Guardian LVSによる論理ゲート認識

はじめに

本稿は、トランジスタ・レベル回路の一般的なCMOSゲートの認識について説明します。ルールに基づいた手法を使用した構造認識アルゴリズムは、論理ゲート識別[1, 2, 3, 4]において最も効率的です。このクラスのアルゴリズムは高速であり、インバータ、NAND、NOR、AOI、OAIゲートなどの静的な論理ゲートを容易に検出することが可能です。認識した後、変更したネットリストを論理ゲートおよび残りのトランジスタの観点から比較することが可能です。ネットリストの接続情報を示した注釈付きグラフは、トランジスタ・レベルのグラフに比べ、サイズが小さく、構造がより識別しやすくなります。これにより、グラフ同型問題の一例として扱われている回路比較問題[5]がより効率的に解決されます。

ゲート認識

Guardian LVSは、CMOS回路の論理ゲートをトランジスタ・レベルから認識します。ゲートを形成するトランジスタのグループは論理コンフィギュレーションによって表わされます。接続されたトランジスタのグループが論理ゲートを形成しない場合、Guardian　LVSは並列/直列に接続された同じタイプのトランジスタから論理コンフィギュレーションを生成します。これらが変換された後、論理コンフィギュレーションのレベルでLVS検証が実行されます。

論理ゲート認識を実行するには、Guardian LVSのユーザ・インタフェースで次の設定を行う必要があります。

・[プロジェクト設定]ダイアログの[モデル]ページでMOSトランジスタ用に[Logic gates]オプションを選択します。

・[プロジェクト設定]ダイアログの[ネット]ページで電源ネットおよびグラウンド・ネットを指定します。接尾辞「:P」、「:G」を使用してSPICEネットリストの電源ネットとグラウンド・ネットを指定することができます。この場合、[構文]ページの[コロン(：)以降のネット名の接尾辞を削除]オプションをオンにします。接尾辞はネット名のコロンの後ろから削除され、指定したネットは電源ネットおよびグラウンド・ネットとして解釈されます。

注記：インバータを認識するために [モデル]ページでMOSト

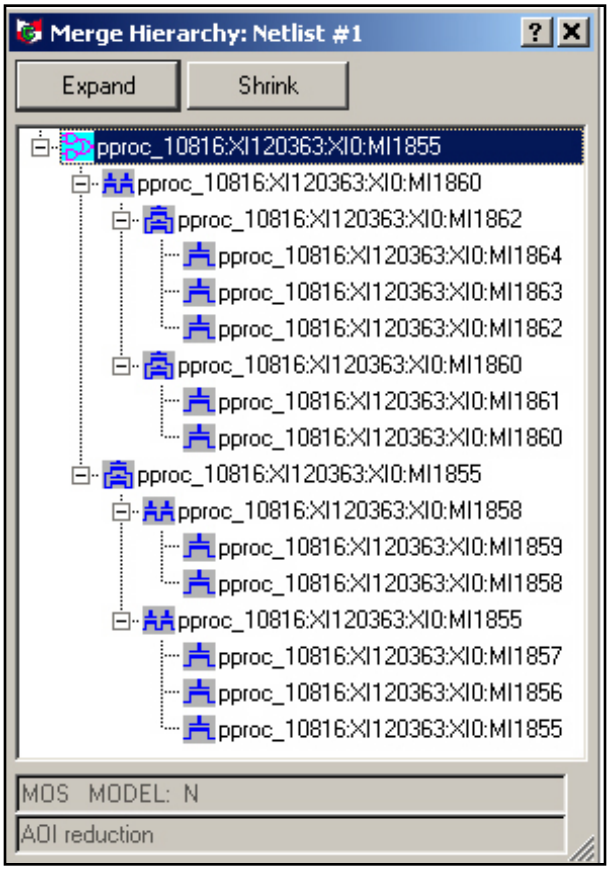

図1. AOIゲートの構造

ランジスタ用に[並列縮退]オプション、[直列縮退]オプションをオンにする必要はありません。ただし、より複雑な論理ゲートを認識する場合には、これらのオプションをオンにする必要があります。

論理ゲートは正しく構成されたMOSトランジスタに対してのみ生成されます。

・論理ゲートのすべてのトランジスタは、同じ端子数を持つ必要があります。

・論理ゲートのP型とN型コンポーネントは、縮退後も同じ素子数を持つ必要があります。

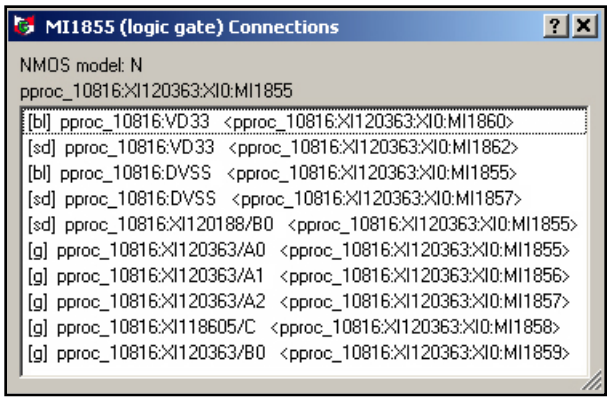

図 2. AOI ゲートの接続

・論理ゲートのP型とN型コンポーネントは、トランジスタのゲート端子に接続された異なるネットと同じ組み合わせを持つ必要があります。

・P型（N型）トランジスタのバルク端子は、１つの電源ネット（グラウンド・ネット）に接続される必要があります。

ユーザは、トランジスタのモデルをP型またはN型コンポーネントに構成するかどうかを[モデル]ページの[モデルのマッチング]オプションを使用して選択することができます。

Guardian LVSは論理ゲートの端子グループの論理等価性を検出するので、等価端子およびデバイスをトランジスタ・レベルで入れ替えることができます。たとえば、ユーザはNANDまたはNORゲートの入力端子を入れ替えることができます。

相違点およびマッチするノードがゲートレベルでレポートされ、確認ツールを使用して論理ゲートの構造または接続情報を調査することができます。たとえば、[マージ・ツリー]（図1）はAOIゲートがどのように生成されたかを示しています。

[接続先をすべて表示]は、論理ゲート接続情報、つまり、ゲートの端子に接続されたすべてのネットのリストを表示します（図2を参照)。

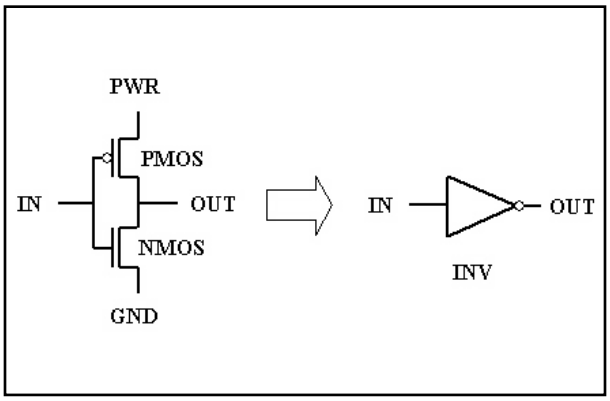

図 3. INV

認識される論理ゲートのタイプ

Guardian LVSは次の論理ゲートのタイプを認識します。

・INV: CMOSインバータ(図3)

正しく構成された（全項参照）P型およびN型トランジスタのペア(図3)が論理コンフィギュレーションINVに置き換えられます。

・NAND_N: N個の入力を持つCMOS NAND

並列/直列に接続されたPMOSおよびNMOSトランジスタで正しく構成された2つのグループをN個の入力1個の出力を持つNANDゲートに置き換えることができます。これらを縮退すると、縮退後のNANDゲートは入れ替え可能なN個の入力端子を持ちます。

NANDゲートを生成するには、[モデル]ページの[並列縮退]オプションおよび[直列縮退]オプションをオンにします。

・NOR_N: N個の入力を持つCMOS NOR（図5）

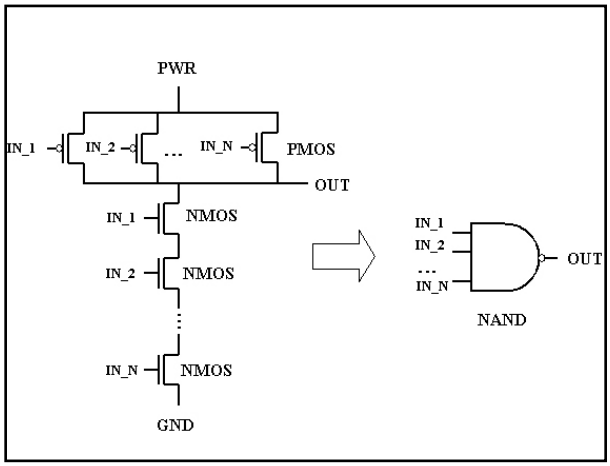

図 4. NAND_N

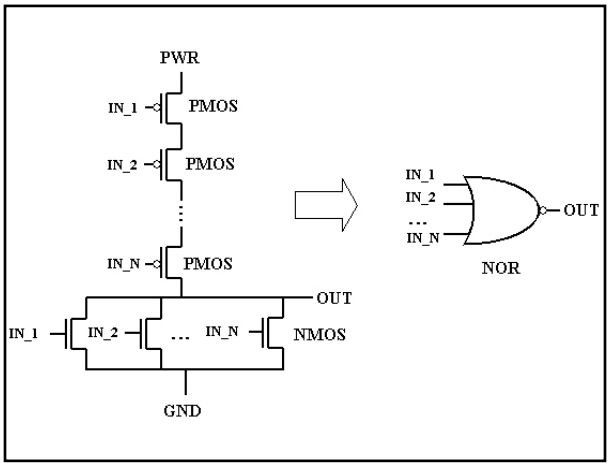

図 5. NOR_N

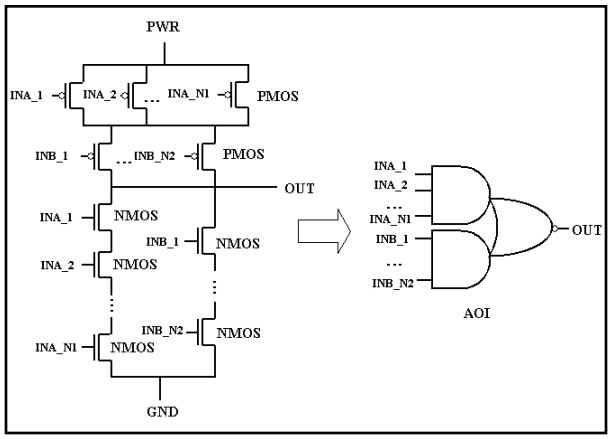

図 6. AOI_N1_N2

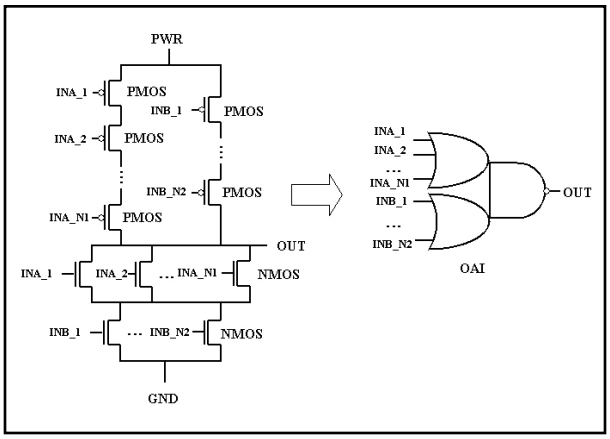

図 7. OAI_N1_N2

並列/直列に接続されたPMOSおよびNMOSトランジスタで正しく構成された2つのグループをN個の入力1個の出力を持つNORゲートに置き換えることができます。N個の入力端子はすべて入れ替え可能です。

NORゲートを生成するには、[モデル]ページの[並列縮退]オプションおよび[直列縮退]オプションをオンにします。

・AOI_N1…_NK:それぞれ　N1，　N2…NK入力を持つK個のAND素子およびOR-Invertブロックで表わされるCMOS AND-OR Invert論理コンフィギュレーション(図6)

AOIゲートの各ANDコンフィギュレーション内の入力端子は入れ替え可能です。たとえば、図6では入力端子INA_1、INA_2，…INA_N1が入れ替え可能で、入力端子INB_1，　…INB_N2は入れ替えられることができます。また、並列に接続されているPMOSトランジスタのグループが入れ替え可能です。したがって、INB_1, …, INB_N2ネットに接続された並列トランジスタはINA_1, INA_2, …, INA_N1に接続されたトランジスタのグループ上に配置することができます。

AOIゲートを生成するには、[モデル]ページの[並列縮退]オプションおよび[直列縮退]オプションをオンにします。

・OAI_NI_..._NK: それぞれN1，N2，…，NK入力を持つK個のOR素子およびAND-Invertブロックで表わされるCMOS OR-AND Invert論理コンフィギュレーション(図7)

OAIゲートの各ORコンフィギュレーション内の入力端子が入れ替え可能です。たとえば、図7では入力端子INA_1、INA_2, …INA_N1が入れ替え可能で、入力端子INB_1，　…INB_N2は入れ替えられることができます。また、並列に接続されているNMOSトランジスタのグループが入れ替え可能です。したがって、INB_1, …, INB_N2ネットに接続された並列トランジスタはINA_1, INA_2, …, INA_N1に接続されたトランジスタのグループ上に配置することができます。

OAIゲートを生成するには、[モデル]ページの[並列縮退]オプションおよび[直列縮退]オプションをオンにします。

まとめ

トランジスタ・レベル回路の論理ゲート認識を提案し、Guardian LVSで実現しました。この認識により、より有効なLVSネットリスト比較を実行することができます。

参考文献

[1] A.Lester, P.Bazargan-Sabet, A.Greiner, “YANGLE, a second generation functional abstractor for CMOS VLSI circuits,” *Proceedings of the10th International Conference on Microelectronics*, 1998, pp.265-268.

[2] M.Boehner, “LOGEX – an automatic logic extractor from transistor to gate level for CMOS technology,” *Proc. IEEE/ACM Design Automation Conference,* 1988, pp.517-522.

[3] L.Yang, C-J.R.Shi, “FROSTY: A program for fast extraction of high-level structural representation from circuit description for industrial CMOS circuits,” *Integration, the VLSI Journal,* V. 39, N 2, 2005.

[4] L.Yang, C-J.R.Shi, “FROSTY: A fast hierarchy extractor for industrial CMOS circuits,” *UWEE Technical Report,* N UWEETR-2003-0011, 2003.

[5] C.Ebeling, “Geminill: A Second Generation Layout Validation Program”, *IEEE International Conference on Computer-Aided Design,* 1988, pp. 322-325.