

AccuCellとAccuCoreにおける、ラッチおよびフリップ・フロップのモデリング

はじめに

本稿では、AccuCellとAccuCoreにおけるラッチおよびフリップ・フロップのファンクションを自動認識する方法について説明します。

概要

AccuCellとAccuCoreは、図1で示すようなラッチおよびフリップ・フロップ構造を自動的に認識します。その方法は、あらかじめ定義されたファンクション記述を含む内部テンプレートに対して単純にパターン・マッチングするのではなく、信号強度に基づいた、状態ベースのOBDD(Ordered Binary Decision Diagram)アルゴリズムを採用しています。このアルゴリズムの処理速度は、単にマッチングさせる方法より速く、また、RC素子を含むさまざまな論理回路のスタイルに対して、実際のファンクションを抽出することが可能です。

フリップ・フロップは、パターン・ファイルの一部として定義されていない限り、常に最初にラッチに分割されます。ただし、デフォルトで、分類(classification)中にマスタースレーブ型ラッチ・ペアを形成し、圧縮を図り、フリップ・フロップへ戻します。「フリップ・フロップ」として分類された構造は、エッジ・トリガー型デバイスとしてキャラクタライズされ、モデリングされます。

その他のタイプも、ラッチおよびフリップ・フロップ構造として認識される可能性はありますが、確実ではありません。クロックの方式が「パルス」または「グリッチ」スタイルのラッチやフリップ・フロップである場合、自動では認識されません。エッジ・トリガー型のフリップ・フロップであってもその様に分類されない場合は、KEEP_SUBCKTやKEEP_INSTコマンドを使用してサブサーキットを保存するか、またはFIND_SUBCKTコマンドを使用して形状(topological)パターン・マッチングを実行する必要があります。パターンが一致すると分類(classification)が行われます。

自動的にファンクションの抽出および分類ができない構造に関しては、手動または定義されたソリューションのライブラリから提供されるイクエーション(.eqn)ドリブン情報またはベクトル・テーブル・ファイル(.tbl)ドリブン情報を使用します。この件の詳細は、アプリケーション・ノート「AccuCellとAccuCoreにおける、ラッチおよびフリップ・フロップのユーザ定義による論理認識」を参照してください。またはシルバコの担当エンジニアにお問い合わせください。

ファンクションの自動抽出

ファンクションの自動抽出は、まず最初に、各ノードに対してプルアップ経路とプルダウン経路を高速にチェックすることで、弱いスタティック・ドライブか、強いスタティック・ドライブかを識別します。これにより、信号の向きが正確に識別できます。そして、パーティション/セルの各出力に対して、OBDD(Ordered Binary Decision Diagrams)で表現された4つのサブファンクションが抽出されます。ファンクションは、ネットリストを詳細に調べることで識別されたプライマリ入力および状態点(State-Point)によって表現されます。

DEBUG_BDDS -0: 包括的な情報を含むイクエーション・ファイルの表示をコントロールします。

PRINT_EQNS -0: ベーシックなイクエーション・ファイルの表示をコントロールします。

状態点(State-Point)の識別

順序タイプ論理回路、たとえばラッチ、フリップ・フロップ、レジスタおよびメモリセルは、「状態保持(State Holding)」回路の例です。この回路では、.f0(放電路"0")、.f1(充電路"1")、f0'(放電路がない"NOT 0")、f1'(充電路がない"NOT 1")で論理を完全に表すことができます。

次の2条件を満たす場合、この回路には状態保持が存在します。
(a) $(f0 \neq 0 \text{ OR } f1 \neq 0) \text{ AND } (f0' \text{ AND } f1')$ — すなわち、フローティング状態。

(b) 保持周期がクロック周波数よりも非常に大きい回路 — すなわち、実効ノード容量がユーザ指定のしきい値よりも大きい場合。

この時点で、各出力および内部状態に対して4つの最終的なイクエーションが生成されます。outputは、出力ピンの名前を表します。

output.0 — 出力が0の場合。(0 in PRINT_EQNS output)
output.1 — 出力が1の場合。(1 in PRINT_EQNS output)
output.z — 出力がハイ・インピーダンスの場合。(2 in DEBUG_BDDS output)
output.x — 出力が不定な場合。(3 in DEBUG_BDDS output)

最後に、この4方程式を解析し、分類を決定します。

方向性決定(および分割)の一部、およびある条件での最終的なファンクションの決定として、強いノードの認識に関するイクエーション(.4 と .5 in DEBUG_BDDS output)が生成されることもあります。

まとめ

ユーザ定義のパターン・マッチングではなく、アルゴリズム・アプローチを採用しているため、複雑なフリップ・フロップ、ラッチ、および大規模なフットレス/フット付ドミノ回路など、さまざまなセル・タイプのファンクション抽出が可能です。この結果、分割(AccuCoreのみ)ファンクション抽出やベクトル生成はすべて、共有のコマンド・セットで制御された反復プロセスに相互関係を持ちます。ひとつのフロー部分に現れたコマンドは、他の段階で付加的な役割を持つこともあり、ログ・ファイルに「out of sequence」と表示されることもあります。

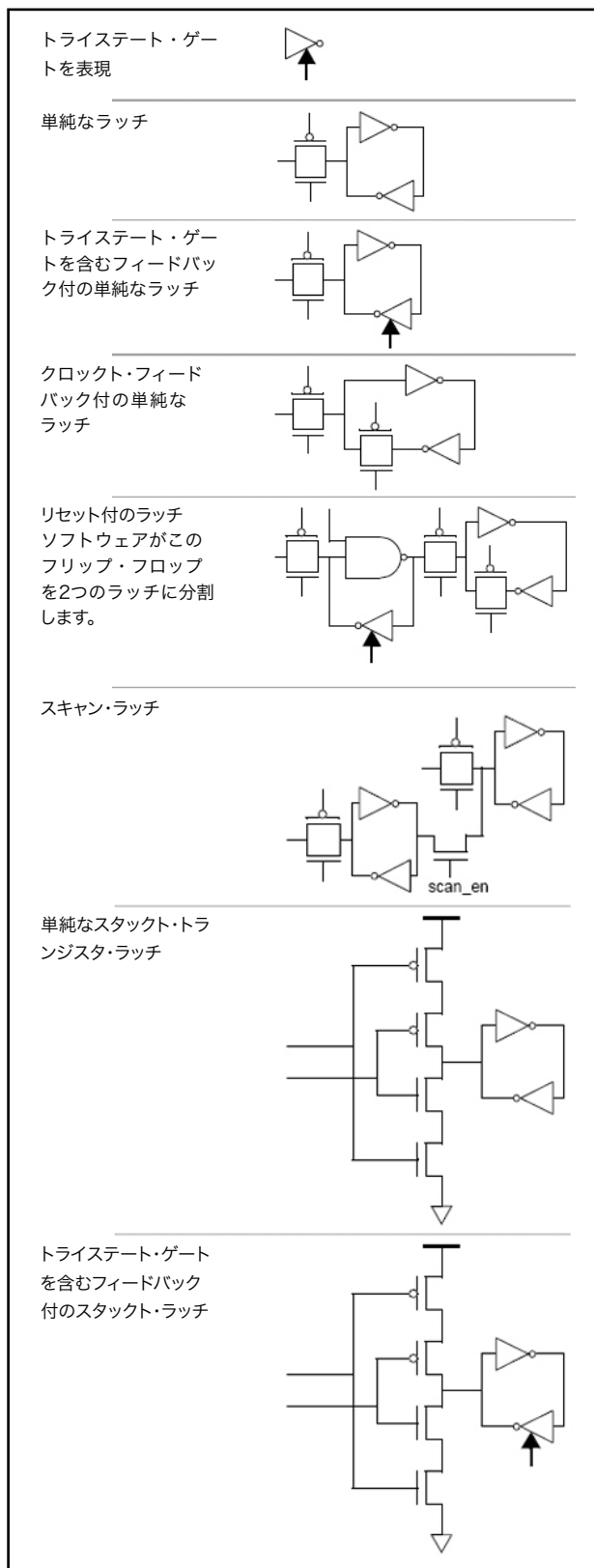


図1: サポート可能な構造