ハードウェアセキュリティ検査システムの開発
- RISC-V プロセッサ向けファザー -

1　背景

ソフトウェアのバグ・脆弱性とその影響は何十年も前から知られており、既にその検出と緩和のための様々な技術が確立されている。しかし、ハードウェアのバグ・脆弱性の脅威は、最近になってようやく重要視されるようになった。これは近年、Spectre や Meltdown 攻撃等のプロセッサのマイクロアーキテクチャに対する脆弱性が発見され、ハードウェアのバグ・脆弱性を利用することにより、ソフトウェアのセキュリティ保護を完全に突破できることが明らかになったからである（図 1）。

RISC-V はオープンソース ISA（命令セットアーキテクチャ）の一つで、OS やコンパイラをはじめとするソフトウェアエコシステムが着実に成長しており、多くの個人や組織が RISC-V ISA を実装するプロセッサを実装している。RISC-V にはライセンス費用や使用料がかからないため、従来より安価にプロセッサの開発ができ、今後 IoT、スマートフォン、自動運転車をはじめとする様々な分野で利用されていくことが予想される。このように手軽に RISC-V プロセッサの開発が行える時代になったが、プロセッサのバグ・脆弱性を発見するための技術やツールは不十分であり、プロセッサのバグ・脆弱性が問題となっている。

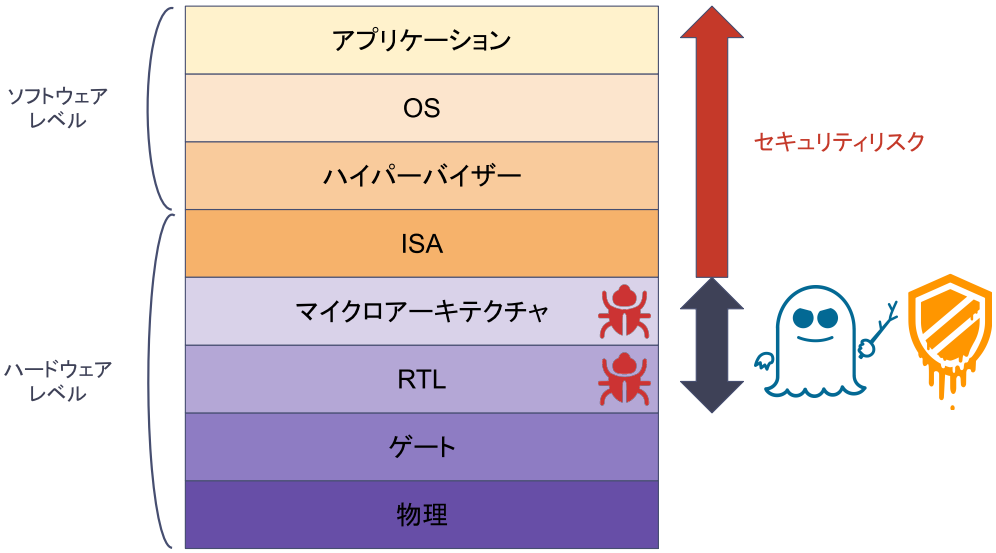

図 1: ハードウェア脆弱性

2　目的

本プロジェクトでは RISC-V プロセッサのバグ・脆弱性を自動で発見するためのファザーを開発し、既存のツールでは検出が難しいバグ・脆弱性を発見することを

目的とした。プロセッサの開発は図 2 に示すような工程で行われ、各段階でバグ・脆弱性が挿入される可能性がある。本プロジェクトでは HDL（Hardware Description Language）を用いて記述された RISC-V プロセッサの RTL（Register Transfer Level）をテスト対象とした。

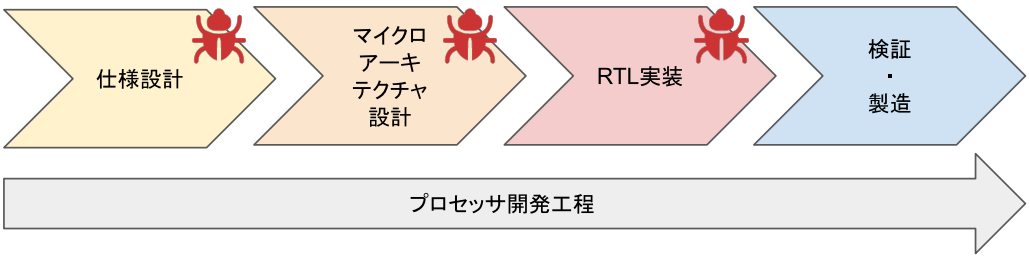

図 2: プロセッサ開発工程

3 開発の内容

本プロジェクトでは(1) RISC-V プロセッサを検査対象にし、(2) プロセッサ向けのカバレッジ情報と変異アルゴリズムを用いて、(3) 既存ツールに比べて効率よくバグ・脆弱性の発見が可能な RISC-V プロセッサ向けのファジングツールである MicroFuzz を開発した。

MicroFuzz の全体像は図 3 に示すように、ファザーが RISC-V のアセンブリファイルを作成し、コンパイラによって生成されたバイナリファイルを入力として、テスト対象となるプロセッサと命令セットシミュレータを実行する。実行時に得られたレジスタやメモリへのアクセス情報を比較し、バグ・脆弱性が発生していないかを検査する。テスト対象となる RISC-V プロセッサのバグ・脆弱性を効率よく検出するためにはファザーの性能が重要になる。

ファザーの全体像は図 4 に示すように、シード集合から取り出した命令列をプロセッサ向けの変異アルゴリズムを用いて変異させ、複数の命令列を作成する。その命令列を入力として、テスト対象となる RISC-V プロセッサを実行する。実行時に得られた情報をファザーにフィードバックし、実行に利用した命令列を評価する。評価はコードカバレッジや RISC-V ISA に基づく機能カバレッジ、さらにユーザが設定ファイルを通して指定したマイクロアーキテクチャ状態に基づいて行われ、重要な入力であればその入力をシード集合に追加する。これにより次回以降の変異のシードとしてその重要な命令列が使われるようになり、より重要な命令列を生成できる可能性が高まる。これらの一連の動作を繰り返し実行する中で、テスト対象となる RISC-V プロセッサのバグ・脆弱性の検出を行う。

MicroFuzz を用いてオープンソースで開発されており RISC-V プロセッサである Ibex と RSD を対象に、複数のバグ・脆弱性の検出に成功した。これは MicroFuzz が既存ツールでは発見できないバグ・脆弱性を検出できることを示している。実際に検出したバグ・脆弱性を GitHub Issue を通じて報告した。検出したバグ・脆弱性には特権命令に関するものが多かった。

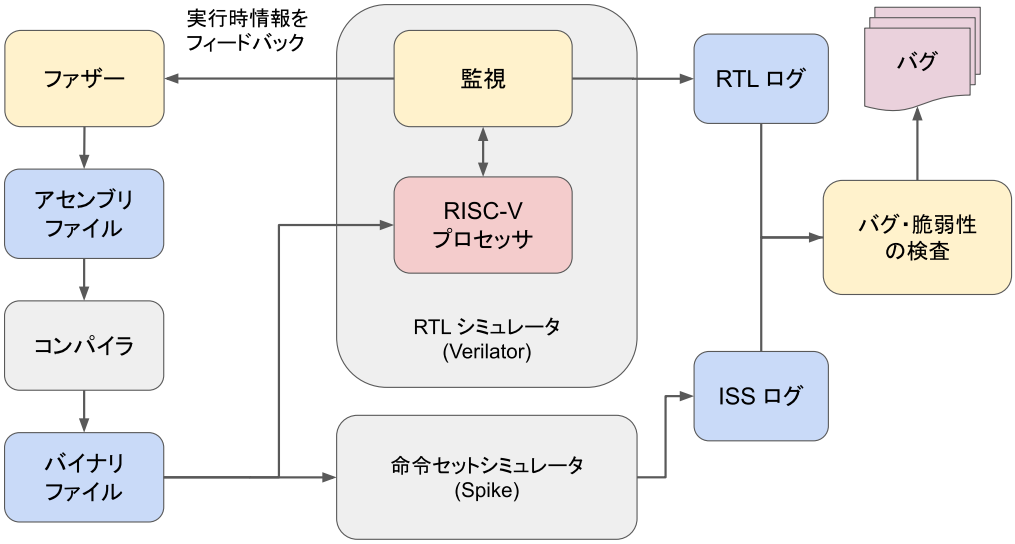

図 3: MicroFuzz の全体像

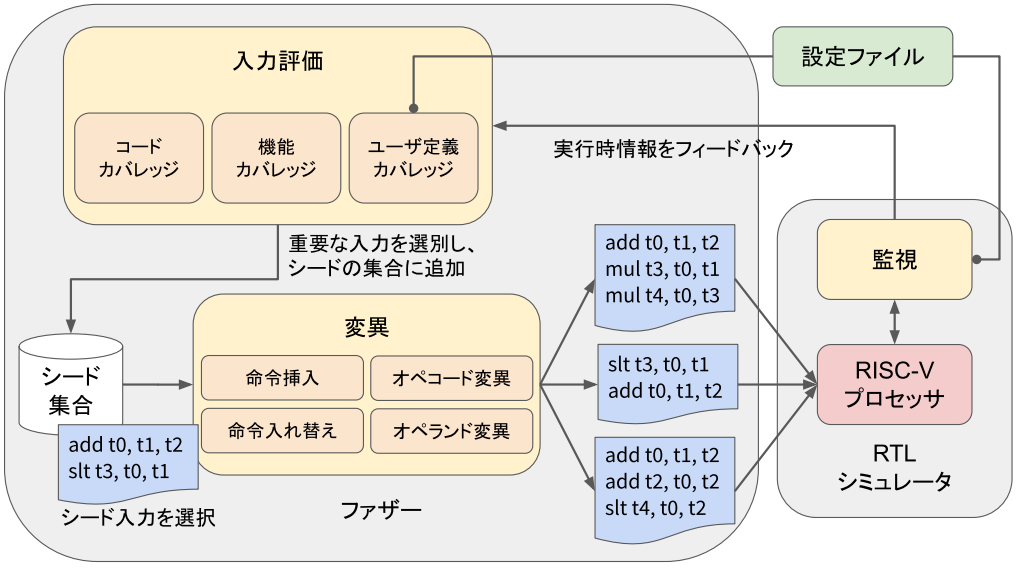

図 4: ファザーの全体像

4　従来の技術（または機能）との相違

本プロジェクトの特徴は、これまでプロセッサのテスト手法としてあまり注目されてこなかったファジングを用いて、RISC-V プロセッサ向けのファザーである MicroFuzz を開発し、従来の RISC-V プロセッサのテストツールに比べて効率よく RISC-V プロセッサのテストを可能にしたことである。従来の RISC-V テストツールである RISC-V Compliance は事前に用意されたユニットテストであるため、テスト対象となる各プロセッサのマイクロアーキテクチャ実装に合わせたテストが行えず、テストカバレッジが限られていた。MicroFuzz は実行時の情報を利用することで、テスト対象となる各プロセッサのマイクロアーキテクチャに合わせた命令列を生成し、高いテストカバレッジを達成できる。実際に既存のテストツールを用いて既にテストを行っている RISC-V プロセッサからも、MicroFuzz によって未発見のバグ・脆弱性を発見することができた。

5　期待される効果

本プロジェクトの成果により、既存ツールでは検出が難しかったバグ・脆弱性を効率よく検出可能になる。既存ツールより性能のよい RISC-V プロセッサのテストツールとして、RISC-V コミュニティに貢献し、よりセキュアなプロセッサの開発を可能にする。さらに本プロジェクトの成果はプロセッサに対するファジングの有用性を示すものであり、プロセッサ向けファザーの実用化や研究の活性化が期待される。

6　普及（または活用）の見通し

本プロジェクトの成果を論文として発表する予定である。最新のファジング手法を用いて、プロセッサのバグ・脆弱性を発見した初のプロジェクトであるため、学術的にも新規性がある。

7　クリエータ名（所属）

- 杉山 優一（東京大学大学院情報理工学研究科）

（参考） 関連 URL

- 検出したバグ：
  - https://github.com/lowRISC/ibex/issues/1277
  - https://github.com/lowRISC/ibex/issues/1282
  - https://github.com/rsd-devel/rsd/issues/37
  - https://github.com/rsd-devel/rsd/issues/38
  - https://github.com/rsd-devel/rsd/issues/39