



先進のRISC-Vプロセッサ検証手法

Shuzo Tanaka, eSOL TRINITY Co., Ltd.

Co-Author: Simon Davidmann & Lee Moore – Imperas Software

November 16-18, 2022

- **Imperas社紹介**
- RISC-V processor 設計検証 (DV:Design Verification)の課題
- 課題解決へのステップ
 - 1) Collaboration
 - 2) Standardization
 - 3) Methodology
 - 4) Verification IP
- Key technologies: reference models, verification IP
- Summary

Imperas会社紹介



- Imperasを設立したチームのメンバーは、EDAツール、FPGAおよびプロセッサIPに関する技術バックグラウンドを持っています。
- ビジョン
 - EDAツールのようなシミュレーション手法が、今後のソフトウェア開発、テスト等で必須になる。
 - Imperas製品は、ツールへの技術要求から製品アーキテクチャを検討し、開発されています。モデリング要件からのボトムアップで開発されたツールではありません。
- Imperas社は、RISC-Vの普及により、新たにRTLの設計検証環境の技術および手法をポートフォリオに加えました。
- 本日の講演
 - RISC-V対応ソフトウェアシミュレーションおよびRTL設計検証環境についてお話しします。

Imperas社のRISC-Vへの取り組み



- Q2 2016: DAC – 最初のRISC-Vとの出会い – まだアカデミックな状況と認識
- Q4 2016: RISC-V Workshop (@ Google) – 350社が参加、ISAは収束しつつある状況
- Q1 2017: ImperasがRISC-V Foundationに参加; 最初のRISC-V processor modelを開発
- Q3 2017: ImperasがCompliance Working Groupに参加; ISSの構築・寄贈とテストを実施
- Q1 2018: ImperasがRISC-Vコアへの命令追加、最適化の手法を導入
- Q2 2018: ソフトウェア開発および設計検証用途にImperasのRISC-V modelsを初めて購入して頂いた
- Q1 2019: ImperasのRISC-Vモデルを設計検証モデルとして使用したRISC-V SoCの最初のテープアウト
- Q2 2019: ImperasとGoogleが、命令ストリームジェネレータを使用したDVフローで協業を開始
- Q1 2020: Imperasは、OpenHW Groupおよび各メンバーとCore-VコアのDVに関する研究を開始
- Q1 2021: Imperasは、DVConシリコンバレーカンファレンスにて、RISC-VのDVに関する論文を2本発表 (OpenHW、Nvidia Networkingと共同)
- Q4 2021: Imperasは、ImperasDV RISC-V向け検証製品を製品ラインナップとして発表
- Q1 2022: Imperas社は、RVVI (RISC-V Verification Interface) をRISC-V DVコミュニティのオープンスタンダードとしてGitHubで公開しました

Imperas社のRISC-V関連のお客様 およびパートナー



The most complex RISC-V processor projects use Imperas

Users

- Nvidia Networking (Mellanox)
- **NSITEXE (DENSO subsidiary)**
- NXP
- Silicon Labs
- Nagravision
- Dolphin Design
- lowRISC (Ibex)
- EM Micro US
- Top 10 semiconductor company with embedded, GPU use cases
- Top-tier systems company (AI application)
- Largest automotive ADAS/AI company
- Startup building accelerator based on multiprocessor RV64
- **Japanese government projects "TRASIO" and "RVSPF"**
- Numerous universities around the world
- 100+ organizations using free riscvOVPsimPlus

Partners

- RISC-V Intl
- OpenHW Groupe
- CHIPS Alliance
- Google (Open source ISG)
- Valtrix (test generation tools)
- Andes (processor IP vendor)
- SiFive (processor IP vendor)
- Cudasip (processor IP vendor)
- MIPS (processor IP vendor)
- Microchip (Microsemi FPGA Mi-V processor IP)
- Intel FPGA (Nios-V processor IP)
- Intel (RISC-V Pathfinder IDE)

- Imperas社紹介
- **RISC-V processor 設計検証 (DV:Design Verification)の課題**
- 課題解決へのステップ
 - 1) Collaboration
 - 2) Standardization
 - 3) Methodology
 - 4) Verification IP
- Key technologies: reference models, verification IP
- Summary

RISC-V Processor設計検証の課題



- IPの機能選択は、設計および実装に大きく影響します。
 - 開発者は検証への影響を考慮する必要があります。
- 現在のSoC開発に掛かるコストの50%はハードウェアDVです (実績のあるCPU IPを購入した場合)
 - 自社でCPUを開発する場合には、スケジュール、リソース、品質等のより多くの課題を解決する必要があります。
- プロセッサDVは多くのチームにとって初めての経験となります。
- 既存のDVメソッドロジはSoCとSoCサブシステムのDVにフォーカスしており、プロセッサDVの課題に完全に対処していません。
- 2021年の時点でプロセッサDVツールは、まだ製品として販売されていませんでした。

議題



- Imperas社紹介
- RISC-V processor 設計検証 (DV:Design Verification)の課題
- **課題解決へのステップ**
 - 1) **Collaboration (協業)**
 - 2) Standardization
 - 3) Methodology
 - 4) Verification IP
- Key technologies: reference models, verification IP
- Summary

Collaboration



- オープンソース、非営利団体、および産業界との協力によるユニークな融合
- riscv-dv: ランダム命令ストリームジェネレータ (Google/CHIPS Alliance)
- core-v-verif: UVMベースのRISC-Vプロセッサ向け検証環境 (OpenHW Group)
- RVVI: RISC-Vプロセッサ検証のためのオープンスタンダードなインタフェースとAPI (OpenHW Group, 産業用ユーザ)

議題



- Imperas社紹介
- RISC-V processor 設計検証 (DV:Design Verification)の課題
- **課題解決へのステップ**
 - 1) Collaboration
 - 2) **Standardization (標準化)**
 - 3) Methodology
 - 4) Verification IP
- Key technologies: reference models, verification IP
- Summary

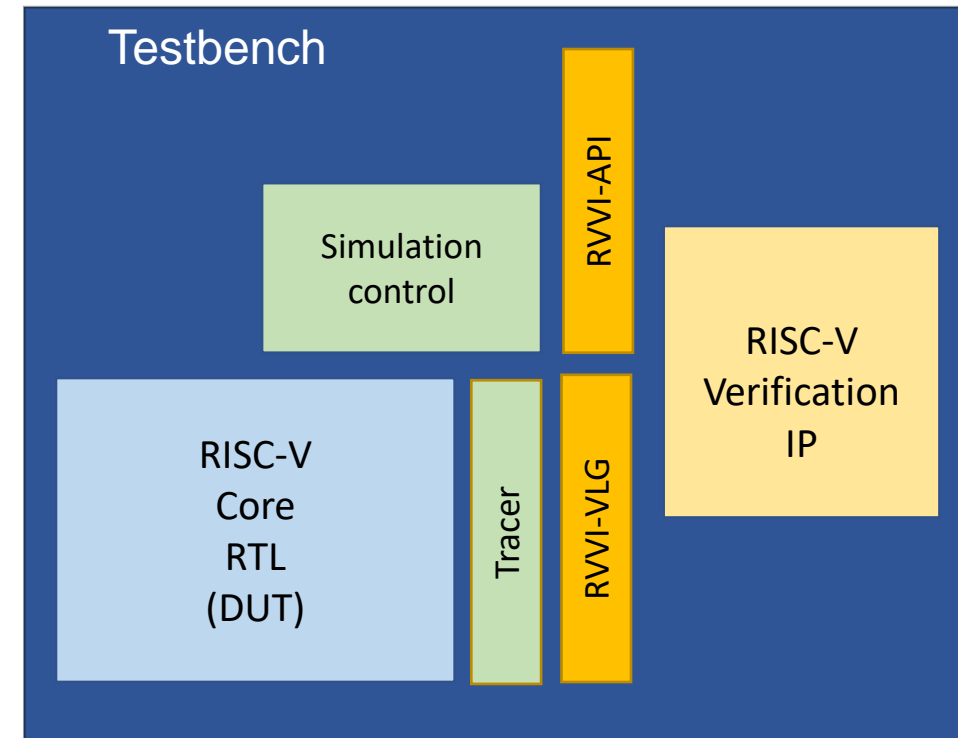
Standardization



- 標準化は、IPユーザーとIPベンダーの両方にとって良いことです。
- IPユーザーにとって良いこと:
 - ベストプラクティスをサポート
 - 再利用とポータビリティ
 - 迅速な立上げと実行
- IPベンダーにとって良いこと:
 - IPとのインターフェースに多くのコンフィギュレーションや多様な方法を提供する必要がない
← 高い品質に繋がる
 - 容易な顧客サポート
- 標準UVM(Universal Verification Methodology)は良いサンプルとなります。
 - UVM準拠のIPは統合し易く、新人エンジニアでもすぐにテストベンチの作成ができます。

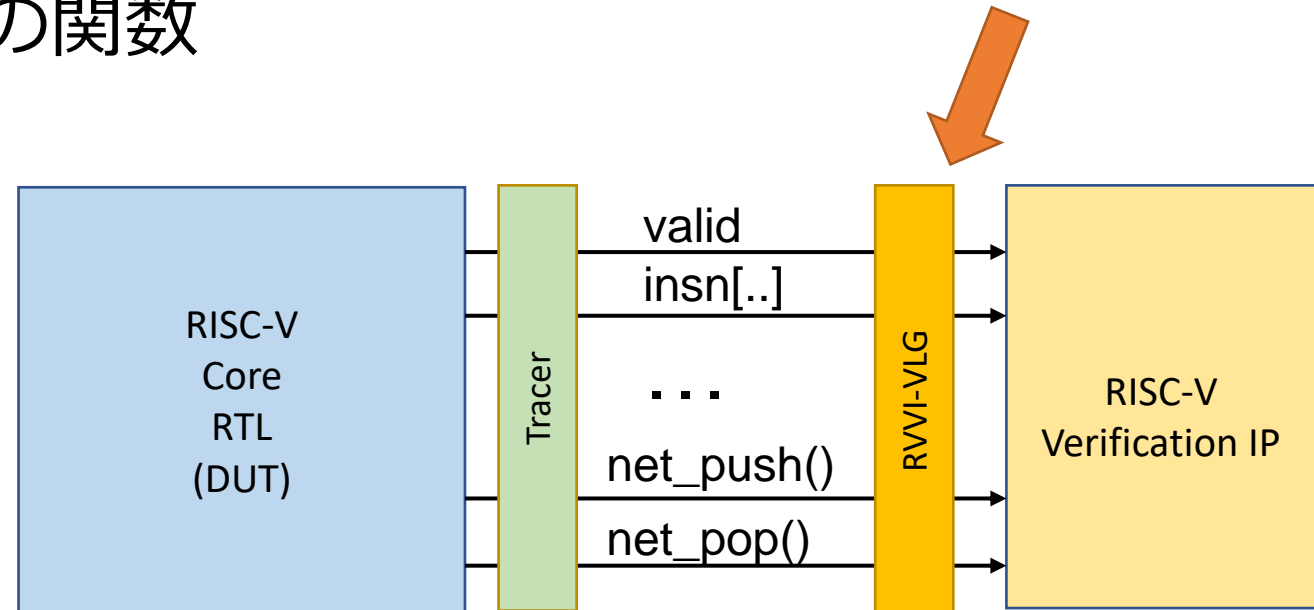
Standardization: RVVI

- RVVI = **R**ISC-**V** **V**erification **I**nterface
 - <https://github.com/riscv-verification/RVVI>
- 2年越しの成果
 - Imperas, EM Micro, NSITEXE, OpenHW, Silicon Labs, ...
- テストベンチとRISC-V検証用IPの通信を標準化
- 2つのパート:
 - **RVVI-VLG**: RISC-V VIPとの信号レベルインタフェース
 - **RVVI-API**: RISC-V VIPとの機能レベルインタフェース



RVVI-VLG

- トレーサが抽出する情報の定義
- SystemVerilogのインタフェース
- 非同期イベントを処理するための関数
 - 例えば、割込み、デバッグ要求等
- <https://github.com/riscv-verification/RVVI/tree/main/RVVI-VLG>



RVVI-API

RVVI-API



rvviRefEventStep()
rvviRefGprsCompare()
rvviRefPcCompare()
rvviRefCsrsCompare()
⋮
rvviRefGprGet()
rvviRefPcGet()
rvviRefInsBinGet()
rvviRefCsrGet()



- RISC-VプロセッサのVIPが実装すべき標準機能
- ステップ・アンド・コンペア・メソドロジーをサポート
- C言語版とSystemVerilog版を用意
- <https://github.com/riscv-verification/RVVI/blob/main/include/host/rvvi/rvvi-api.h>

議題



- Imperas社紹介
- RISC-V processor 設計検証 (DV:Design Verification)の課題
- **課題解決へのステップ**
 - 1) Collaboration
 - 2) Standardization
 - 3) Methodology (方法)**
 - 4) Verification IP
- Key technologies: reference models, verification IP
- Summary

Methodology



- 既知の問題に対処するために設計されたベストプラクティス
- 成功例と失敗例の積み重ね
- ImperasはRISC-VプロセッサのDV手法を定義しています。
 - 顧客やエコシステム・パートナーとの協働
- 成果: 非同期ステップアンドコンペア手法

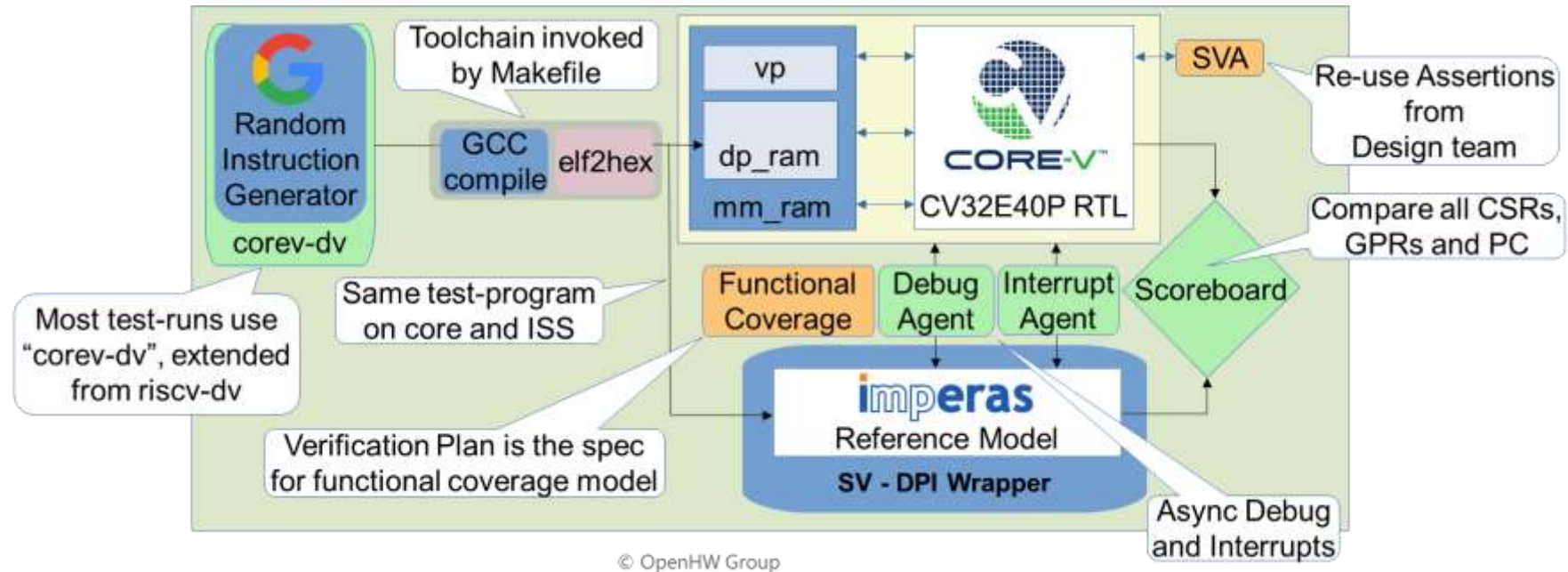
RISC-Vプロセッサ 非同期ステップコンペア DV



- Stimulus
 - Directed and constrained-random テスト
 - Random Instruction ストリーム生成
- RISC-V リファレンスモデル
 - 高品質な検証済み、命令アキュレートなプロセッサモデル
 - プロセッサのカスタマイズに対応した構成が可能
 - DUTと同じソフトウェアを実行
 - 非同期イベントへの応答が可能
- 比較
 - DUTとリファレンスモデルは同期して動作
 - 重要なイベントごとに内部状態を完全に比較
 - 例：命令のリタイア、トラップ
 - エラーは即座に報告
 - マルチハートプロセッサ、アウトオブオーダーパイプライン、マルチイシューパイプラインのサポート
- 機能カバレッジ
 - 命令、レジスタ
 - 注目すべきシーケンスとシナリオ
 - 非同期イベント

OpenHW CV32E40P (core-v-verif)

Success!



しかし大変な作業だった ... どうすれば改善できるのか？

議題



- Imperas社紹介
- RISC-V processor 設計検証 (DV:Design Verification)の課題
- **課題解決へのステップ**
 - 1) Collaboration
 - 2) Standardization
 - 3) Methodology
 - 4) **Verification IP (検証用IP)**
- Key technologies: reference models, verification IP
- Summary

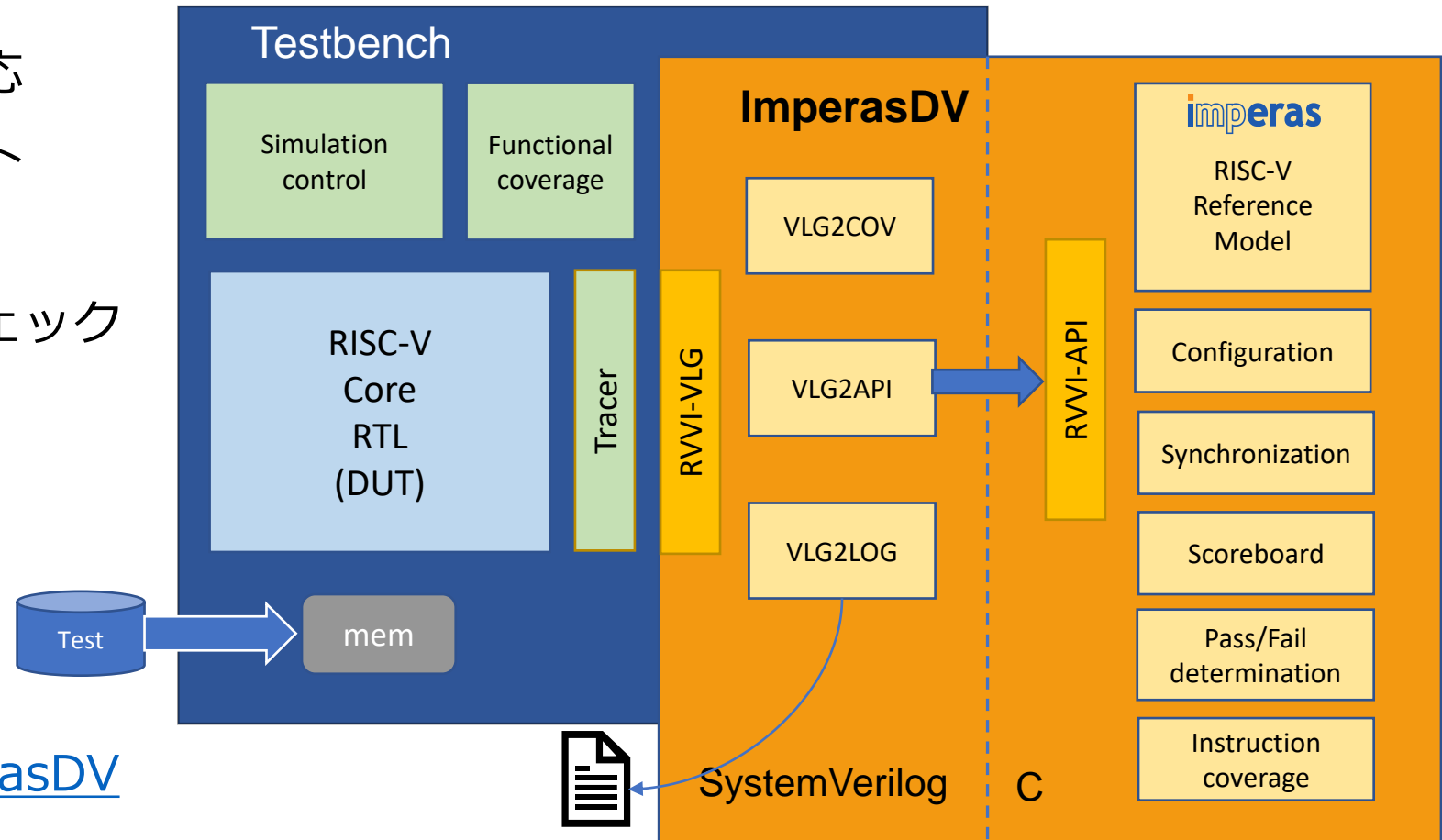
Verification IP



- 現場での実践
 - 他の人の知識や経験から利益を得る
 - 標準規格（RVVIなど）に準拠し、ベストプラクティスをサポートする
 - 時間とリソースの節約
 - 高い品質
 - カスタマー・サポート
-
- RISC-Vのエキスパートでなくても、カスタムRISC-Vプロセッサを検証できます。

ImperasDV: RISC-V VIP

- RVVIをサポート
- 非同期ステップアンドコンペア対応
- ランダム命令ストリームをサポート
- 非同期イベント対応
- 同期を管理し、すべての比較とチェックを行います
- ユーザがトレーサを実装します。
 - RVVIの要件を使用



<https://www.imperas.com/ImperasDV>

ImperasDV: Test Suite Availability



- 現在、以下の50以上のテストスイートが無料で提供されています。
 - I, M, C, F, D, B, K, V, P
 - 提供されるベクトルテストスイートは、特定のベクトルエンジン構成です。
- 現在、2つの市販のテストスイートがあります。
 - ベクタテストスイート
 - RISC-Vのあらゆるベクタ構成をサポートします。ユーザがベクタ構成を選択し、Imperasがテストスイートを生成します。
 - ベクタ命令とレジスタをテスト
 - 物理メモリ保護（PMP）テストスイート
 - RISC-VのPMPコンフィギュレーションをサポートします。ユーザがPMPコンフィギュレーションを選択し、Imperasがテストスイートを生成します。
 - PMPのCSRと例外をテスト

ImperasDV: Functional Coverage Library Availability



- 異なるRISC-V拡張グループに対応したSystemVerilog
 - 現在、RV[32,64]IMCで使用可能です。
 - オープンソース、拡張可能。
 - RVVI-TRACEデータを使用
-
- RISC-Vの全仕様に対応した命令カバレッジを実現

議題



- Imperas社紹介
- RISC-V processor 設計検証 (DV:Design Verification)の課題
- 課題解決へのステップ
 - 1) Collaboration
 - 2) Standardization
 - 3) Methodology
 - 4) Verification IP
- **Key technologies: reference models, verification IP**
- Summary

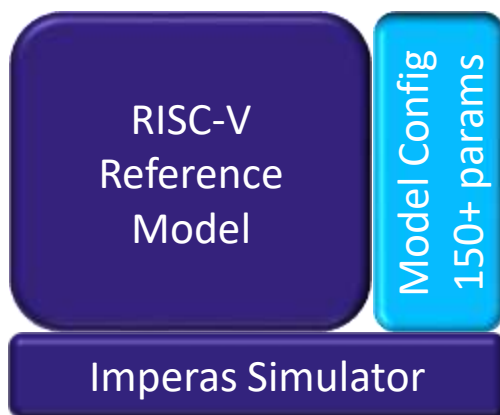
RISC-V Model Requirements

Not Just for DV; Also for SW Dev



- FIXされた仕様の全バージョンとまだFIXされていない拡張機能を含むISAのモデル化
 - 次のプロジェクトのためにモデルを簡単に更新、設定可能
 - カスタム命令、レジスタなど、ユーザによる拡張が可能
 - Andes, SiFive, OpenHW, Cudasip, MIPSなどの実際のプロセッサIPをモデル化できます。
 - カバレッジメトリクスを含む明確なテストプロセス
 - SystemVerilog、SystemC、Imperasなどの他の仮想プラットフォーム・シミュレータとのインターフェイス
 - GDB/Eclipse、Imperas MPDなどのソフトウェアデバッグツールとのインターフェイス
 - プロセッサの内部状態へのアクセスなど、ソフトウェア解析ツールとのインタフェース
 - 実行時間計測への拡張性を含む、アーキテクチャ探索ツールとのインタフェース
-
- Imperasのモデルとシミュレータは、これらの要件を全て満たすように構築されています。過去12年以上にわたるRISC-V ISA以外での開発経験が生かされています。

Imperasのリファレンスモデル



<http://www.imperas.com/riscv>

- ImperasはRISC-Vフル仕様のモデルを提供します。
- 製品品質のRISC-Vプロセッサモデル/シミュレータはコンプライアンス、検証およびテスト環境として使用されています。
- 全ての機能が実装されたコンフィギュレーション可能なモデル / シミュレータ
 - ユーザーおよび特権モード仕様をサポートした全ての32bitおよび64bit機能
 - ベクター拡張命令 : versions 0.7.1, 0.8, 0.9, 1.0をサポート
 - ビット操作拡張命令 : versions 0.91, 0.92, 0.93, 1.0.0をサポート
 - ハイパーバイザー : version 0.6.1をサポート
 - K-Cryptoスカラー : version 0.7.1, 1.0.0をサポート
 - デバッグ : versions 0.13.2, 0.14, 1.0.0をサポート
- モデルのソースコードはApache 2.0 open sourceライセンスで公開
- リファレンスモデルとして以下のお客様で使用されています。
 - Mellanox/Nvidia, Seagate, NSITEXE/Denso, Google Cloud, Chips Alliance, lowRISC, OpenHW Group, Andes, Valtrix, SiFive, Cudasip, MIPS, Nagra/Kudelski, Silicon Labs, RISC-V Compliance Working Group, ...

ImperasのモデルはRISC-V Goldenリファレンスモデルとして利用されています

Imperasモデルの拡張性について

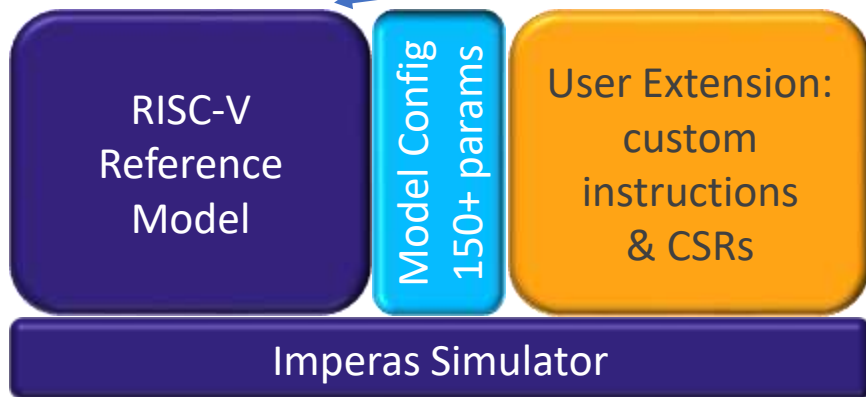


Imperasが開発およびメンテナンスを行う基本モデル

- 基本モデルは全てのRISC-V仕様が実装されています
- どの拡張ISAを使用するかを自由にコンフィギュレーションできます。
- また、拡張ISAの中のどのバージョンの拡張命令を使用するかをコンフィギュレーションできます。
 - ISA拡張命令仕様のアップデートに合わせてモデルも更新されます。
- RISC-Vのオプション仕様のコンフィギュレーションが可能です
 - 例えば、オプションのCSRs, read onlyまたはread/writeビットオプション等

Imperasは拡張可能な基本モデルの拡張方法を提供します

- カスタム命令追加の為のテンプレート
- 機能追加の為のコード
- ユーザーガイド/多くのサンプルを含むリファレンスマニュアル
 - プロセッサモデルの拡張サンプル等



- 分割され、重複の無いソースファイルがメンテナンスを容易にします
- Imperasまたはお客様が拡張部分を開発可能です
- ユーザー拡張のソースコードはお客様の権利となります

Imperasのモデルは容易に拡張、およびメンテナンスが可能なような構造に設計されています

OVP Library of RISC-V Fast Processor Models



- OVPはこちらから利用可能 <https://ovpworld.org>
 - RV32/64 IMAFDCEVBHKP M/S/U 特権モードの汎用モデルまたはエンベロープモデル
 - プロセッサIPベンダーのモデル Andes、Cudasip、MIPS、OpenHW、SiFive
 - RISC-Vプロセッサを自作するユーザ向けのカスタムモデル
- テスト駆動開発(TDD)手法でモデルを構築
 - 機能追加と同時にテストも構築
 - 継続的インテグレーション(CI)テストフローを採用
 - > モデル+シミュレータで15,000以上のテストを実施
 - プロセッサIPベンダによる追加テストによるモデルの検証

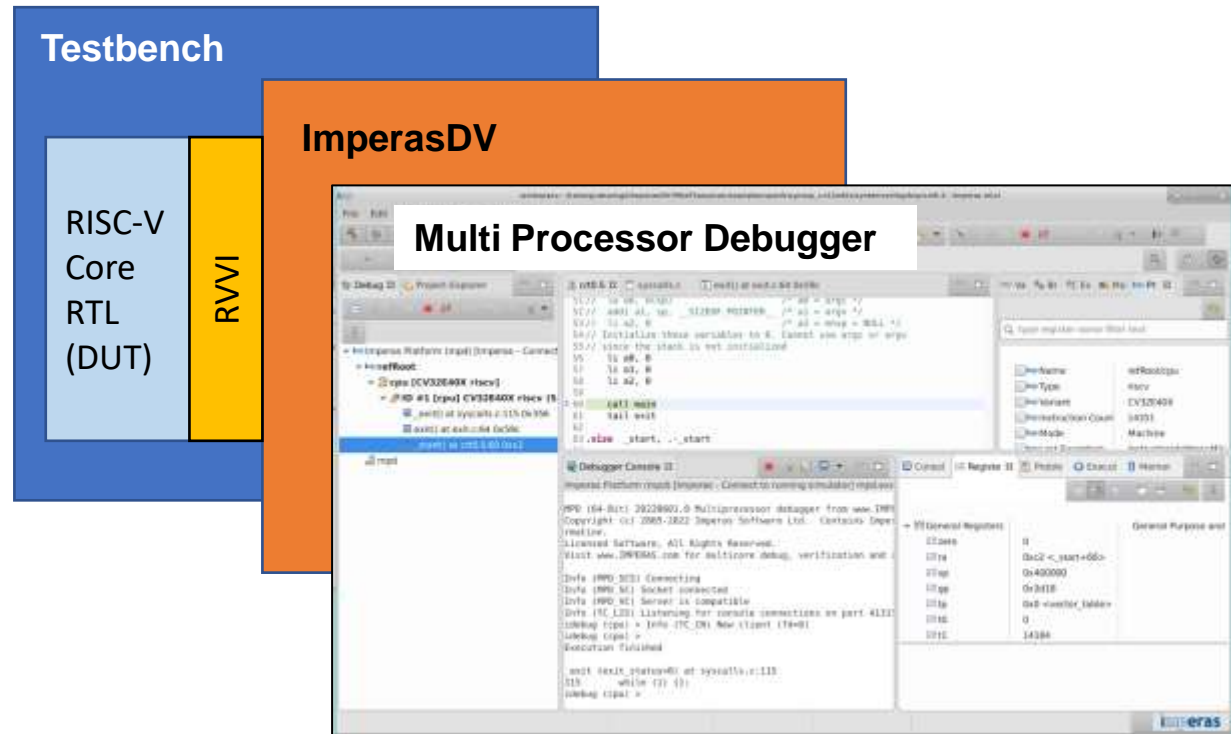
“The Imperas virtual platform solutions for software development, debug and test, along with their open-source models, will help accelerate SoC and embedded software development for our customers.”

Charlie Hong-Men Su, Ph.D., Andes Technology CTO



ImperasDV: Debug with MPD

- Imperas MPD は Eclipse ベースのデバッグツールです。
- ソースラインまたは命令レベルでのデバッグが可能
- 新しいカスタム命令や追加されたステートレジスタを確認可能
- 最初のミスマッチでブレイクし、SWとRTLを同時にデバッグ可能



なぜRISC-V VIPを使う必要があるのか？



- 開発スケジュールの短縮
 - ドメイン固有のタスクにリソースを割くことができます。
- ベストインクラスのメソドロジー
 - Constrained-random、カバレッジドリブン、非同期ステップアンドコンペア
 - 障害発生時のデバッグ機能
- 標準インターフェース(RVVI)による移植性・再利用性
- 高品質なリファレンスモデル
 - プロセッサの専門家により開発されたモデル
- ドキュメンテーション、トレーニング、継続的なサポート

What's Next for RVVI and RISC-V DV Methodology?

- Stimulus:
 - 非同期イベントをプログラムフローと協調させるためのより良い方法
 - RVVI 仮想周辺機器インタフェース
- 品質
 - 検証品質の定量的な測定
 - 故障注入/mutationテスト
- ファンクショナル・カバレッジ
- ユーザーフィードバックをお願いします！

議題



- Imperas社紹介
- RISC-V processor 設計検証 (DV:Design Verification)の課題
- 課題解決へのステップ
 - 1) Collaboration
 - 2) Standardization
 - 3) Methodology
 - 4) Verification IP
- Key technologies: reference models, verification IP
- **Summary**

まとめ

- RISC-Vプロセッサの開発者は、RISC-Vの柔軟な仕様に対応する為の高品質なRTL検証環境が必要です。
- プロセッサDV手法はImperasとお客様、およびパートナーとで進化させてきました。
- RVVIのようなオープンスタンダードは、効率的なメソドロジーを実現し、RISC-Vエコシステムを発展させるために不可欠です。
- 非同期ステップ・アンド・コンペア手法は包括的で効率的なRISC-V設計検証フローです。
- RISC-V DVのための検証用IP = ベストプラクティス、オープンスタンダード、高品質なリファレンスモデル（例：Imperas OVPリファレンスモデル）



RISC-V Days Tokyo 2022 Autumn



Thank you!

Shuzo Tanaka, eSOL TRINITY Co., Ltd.
sh-tanaka@esol-trinity.co.jp

RVVI: RISC-V verification Interface
<https://github.com/riscv-verification/RVVI>

ImperasDV
<https://www.imperas.com/imperasdv>