枝 e にバッファの入力端子がつながっている場合は端子容量と前記寄生容量の和），C_e は枝 e の寄生容量値で C_{Ca} と C_{Se} の和である．T_a は aggressor a の信号の遷移時間である．victim 上の枝 $e = (u, v) \in E(T)$ に誘導されたグリッチ $NS(e)$ は

$$NS(e) = R_e(I_e/2 + I(T_v)) \quad (6)$$

で与えられる．ここで $I(T_v)$ は部分木 T_v を流れる電流値（$\sum_{e \in E(T_v)} I_e$）を表す．

シンクノード $s \in SI$ におけるグリッチの振幅は，ノード v_0 から出発して s に至るパス上のすべての枝に誘導されるノイズの総和になるので，以下の式で表される．

$$NS(s) = R_S \sum_{e \in E(T)} I_e + \sum_{e \in path(v_0,s)} NS(e) \quad (7)$$

ここで R_S は victim の駆動ゲートのトランジスタのオン抵抗である．各シンクノード $s \in SI$ に対しノイズマージン $M(s)$ が与えられる．回路が正しく動作するためには $NS(s) \leq M(s)$ が成り立つことが必要である．この制約をグリッチ制約と呼ぶ．

ノード v のノイズマージン $M(v)$ を $M(v) = min_s\{M(s) - \sum_{e \in path(v,s)} NS(e)\}$ と定義する．また，各バッファ b_i の入力ピンについてもノイズマージン $M(b_i)$ が与えられており，バッファが挿入された場合には上流の配線木のシンクノードのノイズマージンとなる．

2.4 問題の定式化

本論文で扱うバッファ挿入問題と我々独自の影響度関数 R を定義する．

定義：victim とそれに影響を与える aggressor 群に対し，victim のあるノードにバッファを挿入したときの影響度関数 R は以下のように定義する．

$$R = \sum_{s \in G} NS(s) \quad (8)$$

ここで G は注目する victim に対するすべての aggressor ネットのすべてのシンクノード s の集合を表し，$NS(s)$ はシンクノード s でのグリッチで，式 (7) を aggressor ネットに対して適用することにより計算できる．

問題：バッファ挿入問題とは，victim とそれに影響を与える aggressor 群に対する以下の 5 つが与えられたとき，victim がタイミング制約とグリッチ制約を満足するという条件の下で影響度関数 R を最小化するような victim へのバッファ挿入の解 $P : IN \rightarrow B \cup \{b_0\}$ を算出せよという問題である．

- 集中 RC 回路である配線木 $T = (V, E)$
- 挿入バッファの種類の集合 $B = \{b_1, b_2, \ldots, b_m\}$
- aggressor の信号変化の遷移時間（slew time）
- ノイズマージン $M(s)$，$s \in SI$
- 信号の要求到着時刻 $Q(s)$，$s \in SI$

与えられた回路全体に対するバッファ挿入によるグリッチ違反改善は 3 つのステップ，すなわち，(1) グリッチ制約違反の度合いの最も大きい信号ネット n_k を選択，(2) n_k に対するバッファ挿入問題を解く，(3) 信号ネットのグリッチ更新をグリッチ違反ネットがなくなるまで繰り返すことにより行う．

3. 同時最適化手法

3.1 影響度関数 R

グリッチ $NS(s)$ は，式 (7) から分かるように $1/T_a$ に関する一次式なので以下のように変形できる．

$$NS(s) = \sum_{a \in A_s} X_a/T_a \tag{9}$$

ここで A_s はシンクノード s からソース v_0 までのパス上の配線に隣接するすべての aggressor ノード a の集合，X_a は以下のような多項式になる．

$$X_a = (R_{\rm S} + R(v_0, u)) \cdot C_e \cdot (C_{\rm Ca}/C_{\rm Se})\mathrm{Vdd}$$

$$R(v_0, u) = \sum_{e \in path(v_0,s) \cap path(v_0,u)} R_e + \sum_{e \in Q} R_e/2$$

$$Q = path(v_0, s) \cap \{(u, v)\}$$

ここでノード a と結合容量を持つ victim の枝を (u, v) としている．

これを式 (8) に代入すると影響度関数 R は，

$$R = \sum_{s \in G} \sum_{a \in A_s} X_a/T_a = \sum_{a \in A_{\rm A}} X_a/T_a. \tag{10}$$

ここで $A_{\rm A} = \{a \in A_s | \forall s \in G\}$，すなわち注目する victim に隣接する aggressor ネットに隣接するノードからなる集合であり，victim 自身のノードも含む．そこで集合 $A_{\rm A}$ を victim ノードの集合 $V(T)$ とそれ以外 $A_{\rm A} - V(T)$ に分類すると R は以下のようになる．

$$R = \sum_{a \in V(T)} X_a/T_a + R_0 \tag{11}$$

$$R_0 = \sum_{a \in A_{\rm A} - V(T)} X_a/T_a \tag{12}$$

X_a と R_0 は victim へのバッファ挿入の有無に依存しない定数であるので，後述のバッファ挿入アルゴリズムの実行前にあらかじめ計算しておくことができる．

ノード v における影響度関数 $R(v)$ を，T_v のノード集合 $V(T_v)$ を用いて以下で定義する．

```
/* 全体アルゴリズム */
Algorithm GDP(T, B)
Foreach node v ∈ V in topological order from sink to source
    If v is a sink node s
      /* シンクノード s の解を設定 */
      S (v) ← { (R0, M(s), 0, (Cs, Q(s))) }   /* S(v): ノード v の解候補集合 */
    Else
      /* 子ノードの解候補をマージして親ノードの解候補を作る */
      Let l(v) = left child node of v,   r(v) = right child node of v
      Compute S (v) ←BottomSolutions(v, S(l(v)), S(r(v) ))
    If v is not the source
      /* 各解候補を上流の枝の抵抗値を考慮して更新する */
      Compute S(v) ← TopSolutions(v, S(v))
    Else   /* v is the source */
      /* 最適解を選んで出力する */
      find the solution  ( r, m, i, (c,q)) ∈ S(v) with min. r and m≧0 and q≧0
      exit

/* 子ノードの解候補をマージして親ノードの解候補を作る */
Algorithm BottomSolutions(v, S(l(v)), S (r(v) ) )
S (v) ← Φ   /* S(v)はノード v の解候補集合 */
Foreach pair (rL, mL, iL, SimrL)∈S (l(v)), (rR, mR, iR, SimrR) ∈ S (r(v))
    /* r, m, i の更新(マージ) */
    r = rL+rR
    m = min(mL, mR)
    i = iL+iR
    /* v にバッファを挿入しない場合の解候補を作成 */
    Compute S' ← CombineSub( SimrL, SimrR)
    S (v) ← S (v) ∪ ( r, m, i, S')
    /* バッファ挿入する場合の解候補を作成 */
    Foreach buffer b ∈B
      Foreach (c, q) ∈S'
        If ( Rs(b)*i < m ) then   /* b を挿入してもノイズ制約を満たす */
          /* r,m,i,c,q を更新する */
          q' ← q - (Rs(b)*c-Ks(b))   /* バッファの遅延を差し引く */
          S (v) ←S (v) U {(r, M(b), 0, (C(b), q') ) }
    /* Property 1 を用いて不要な解を削除する */
    S (v) ← S (v) − {(r', m', i', (c',q')) ∈S (v) |
                          ∃(r, m, i, (c, q)) ∈S (v)
                            s.t. r≦r', c≦c', q≧q', i≦i', m≧m'}

/* 各解候補を上流の枝の抵抗値を考慮して更新する */
Algorithm TopSolutions(v, S(v))
S(v) ← Φ
Let e = upper edge of v
Foreach (r, m, i, Simr) ∈S (v)
    Foreach (c,q) ∈Simr
      /* r,m,i,c,q を更新する */
      r' ← r + Xv * Tv
      i' ← i + Ie
      m' ← m - Noise (e)
      q' ← q – c*Re
      c' ← c +Ce
      If ( m' ≧0 && q' ≧0 ) then   /*タイミング・ノイズ制約を満たす*/
        S(v) ← S (v) U { (r', m',i', (c', q')) }
    /* Property 1 を用いて不要な解を削除する */
    S(v) ← S(v) ¥ {{(r', m', i', (c',q')) ∈S(v) |
                          ∃(r, m, i, (c, q)) ∈S(v)
                            s.t.  r≦r', c≦c', q≧q', i≦i', m≧m'}

/* 子ノードの解候補をマージして c, q を更新 */
Algorithm CombineSub(Sl, Sr)
S ← Φ
Let Sl = Sl, indexed and ordered by c
Let Sr = Sr, indexed and ordered by c
i ← 1 ; j ← 1;
While( i ≦ |Sl| and j ≦ |Sr| )
      Let (cl, ql) = Sl[i]
      Let (cr, qr) = Sr[j]
      /* c, q の更新(マージ) */
      c' = cl + cr
      q' = min(ql, qr)
      S ← S ∪ { (c', q') }
      If  ( ql ≦ qr )   then  i ← i+1
      If  ( qr ≦ ql )   then  j ← j+1
```

図 5 アルゴリズム
Fig. 5 Algorithm.

$$R(v) = \sum_{a \in V(T_v)} X_a/T_a + R_0 \tag{13}$$

影響度関数 R はソースノード v_0 における影響度関数 $R(v_0)$ に等しい．

3.2 アルゴリズム

提案する手法は，文献 1) の手法の拡張である．アルゴリズムを**図 5** に示す．GDP が全体のアルゴリズムである．本アルゴリズムでは，victim の配線木 T において，全ノードをシンクからソースへむけて深さ優先

探索で順次訪問していく．各ノード v の訪問時に，部分木 T_v に対する解 P（$P : \{v\} \cup IN(v) \rightarrow B \cup \{b_0\}$，$IN(v)$ は T_v の中間ノードの集合）の（複数の）候補を計算し保存しておく．ここで各解候補 P について，ノード v における以下の 5 つのパラメータの値を合わせて保存しておく：影響度関数値 $r = R(v)$，ノイズマージン $m = M(v)$，ノイズ電流 $i = I(T')$，負荷容量 $c = C(T')$，要求到達時刻 $q = Q(v)$．ただし T' は T_v 中のバッファを挿入したノードから下流を削除した部分木を表す．

図 5 に示したように，まずシンクノードに対して自明の 1 つの解を設定する．そして各ノードの解候補に対して，ノードの上流側の枝の抵抗値を考慮して 5 つのパラメータを更新し（BottomSolution），次にその親ノードの訪問時に各子ノードの解候補をマージして親ノードの解候補を作成し，パラメータを更新する（TopSolution）．ソースノードに到達したら，解候補のうち R の最も小さい解候補を採用して出力する．

上記のアルゴリズムの枠組みは文献 1)，2)，4) 等のものと同じであるが，保存するパラメータに r を加えたことが文献 1) と異なる．

本アルゴリズムでは，文献 2)，4) と同じく各解候補に対応するパラメータ (r, m, i, c, q) を階層的なデータ構造に保持する．すなわち，同一の r，m，i を持つ解候補 (r, m, i, c, q)，(r, m, i, c', q') がある場合，(r, m, i, S_{imr})，$S_{imr} = \{(c', q'), (c, q)\}$ の形で保持する．また，TopSolutions および BottomSolutions において，同一ノードの解候補のうち，下述の性質 1 に示すような最適解となりえない解は保持しないで削除している．また BottomSolutions 内で呼び出している CombineSub 関数も最適解となりえない解を生成しないようにしている[1)]．

性質 1：あるノードに解候補 $S = (r, m, i, (c, q))$，$S' = (r', m', i', (c', q'))$ が存在し，

$$r' \leq r,\ c' \leq c,\ q' \geq q,\ i' \leq i,\ m' \geq m \quad (14)$$

ならば，解 S を削除しても最適解を失わない．

S と S' の間に式 (14) が成り立つと仮定する．その親ノードの解 $\overline{S}$ が解 S から BottomSolutions および TopSolutions によって作られた解であるとすると，解 S' から同じ方法で（BottomSolutions において同じ解とマージし同じバッファを挿入して）作られた解 $\overline{S}'$ と，$\overline{S}$ との間にも，式 (14) と同様の式が成り立つことが分かる．ここで S' から $\overline{S}'$ を作る際に解 S を使っていない．このことから，解 S から作られたソースノードのすべての解 $\underline{S}$ に対し，解 S' から S を使わずに作られたソースノードの解 $\underline{S}'$ が存在し，式 (14) と同様の式が成り立つ．よって S を削除しても最適解を失わないことが分かる．□

3.3 影響度関数 R の計算

X_a および R_0 は多項式時間で計算可能である．また提案するアルゴリズムにおいて R の計算処理として行っていることは，(1) シンクノード v における R 値 $R(v)$ として R_0 を代入，(2) 各ノードにおける関数 TopSolutions において式 $r = r + X_a/T_a$ で R 値を更新していることだけであり，それぞれ定数時間で計算可能である．

T_a は文献 2) に示された手法により計算する．各ノードの遷移時間はそのノードを駆動する上流のバッファの挿入位置に依存するため，各ノードの訪問時には決まらないので，いくつかの遷移時間に対する値を保持する方法をとる．図 5 では表現の簡単のために各ノード v の解候補の R_v 値は 1 値であるかのように記述しているが，実際には，いくつかの決められた遷移時間 T_v 値（たとえば t_0，t_1，t_2）に対する値 $R_{v,0}$，$R_{v,1}$，$R_{v,2}$ をテーブルとして保持する．v の上流ノード u の訪問時の R_u の計算では（TopSolutions），R_v のテーブルから R_u のテーブルを以下のように定数時間で計算する．枝 $e = (u, v)$ の伝達関数 H_e とノード v の駆動点アドミッタンス Y_v を計算し，$T_u = t_0$，t_1，t_2 それぞれに対する T_v の値 t'_0，t'_1，t'_2 を計算する．R_v のテーブルを補間することにより t'_0，t'_1，t'_2 に対する R_v 値の近似値を得る．これと $T_u = t_0$，t_1，t_2 を漸化式（$R_u = R_v + X_u/T_u$）に代入して R_u のテーブルを得る．ノード v にバッファを挿入した解候補を作る際（BottomSolutions）とソースノードに到達した際（GDP）は，駆動するバッファの R_S と Y_v から遷移時間 T_v を計算し 1 値の R_v を計算する．詳細は文献 2) を参照されたい．

4. 多項式時間の計算手法

文献 4) のバッファ挿入手法では各配線セグメントの容量値は数種類の離散値に限定（実際の容量値をあらかじめ設定された離散値に丸め込んだ値を用いる）しているが，解析精度の大きな低下がないことが示されている．これは配線の分割によって各セグメントは一定の長さ以下となりかつ均等な長さになっているためである．本手法でも同様に，各配線セグメントの容量値，抵抗値としてはあらかじめ設定された離散値を用いる．この離散値の数を c_{max} とし，またノイズの計算に用いる各ノードの遷移時間についても離散値に丸め込むこととし，この離散値の数を t_{max} とする．すると解候補のパラメータ c のとりうる値の数

は $O(n\ c_{max})$（$n = |V|$）となり[4]，また Devgan の見積式から I_e のとりうる値は $O(c_{max}^2\ t_{max})$，m および r のとりうる値は $O(n^2\ c_{max}^3\ t_{max})$ となる．したがって，アルゴリズム中で解候補の数は回路規模 n の多項式オーダで抑えられ，アルゴリズムの計算複雑度も多項式時間となる．

また，文献 1)，4) の手法と同様，本アルゴリズム中では最適解となりえない解を適宜削除しており，この処理により解候補の数は大幅に削減され，処理時間の増大をおさえることができる．

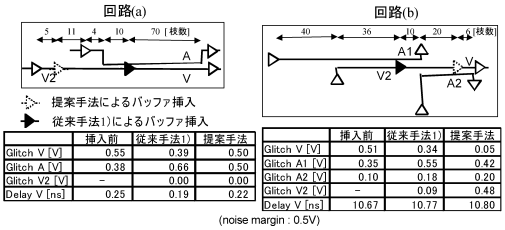
5. 評価実験

5.1 テスト回路での評価実験

提案手法の効果を示すため，人手で作成した回路で，提案手法と文献 1) の手法を実行して結果を比較した．使用した回路の形状を**図 6** に示す．図 6 中の配線長は配線木で表現したときの枝数である．配線木を構成する枝（配線線分）1 本あたりの平均の配線抵抗，対基板容量，結合容量はそれぞれ，23 Ω，3.3 fF，3.2 fF であり，回路中のバッファはすべて内部抵抗 $R_S = 1.1\,\mathrm{k\Omega}$，入力ピン容量は 3.5 fF である．電源電圧 $Vdd = 1.5\,\mathrm{V}$ でノイズマージンは全セルの入力ピンで 0.5 V に設定されている．タイミング制約はゆるく設定しており，すべてのケースでタイミング違反は発生していない．提案手法および従来手法 1) の実行結果のバッファ挿入位置と，バッファ挿入前後の各ネットのグリッチ，victim ネットの遅延値を図 6 に示す．(a) の回路では，バッファ挿入前は victim ネット V がグリッチ制約違反，aggressor ネット A はグリッチ許容内であるが，従来手法 1) により遅延最小のバッファ挿入をすると V のグリッチ制約違反は解消するが A がグリッチ制約違反を発生する．一方，提案手法を用いたバッファ挿入では V，A ともにグリッチ制約違反のない結果が得られる．回路 (b) でも同様の結果が得られている．提案手法では，従来手法 1) の場合と同じ数のバッファの挿入により，aggressor ネットのグリッチ制約違反を起こすことなく victim ネットのグリッチ制約違反を解消することができた．

5.2 実設計回路での評価実験

次に，実設計回路での提案手法の効果を示す実験を行った．使用したデータは，0.18 μm プロセス，6 層配線，Vdd=1.8 V，ランダムロジック部が約 500 KG 規模の実設計回路である．本データに提案手法を適用した結果，1,670 個のバッファが挿入されグリッチエラー数は 774 から 26 へ減少した．処理時間は 3 GHz の計算機上で 10 分であった．**図 7** は当該回路のレイアウト結果から提案手法によってバッファが挿入されたネットの 1 つを抜き出して示したものである．提案手法と従来手法 1) によるバッファ挿入前後についてのグリッチを図 7 に示す．従来手法 1) ではグリッチ制約違反が残っているが，提案手法ではグリッチ制約違反は解消されており，実設計回路でも提案手法が効果のあることを示している．

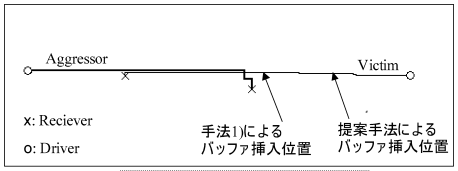


	挿入前	従来手法1)	提案手法
Glitch V [V]	0.55	0.39	0.50
Glitch A [V]	0.38	0.66	0.50
Glitch V2 [V]	–	0.00	0.00
Delay V [ns]	0.25	0.19	0.22

	挿入前	従来手法1)	提案手法
Glitch V [V]	0.51	0.34	0.05
Glitch A1 [V]	0.35	0.55	0.42
Glitch A2 [V]	0.10	0.18	0.20
Glitch V2 [V]	–	0.09	0.48
Delay V [ns]	10.67	10.77	10.80

(noise margin : 0.5V)

図 6 実験回路とバッファ挿入結果
Fig. 6 Experimental circuit and buffer insertion results.



total glitch [V]

	victim	aggressor
挿入前	0.51	0.41
提案手法	0.49	0.47
従来手法	0.46	0.54

(noise margin : 0.5V)

図 7 配線のレイアウトとバッファ挿入結果
Fig. 7 Layout of wiring and buffer insertion results.

5.3 標準ツールとの比較

業界標準の市販ツールと提案手法を実行して結果を比較した．実設計回路（510 KG，0.18 μm プロセス，Vdd=1.8 V）の配置配線結果（タイミング違反なし）に対して両者を実行した．提案手法は，(a) 3 章のアルゴリズムによるバッファ挿入位置の決定，(b) 市販ツールよるバッファの配置とバッファに接続する配線の結線，(c) 配線の RC 抽出，(d) 提案手法のモデル[7] によるクロストークノイズの計算の順に実行する．市販ツールはバッファ挿入だけでなく結合容量を減らす配線経路の修正も行う．クロストークノイズは提案手法と同じモデル[7] で計算した．結果を**表 1** に示す．提案手法のアルゴリズム（図 5 の GDP）はのべ 1,820 回適用され，回路上のすべてのグリッチ制約違反を解消した．ただし，バッファの配置と再配線の際に既配置のセルや既配線パターンのためにバッファ配置位置と配線経路が若干変わっているために，表 1 に示した最終の結果ではノイズ違反が 70 個発生している．本実験では比較を明瞭にするためにノイズマージンを

表 1 提案手法と市販ツールの実行結果
Table 1 Results of proposed method and commercial tool.

	ノイズ違反ノード数	バッファ挿入数	タイミングスラック [ns]	CPU時間 [分]
初期状態	958	-	0.077	-
提案手法適用後	70	1,820	0.097	169
市販ツール適用後	175	267	0.000	227

小さめの値 0.5 V に設定した．両者ともノイズの違反が残っているが，提案手法のほうがノイズの削減で上回っており処理時間も短い．提案手法適用後のタイミング違反は発生していない．市販ツールのほうがバッファ挿入数が少ないのは配線経路修正を併用するためと考えうるので，本実験ではバッファ挿入処理単体のバッファ挿入数の観点での性能の公平な比較はできないが，処理時間，クロストーク削減，タイミングの面では提案手法が優れていることが示された．

6. むすび

レイアウト結果において victim と aggressor 間の相互影響を同時に考慮したバッファ挿入によるクロストークノイズ改善問題を取り扱い，独自の影響度関数 R を導入しこれを最適化するアルゴリズムを提案した．その計算複雑度を問題規模の多項式時間に抑えるための関数 R の計算およびそのダイナミックプログラミングによる最適化手法を考察した．提案手法を実際の LSI 回路に適用しその有効性を確認した．

本論文の手法では簡単のため，Elmore の遅延モデルおよび Devgan のクロストークノイズのモデルを用いたが，これを Chen ら[2] のモーメントマッチングによる遅延とノイズの計算モデルのような高精度なモデルに置き換えることは容易であると考えられる．

参考文献

1) Alpert, C.J. and Devgan, A. and Quay, S.T.: Buffer Insertion for Noise and Delay Optimization, *DAC*, pp.362–367 (1998).
2) Chen, C.-P. and Menezes, N.: Noise-Aware Repeater Insertion and Wire-Sizing for On-Chip Interconnect Using Hierarchical Moment-Matching, *DAC*, pp.502–506 (1999).
3) van Ginneken, L.P.P.P.: Buffer placement in distributed RC-tree networks for minimal Elmore Delay, *ISCAS*, pp.865–868 (1990).
4) Lillis, J., Cheng, C.-K. and Lin, T.-T.Y.: Optimal Wire Sizing and Buffer Insertion for Low Power and a Generalized Delay Model, *J. of Solid-State Circuits*, Vol.31, No.3, pp.437–447 (1996).
5) 橋本ほか：ポストレイアウトトランジスタ寸法最適化によるクロストークノイズ削減手法，情報処理学会研究報告，No.SLDM103-6 (2001).
6) Elmore, W.C.: The transient response of damped linear network with particular regard to wideband amplifiers, *J. Applied Physics*, Vol.19, No.1, pp.55–63 (1948).
7) Devgan, A.: Efficient coupled noise estimation for on-chip interconnects, *ICCAD*, pp.147–151 (1997).

(平成 17 年 10 月 21 日受付)
(平成 18 年 4 月 4 日採録)

定兼 利行（正会員）
1990 年京都大学理学部数学専攻卒業．同年三菱電機（株）入社．2003 年（株）ルネサステクノロジへ転籍．製品技術本部設計技術統括部にて LSI 設計手法および EDA の開発に従事．

寺井 正幸（正会員）
1976 年大阪大学工学部電子工学科卒業．1978 年同大学大学院工学研究科博士前期課程修了．同年三菱電機（株）入社．2003 年（株）ルネサステクノロジへ転籍．2005 年大阪学院大学情報学部教授．LSI の CAD の研究に従事．工学博士．電子情報通信学会会員．

堀場 康孝
1964 年名古屋大学工学部電子工学科卒業．1966 年同大学大学院修士課程修了．同年三菱電機（株）入社．以来，バイポーラ/CMOS のアナログとディジタル LSI，ディジタル信号処理 LSI の研究開発に従事．2001 年関西大学工学部教授．工学博士．電子情報通信学会会員．