ビアスイッチFPGAにおけるスニークパス問題のSAT符号化を用いた検証

土井 龍太郎[1,2,a)]　橋本 昌宜[1,b)]

概要：従来 FPGA の性能ボトルネックである SRAM 型スイッチを，不揮発性メモリの一種であるビアスイッチで置換した FPGA に関する研究・開発が行われている．ビアスイッチ FPGA は縦と横に走る信号配線の交点にビアスイッチを配置したクロスバー回路により配線接続の切り換えを実現する．しかし，スイッチのプログラミングに共用の信号配線を使用するため，回路のプログラミング状態によっては，プログラミング信号が回り込んで意図しないスイッチに与えられるスニークパス問題が生じる．本稿では，FPGA の正常な再構成を阻害するスニークパス問題について，SAT 符号化を利用した数学的な検証に取り組む．回路動作などを論理式で表現し，SAT ソルバにより検証した結果，適切なオペレーション制約下での FPGA 運用によりスニークパス問題が回避できることがわかった．

キーワード：不揮発性ビアスイッチ FPGA，スニークパス問題，クロスバープログラミング，SAT 符号化，動作検証

1. 序論

近年，デバイスの微細化・高集積化に伴う ASIC (Application-Specific Integrated Circuit) 開発費の高騰や，電子機器製品サイクルの短縮による短納期，少量多品種生産，頻繁な仕様変更への対応要求等を背景として，FPGA (Field-Programmable Gate Array) に代表される再構成可能回路に対する期待が高まっている．しかし，FPGA には信号遅延，消費電力，回路面積が大きいという課題が存在する [1]．これらの課題は，再構成を実現するために，配線接続を切り換えるスイッチ回路が FPGA 上に大量搭載されていることに起因する．最も一般的な SRAM 型 FPGA では，スイッチ機能を担うトランスミッションゲートとスイッチのオン・オフ状態を保持する SRAM (Static Random Access Memory) セルによりスイッチ回路が構成される．しかし，MOS (Metal-Oxide-Semiconductor) トランジスタにより実現されるトランスミッションゲートは抵抗・容量が大きく，6 個のトランジスタを使用する SRAM セルは面積が大きい．そのため，従来のスイッチ回路の使用は配線性能や面積効率の低下を招く [2].

従来の FPGA が抱える課題の解消を目指し，性能ボトルネックである SRAM 型スイッチの代わりに抵抗変化型メモリである RRAM (Resistive RAM) をスイッチ回路として利用する FPGA の研究・開発が広く行われている [3–9]．しかし，これらの RRAM を用いた FPGA では，スイッチをプログラミングするため，スイッチ 1 個につき 1〜2 個のアクセストランジスタが付随する．スイッチのサイズと比較してアクセストランジスタのサイズが大きいため，スイッチ 1 個あたりのサイズがアクセストランジスタのサイズで決定されてしまい，回路面積の削減を阻む要因となっている．そこで，FPGA 応用向けに開発されている RRAM の一種である原子スイッチに，アクセストランジスタではなくバリスタを選択デバイスとして加えたビアスイッチと呼ばれる不揮発性デバイスの研究・開発が行われている [10].

ビアスイッチ FPGA は縦と横に走る配線の交点にビアスイッチを配置したクロスバー回路により配線接続の切り換えを実現する．しかし，スイッチのプログラミングに共用の信号配線を使用するため，回路のプログラミング状態によっては，プログラミング信号が回り込んで意図しないスイッチに与えられるスニークパス問題という現象が生じる．この現象は FPGA の利点である再構成を阻害する重大な問題であるため，発生条件等の把握が不可欠であるが，検証技術は現状未開発である．本稿では，スニークパス問題の厳密な検証を目指し，SAT 符号化を利用した数学的な

1 大阪大学 大学院情報科学研究科 情報システム工学専攻
2 日本学術振興会 特別研究員 DC
a) doi.ryutaro@ist.osaka-u.ac.jp
b) hasimoto@ist.osaka-u.ac.jp

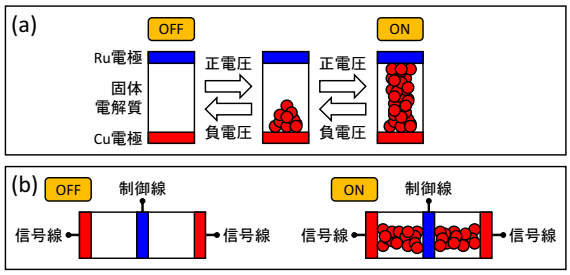

図 1　(a) 原子スイッチおよび (b)CAS の構造と動作

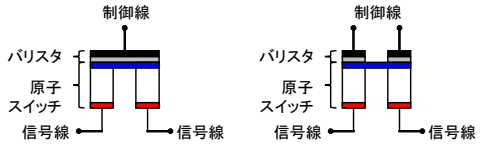

図 2　ビアスイッチの構造 (左: 1V-1CAS 構造, 右: 2V-1CAS 構造)

検証手法の確立に取り組む．回路動作やスニークパス問題の発生条件等を論理式で表現し，SAT ソルバに入力することでスニークパス問題を検証する．検証の結果，本検証手法により正しくスニークパス問題が発見できることを確認した．また，適切なオペレーション制約下においてビアスイッチ FPGA を運用すればスニークパス問題が発生せず，回路の再構成が正常に実行できることが分かった．

本稿の構成は次の通りである．2 節では本稿の検証対象であるビアスイッチおよびビアスイッチ FPGA の構造や動作を概説する．3 節において SAT 符号化を用いたスニークパス問題の検証手法を説明し，その検証結果について 4 節で議論する．最後に 5 節で結論を述べる．

2. ビアスイッチ FPGA の概要

2.1 ビアスイッチ

ビアスイッチは原子スイッチとバリスタにより構成される不揮発性ナノスイッチである [10]．まず各デバイスについて説明する．

原子スイッチは図 1(a) のように銅電極とルテニウム電極の間に固体電解質を挟んだ構造を有し，電極間に電圧を印加することで銅イオンによる架橋の形成・消失が可逆的に繰り返せる素子である．銅電極に正電圧を印加すると電極間に架橋が形成されてオン状態 (低抵抗状態) になり，負電圧を印加すると電極間の架橋が消失してオフ状態 (高抵抗状態) になる [8]．原子スイッチは，電源供給がなくてもオン・オフ状態が維持される不揮発性を有する．信頼性向上のため，図 1(b) のように 2 個の原子スイッチを逆方向に直列接続した CAS (Complementary Atom Switch) が単位スイッチ素子として用いられる．CAS のプログラミング時は信号線と制御線を使い，2 個の原子スイッチを 1 個ずつ書き換える．一方，通常動作時は信号線のみを使用する [9]．

図 2 のように CAS の制御端子部分にバリスタを接続したデバイスがビアスイッチである．信号端子と制御端子の間に閾値以上の電圧 (プログラミング電圧) が印加されるとバリスタが導通し，原子スイッチにプログラミング電流が流れる．一方，印加される電圧が閾値以下 (通常動作時) ならば，非導通のバリスタにより制御線が信号線と切り離され，ビアスイッチは信号端子のみからなる 2 端子素子とみなせる．このような印加電圧に応じて 2 つの機能を提供するバリスタを利用することで，FPGA 上に多数搭載されるビアスイッチの内，狙ったビアスイッチのみにプログラミング信号を与えることができる [10]．なお，ビアスイッチの構造として，バリスタと CAS を 1 個ずつ用いた 1V-1CAS 構造 (図 2 左) およびバリスタ 2 個と CAS1 個により構成される 2V-1CAS 構造 (図 2 右) が提案されており，両者でプログラミング方法が異なる [10]．ビアスイッチのプログラミング方法については後述する．

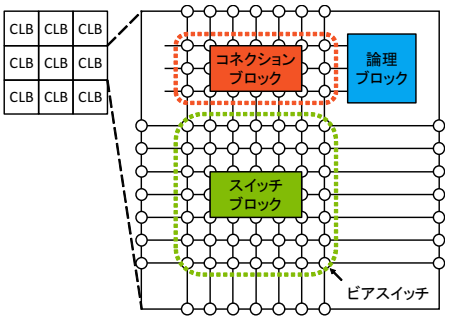

図 3　ビアスイッチ FPGA の構造

2V-1CAS 構造ビアスイッチについては，スイッチサイズが $18F^2$，オン抵抗が 400Ω，容量が 0.14fF と非常に小さい [10,11]．これらの特性により，SRAM 型スイッチをビアスイッチに置換することで FPGA の面積効率や性能が飛躍的に向上する．文献 [11] では，SRAM を用いた FPGA と比較してビアスイッチを用いたクロスバー回路の面積が 26 倍小さく，遅延およびエネルギーが共に 90%以上削減されることが報告されている．

2.2 ビアスイッチ FPGA におけるスニークパス問題

ビアスイッチ FPGA は図 3 のように CLB (Configurable Logic Block) を敷き詰めた構造を有し，各 CLB は縦横の配線の交点にビアスイッチを配置したクロスバー回路および論理ブロックからなる [11]．クロスバー回路では，縦横配線間および配線—論理ブロック間の接続をビアスイッチにより切り換える．論理ブロックでは，組み合わせ回路や順序回路を構築する．

次に，クロスバー回路におけるビアスイッチのプログラミング方法およびスニークパス問題について説明する．

まず，1V-1CAS 構造のクロスバー回路でのプログラミング方法を図 4 に示す．図 4 は，2x2 のクロスバー回路において，左下，左上の順にビアスイッチをオン状態にする流れをステップごとに示している．1V-1CAS 構造のクロ

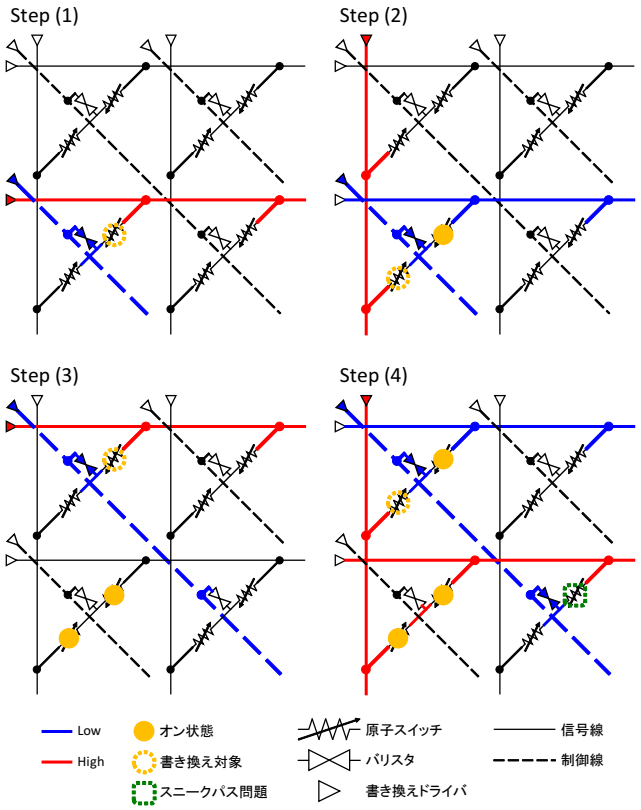

図 4　1V-1CAS 構造クロスバー回路におけるプログラミング

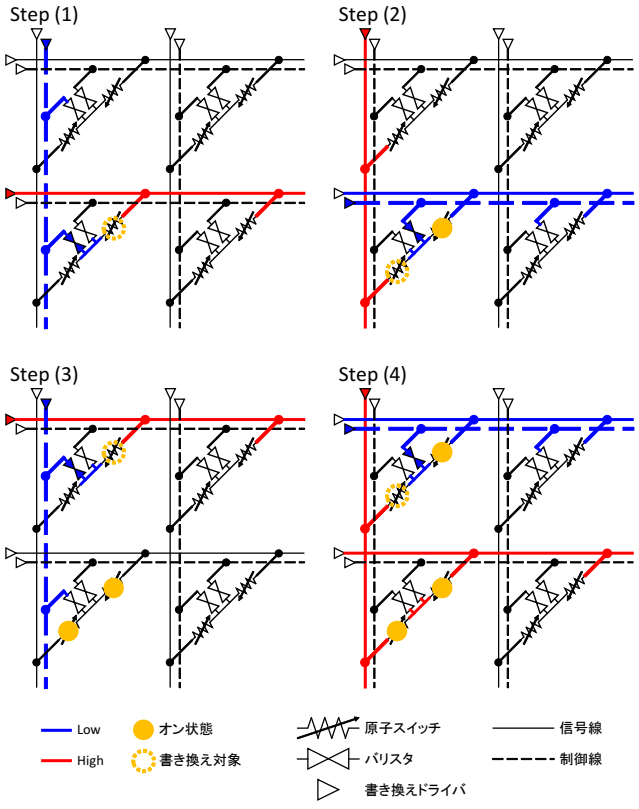

図 5　2V-1CAS 構造クロスバー回路におけるプログラミング

スバー回路では，斜め方向の制御線と縦横いずれかの信号線に書き換えドライバにより異電位を与え，原子スイッチに電圧を印加する．なお，それら以外の配線はフローティングにする．図 4 のステップ (1) と (2) により，正しく左下のビアスイッチ (2 個の原子スイッチ) をオン状態にできることが分かる．しかし，続く左上のビアスイッチのプログラミングは正常に実行できない．ステップ (4) において，ステップ (3) までにオン状態にした原子スイッチを経由してプログラミング信号が右下のビアスイッチに回り込み，意図しない原子スイッチに電圧が印加されるためである．このように，プログラミング信号の回り込みにより狙ったスイッチ以外の状態を書き換えてしまう現象をスニークパス問題と呼ぶ．

次に，2V-1CAS 構造のクロスバー回路におけるプログラミングの様子を図 5 に示す．こちらも，2x2 のクロスバー回路において，左下，左上の順にビアスイッチをオン状態にする流れをステップごとに示している．2V-1CAS 構造のクロスバー回路では，縦/横方向の制御線と横/縦方向の信号線を使用して原子スイッチにプログラミング電圧を印加する．なお，それら以外の配線はフローティングにする．2V-1CAS 構造では，左下と左上のビアスイッチを書き換える際にスニークパス問題は発生しない．しかし，ステップ (4) の次に右上や右下のビアスイッチをオン状態にしようとするとスニークパス問題が発生する．

スニークパス問題は FPGA の正常な再構成を阻害するため，その発生条件等の把握が不可欠である．以降では SAT 符号化を用いてスニークパス問題の検証を行う．

3. SAT 符号化を用いたスニークパス問題の検証

3.1 SAT の概要

SAT(充足可能性問題) とは，与えられた命題論理式の充足可能性を判定する問題である．ある論理式が与えられたときに，その式全体を真にする各命題変数の真偽値割り当てが存在するならば充足可能 (SAT) であるといい，存在しないならば充足不能 (UNSAT) であるという [12]．近年，SAT ソルバ性能の飛躍的向上に伴い，様々な分野への SAT の実用的応用が急速に加速している．SAT を利用した問題解法の一般的な流れは次の通りである．まず，原問題を論理式により表現 (SAT 符号化) する．次に，SAT ソルバを利用し，SAT 符号化された問題の解を求める．そして，その解を復号化することで原問題の解を得る．SAT を用いた問題解法では，原問題をどのように SAT 符号化するかが重要となる [12]．

3.2 スニークパス問題の検証方法

ビアスイッチ FPGA におけるスニークパス問題は，プログラミング信号の回り込みにより複数の原子スイッチのオン・オフ状態を一度に変更してしまう現象である．本稿では，この現象を SAT 符号化により論理式で表現し，SAT

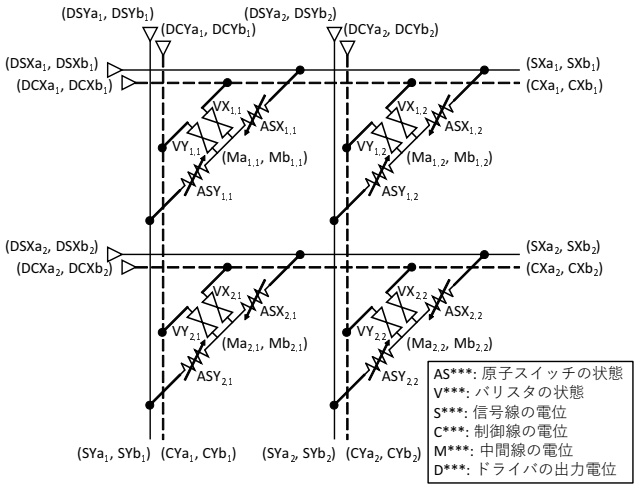

図 6 2V-1CAS 構造のクロスバー回路における変数定義

ソルバを用いて数学的に検証する．検証の流れは次の通りである．まず，スニークパス問題の発生条件を回路動作等と共に SAT 符号化する．詳細は後述するが，原子スイッチのオン・オフ状態や配線電位を命題変数で表し，それらを用いて原子スイッチのオン・オフ動作等の回路動作を論理式表現する．スニークパス問題の発生条件は “複数の原子スイッチのオン・オフ条件が同時に成立する” という命題で SAT 符号化する．次に，SAT 符号化により生成した論理式群を SAT ソルバに入力する．最後に，SAT ソルバの出力結果を解析する．ここでは，SAT/UNSAT の出力および SAT 時の各命題変数への真偽値割り当てを確認することにより，スニークパス問題の発生有無や，各原子スイッチのオン・オフ状態や配線電位がどのような状況の場合にスニークパス問題が起こるかを検証する．

3.3 スニークパス問題の SAT 符号化

ここでは，ビアスイッチ FPGA の回路動作やスニークパス問題の発生条件を SAT 符号化する方法について説明する．本稿では，基本的に 2V-1CAS 構造の 2x2 クロスバー回路を対象に説明するが，それ以外のサイズや 1V-1CAS 構造のクロスバー回路の場合でも同様に SAT 符号化できる．

まず，クロスバー回路の各構成要素の状態を表す論理変数を図 6 のように定義する．原子スイッチとバリスタはオン (導通) またはオフ (非導通) の 2 状態が存在するため，命題変数 1 個を使用して “1(真)” の場合をオン，“0(偽)” の場合をオフと定義する．配線電位と書き換えドライバの出力電位は Low, High, Hi-Z の 3 状態を取りうるため，命題変数 2 個の組 (a, b) を使用して (0, 0) の場合を Low，(0, 1) の場合を High，(1, 0) の場合を Hi-Z と定義する．

次に，定義した論理変数を用いてクロスバー回路の動作を論理式で表現する．例えば，原子スイッチのオン条件とオフ条件の論理式表現は式 (1) と (2) のようになる．式 (1) 全体が真になるのは同値 (⇔) の両辺が共に “0” または共に “1” のときである．右辺は (SXa_1, SXb_1) が (0, 1) かつ $(Ma_{1,1}, Mb_{1,1})$ が (0, 0) のとき，すなわち信号線 SX_1 に High の電位かつ中間線 $M_{1,1}$ に Low の電位が与えられたときのみ “1” となり，そのときのみ左辺も “1” となる．したがって，左辺の $ON_{ASX1,1}$ は原子スイッチ $ASX_{1,1}$ をオンにするような電圧が印加されているときに “1”，印加されていないときに “0” になる変数となる．同様に，式 (2) 中の $OFF_{ASX1,1}$ は原子スイッチ $ASX_{1,1}$ をオフにするような電圧が与えられているかを表す変数となる．

$$ON_{ASX1,1} \Leftrightarrow (\neg SXa_1 \ \wedge \ SXb_1) \wedge (\neg Ma_{1,1} \wedge \neg Mb_{1,1}) \quad (1)$$

$$OFF_{ASX1,1} \Leftrightarrow (\neg SXa_1 \ \wedge \ \neg SXb_1) \wedge (\neg Ma_{1,1} \wedge Mb_{1,1}) \quad (2)$$

スニークパス問題は，信号の回り込みによりクロスバー回路上の複数の原子スイッチに電圧が印加され，オン・オフ状態が一度に書き換わってしまう現象である．この現象は，式 (1) と (2) に示したような原子スイッチのオン・オフ条件を用いて “クロスバー回路上に存在する全ての原子スイッチのオン・オフ条件の内，2 個以上が真” という命題で SAT 符号化できる．この命題の論理式記述を SAT ソルバに入力した際，出力が SAT ならばスニークパス問題が発生することを意味し，そのときの各論理変数への真偽値割り当てを確認することでスニークパス問題発生時の各原子スイッチや配線電位の状況を取得できる．一方，SAT ソルバの出力が UNSAT ならばスニークパス問題が発生しないことを意味する．

本稿では，スニークパス問題の対策を検討するため，オペレーション制約を課した場合におけるスニークパス問題の検証も行う．文献 [10] によると，2V-1CAS 構造のクロスバー回路においては，いずれかの行 “および” 列でオン状態のビアスイッチが複数になるような，1V-1CAS 構造においては，いずれかの行 “または” 列でオン状態のビアスイッチが複数になるような書き換えオペレーションを行うときにスニークパス問題が発生する．しかし，これらの仮説は厳密に検証されていないため，次に示す 2 種類のオペレーション制約を導入して検証する．1 つ目は “列 OK 行 NG 制約: 同一行でオン状態のビアスイッチが複数個になるような書き換えオペレーションは禁止 (同一列は許可)”，2 つ目は “列 NG 行 NG 制約: 同一列および同一行でオン状態のビアスイッチが複数個になるような書き換えオペレーションは禁止” である．これらの制約の論理式記述を SAT ソルバの入力に追加することで，各制約を満たす範囲内でスニークパス問題が生じるかどうかを検証できる．

なお，前述した他にも以下に列挙する SAT 符号化が必要である．これらの SAT 符号化により，原子スイッチ，バ

リスタ，配線，ドライバというクロスバー回路の全構成要素のふるまいをモデル化でき，回路動作の論理式表現が得られる．

- バリスタの導通条件: バリスタへの印加電圧による導通・非導通をモデル化
- 配線電位の決定制約: ドライバにより信号線や制御線が駆動される動作をモデル化
- 書き換えドライバの制約: 動作させるドライバの数 (全部で 2 個) や位置 (縦，横，斜め方向) の条件
- 割り当て禁止の変数組の定義: 配線・ドライバ電位への変数組 (1, 1) 割り当ての禁止

信号線については電位決定制約が複雑になる．クロスバー回路では信号線同士がビアスイッチにより接続されており，各ビアスイッチのオン・オフ状態 (FPGA のコンフィギュレーション) に応じて信号線の電位決定制約が変化するためである．例えば，クロスバー回路上の全ビアスイッチがオフ状態であれば，各信号線は他の信号線と接続していないため，各信号線に対して独立に電位決定制約を定義できる．すなわち，各信号線に直接接続する書き換えドライバが動作していれば信号線はドライバ出力の電位に固定され，ドライバが非動作であれば信号線はフローティングになる．一方，ビアスイッチがオン状態になると，それに接続する 2 本の信号線については直接接続する書き換えドライバの動作状態のみでは電位が決定できない．2 本の信号線それぞれに直接接続する両方の書き換えドライバの動作状態を見て，その組み合わせに応じて配線電位を決定するように制約条件を定義する必要がある．両方のドライバが非動作であれば両信号線はフローティングとなり，一方でもドライバが動作していれば両方の信号線がドライバ出力電位に固定される．このように，信号線の電位決定制約を定義する際は FPGA のコンフィギュレーションによって，動作状態を見るべき書き換えドライバの数や位置が異なる．これに対応するため，現在の SAT 符号化では全通りのコンフィギュレーションに対して別個に電位決定制約を定義している．

紙面の都合上，バリスタの導通条件や動作ドライバの制約についての SAT 符号化の詳細は省略する．

4. スニークパス問題の検証結果

4.1 検証手法の有効性

まず，本稿で提案している SAT 符号化を用いた検証手法によりスニークパス問題が評価可能であるかを確認する．2V-1CAS 構造と 1V-1CAS 構造の 2x2 クロスバー回路それぞれについて，3.3 項で説明した全ての論理式を論理積で繋ぎ，SAT ソルバに入力することでスニークパス問題が発見できるかを検証した．なお，オペレーション制約は，この評価では使用していない．また，SAT ソルバはオープンソースの MiniSat [13] を使用した．検証の結果，

表 1 オペレーション制約のスニークパス問題への影響評価結果

	制約なし	列 OK 行 NG 制約下	列 NG 行 NG 制約下
2V-1CAS 構造	SAT	UNSAT	UNSAT
1V-1CAS 構造	SAT	SAT	UNSAT

2V-1CAS 構造，1V-1CAS 構造共に SAT ソルバの出力は SAT となった．そのときの各変数への真偽値割り当てを解析したところ，実際の回路動作に則した真偽値割り当てになっており，またスニークパス問題が発生する回路状況が得られていることを確認した．このことから，本検証手法によりスニークパス問題が発見できることを示せた．

ただし，現状では 2V-1CAS 構造で発見された全 256 通りの SAT 例および 1V-1CAS 構造で発見された全 456 通りの SAT 例の内の数例について正常動作を確認したのみであり，これら全ての SAT 例がスニークパス問題を表しているのか，またスニークパス問題全体の何%を発見できているのかを，今後確認する必要がある．

4.2 スニークパス問題の対策

ビアスイッチ FPGA において，“スニークパス問題を起こすビアスイッチに対する書き換えは実行しない” というオペレーション制約を設定できれば，その制約下で FPGA を運用する限りはスニークパス問題は発生しない．すなわち，狙ったビアスイッチのみを書き換えられることが保証できるため，回路の再構成が正常に行える．ここでは，そのようなオペレーション制約について検討するため，3.3 項で導入した列 OK 行 NG 制約や列 NG 行 NG 制約を課した場合に，スニークパス問題の検証結果がどのように変化するか評価する．

評価結果を表 1 にまとめる．表中の SAT はスニークパス問題が発生することを表し，UNSAT はスニークパス問題が発生しないことを表している．なお，評価自体は 2x2〜2x7, 3x2〜3x4, 4x2〜4x3, 5x2, 6x2, 7x2 の 14 通りのクロスバー回路サイズで行ったが，いずれの場合にも同様の SAT/UNSAT 出力が得られたため，表中では 1 つにまとめている．

表 1 より，2V-1CAS 構造のクロスバー回路では，制約なしの場合にスニークパス問題が発生し，いずれかのオペレーション制約を追加すればスニークパス問題の発生を防止できることが分かった．一方，1V-1CAS 構造のクロスバー回路においては，制約なし，および列 OK 行 NG 制約下ではスニークパス問題が発生する．SAT 例の総数を確認したところ，列 OK 行 NG 制約では，ある程度のスニークパス問題の発生個数の削減は可能だが，完全な回避はできていない．1V-1CAS 構造では，より厳しい列 NG 行 NG 制約を設定することでスニークパス問題を防止できる．

以上より，適切なオペレーション制約下においてビアス

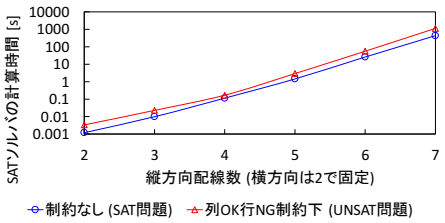

図 7 検証回路規模の増大に伴う SAT ソルバ計算時間の変化

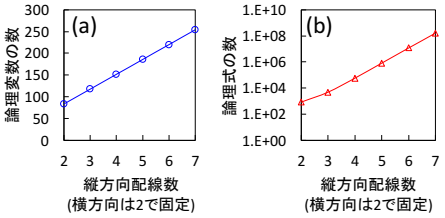

図 8 回路規模増大に伴う (a) 論理変数と (b) 論理式の総数の変化

イッチ FPGA を運用すればスニークパス問題は発生しないことが，本検証手法により確認できた．また，ビアスイッチの構造の違いによって，スニークパス問題を回避するためのオペレーション制約が異なることが分かった．

4.3 検証回路規模の増大による SAT ソルバ計算時間への影響

ここでは，検証するクロスバー回路サイズの増大が SAT ソルバ計算時間へ与える影響について議論する．

図 7 は，検証回路規模の増大に伴う SAT ソルバの計算時間の変化である．出力が SAT になるオペレーション制約なしでの 2V-1CAS 構造のクロスバー回路，および出力が UNSAT になる列 OK 行 NG 制約下での 2V-1CAS 構造のクロスバー回路の 2 通りで検証した．評価は CPU: Xeon X5680 3330MHz，RAM: 96GB の環境で実施し，各クロスバー回路サイズにおいて 10 回ずつ SAT ソルバを実行し，それらの計算時間の平均値をグラフにプロットしている．

今回評価した回路規模においては，計算時間の最大値は約 18 分であり，現実的な時間で検証できている．しかし，図 7 は片対数グラフであり，SAT 問題と UNSAT 問題のどちらも回路規模の増大に対して指数関数的に計算時間が増加していることを表している．この原因を探るため，検証回路規模の増加による論理変数の総数および論理式の総数への影響を評価した．結果を図 8 に示す．論理変数の数については，回路規模の増大に対して線形に増加する傾向が確認できる．一方，論理式数の変化を示した図 8(b) の縦軸は対数軸となっており，回路規模の増大に対して指数関数的に論理式数が増加することが分かる．2x7 という比較的小さいクロスバー回路においても，その動作検証を行うための論理式数は 164,705,202 個と非常に大きくなっている．この指数関数的な論理式数の増加が SAT ソルバ計算時間の増加の原因であると考えられる．

論理式数の内訳を調べたところ，3.3 項で述べた信号線の電位を決定する制約を記述した論理式数が回路規模の増大と共に指数関数的に増加することが確認された．3.3 項で述べたように，現在の SAT 符号化では信号線の電位決定制約を全通りの FPGA コンフィギュレーションに対して別個に記述している．クロスバー回路上に存在する原子スイッチの数を n とすると，各原子スイッチについてオン状態とオフ状態があるためコンフィギュレーションの組み合わせは 2^n 通りとなる．回路規模増大により n が線形に増加するため，信号配線の電位決定制約の論理式数は指数関数的に増加してしまう．

以上の議論より，検証回路規模の増大に伴う論理式数の指数関数的増加によりスニークパス問題検証に要する時間が増加し，大規模回路に対してスケーラブルでないことが示唆された．今後は，論理式数増加の支配的要因である信号線の電位決定制約の SAT 符号化を変更し，論理式数の増加を抑制することが必要になる．あるいは，数学的帰納法等で大規模回路の検証は小規模回路の検証と等価であることを証明し，大規模回路の検証を回避するというアプローチも考えられる．いずれにせよ，本検証手法のさらなる改良による大規模回路検証への対応が今後の課題となる．

5. 結論

本稿では，ビアスイッチ FPGA の再構成を阻害するスニークパス問題について，回路動作やスニークパス問題の発生条件等を SAT 符号化し，SAT ソルバを用いて検証した．その結果，本検証手法により正しくスニークパス問題が発見できること，適切なオペレーション制約下においてビアスイッチ FPGA を運用すればスニークパス問題が発生せず，回路の再構成が正常に実行できること等が分かった．今後の課題としては，本検証手法におけるスニークパス問題の発見能力を評価すること，大規模回路検証に対応できるように検証手法を改良すること等が挙げられる．

謝辞 本研究は，JST，CREST の支援，および JSPS 科研費 JP17J10008 の助成を受けたものである．

参考文献

[1] I. Kuon, et al., IEEE Trans. CAD, 2007.
[2] M. Lin, et al., IEEE Trans. CAD, 2007.
[3] P. E. Gaillardon, et al., IEEE Trans. NANO, 2013.
[4] S. Tanachutiwat, et al., IEEE Trans. VLSI, 2011.
[5] J. Cong, et al., IEEE Trans. VLSI, 2014.
[6] X. Tang, et al., FPT, 2014.
[7] Y. Y. Liauw, et al., ISSCC, 2012.
[8] K. Okamoto, et al., IEDM, 2011.
[9] M. Miyamura, et al., ISQED, 2014.
[10] N. Banno, et al., IEDM, 2015.
[11] J. Hotate, et al., FPL, 2016.
[12] 番原睦則, ほか, 学会誌 “情報処理”, 2016.
[13] “MiniSat,” http://minisat.se.