NEWS 2018

日本語版

ソースコード
カバレッジ

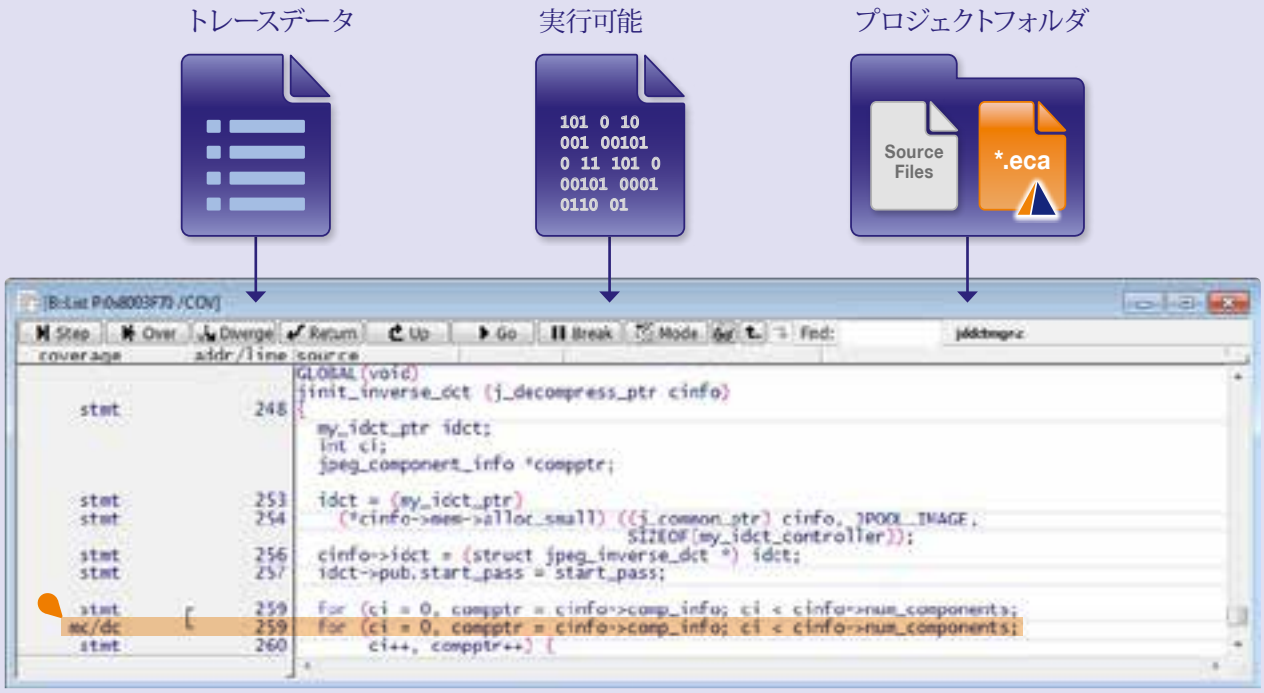
✓ コードのインスツルメンテーション不要

CONTENTS

LAUTERBACH
DEVELOPMENT TOOLS

トレースベースMCDCカバレッジ

MCDCカバレッジ



2018年3月にローターバッハ社はトレースベースのMCDCカバレッジを発表します。これでTRACE32はソースコードレベルで主要なコードカバレッジメトリクスの全てをサポートすることになります。

DO-178Cによる定義 [3]

ステートメントカバレッジ：プログラム内のすべての命令文が最低1回は呼び出される。

デシジョンカバレッジ：プログラムのすべての入り口と出口が最低1回は呼び出され、プログラム内の判断文は、最低1回すべての可能な結果を得ている。

Modified Condition/Decision Coverage (MC/DC) カバレッジ：プログラムのすべての入り口と出口が最低1回は呼び出される。プログラム内の判断文の各条件は、最低 1 回は、すべての可能な結果を得ている。各判断文は、最低1回はすべての可能な結果を得ている。判断文の中の各条件は、その判定文の結果に独立して影響を与えることが示されている。

判定文の結果に独立して影響を及ぼす条件は、

(1)他の全ての可能な条件を変えずに、その条件のみ変えること、もしくは、

(2)結果に影響を与える可能性のあるその他すべての条件を固定して、その条件のみを変えることによって示される。

安全性が問われるシステムの開発において、ソフトウェアが完全にテストされたことを実証するために、コードカバレッジプロセスが使われます。ISO　26262やDO-178Cなどのソフトウェア開発規格では、コードカバレッジの証明が開発サイクルの必須要素であると規定されています。

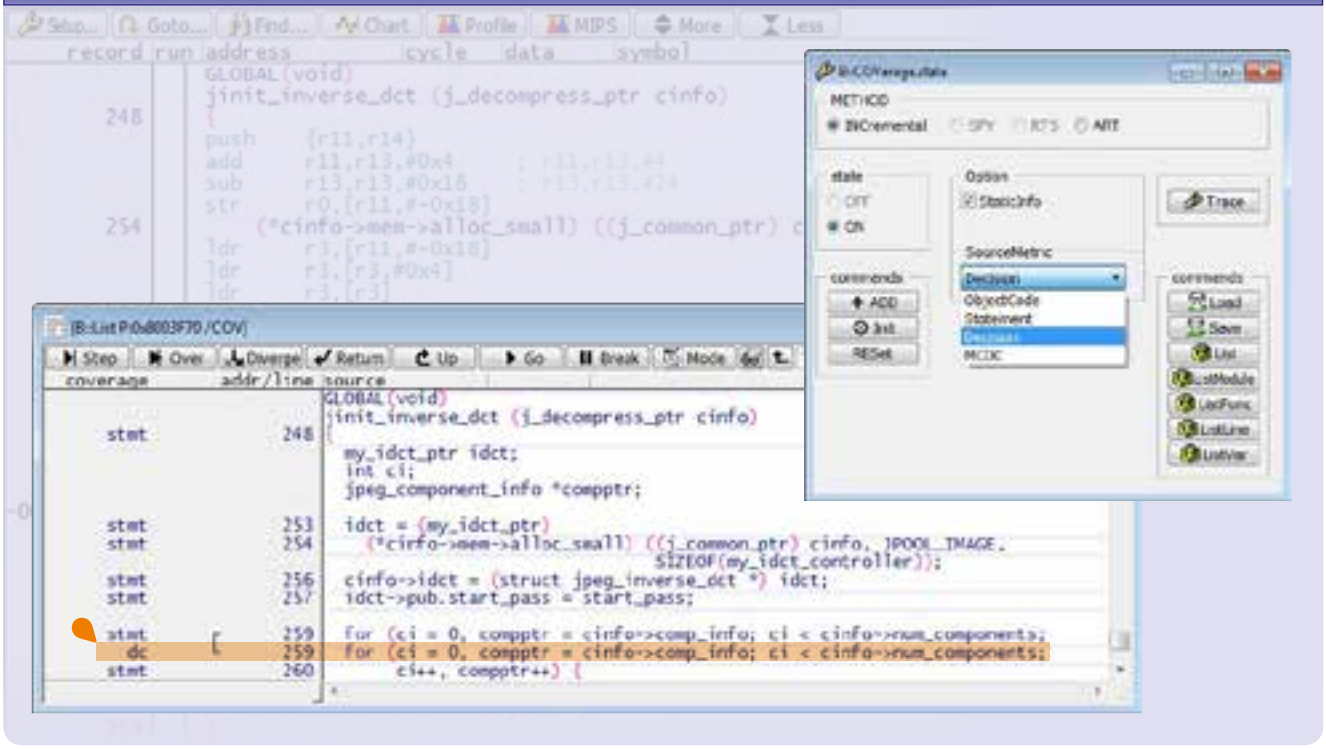
トレースベースコードカバレッジ

リアルタイムトレースツールのマーケットリーダーであるローターバッハ社は、ターゲットコードのインスツルメンテーションを必要としないトレースベースのコードカバレッジ測定を提供します。プログラムの実行に関する情報は、まずオブジェクトコードレベルで追跡されます。これにより、次のカバレッジメトリックを簡単に証明できます。

- **オブジェクトステートメントカバレッジ**　各アッセンブラ命令がテスト間に最低1回は実行されたことを証明します。
- **オブジェクトブランチカバレッジ**　各条件分岐が少なくとも 1 回は成立したこと、および少なくとも 1 回は成立しなかったことを証明します。

オブジェクトコードレベルでのコードカバレッジ測定は、全てのTRACE32トレースツールに機能として備わっています。2017年にはステートメントカバレッジ、デシジョンカ

デシジョンカバレッジ



バレッジが追加され、ローターバッハ社のツールはソースコードレベルのカバレッジを証明します。(上部「デシジョンカバレッジ」図を参照)　義務化が進んでいるMCDCカバレッジ測定の実行に、多くのお客様がTRACE32トレースツールを使用したいと考えています。

MCDC コードカバレッジ

以前は、オブジェクトブランチカバレッジはソースコードレベルでのMCDCカバレッジ測定の代替として適しているという意見が大半でした。しかし、航空業界で商業航空安全チーム(CAST)がこの見解に強く異論を唱えています。([1]参照)

現在、ほとんどの開発者は、MCDCのカバレッジデータを収集するために、ソースコードのインスツルメンテーションに依存しています。　ローターバッハ社のエンジニアは、ターゲットコードを変更する必要なく、MCDCカバレッジを提供するという目標を掲げ、多くの白書や出版物を調査、研究しました。それぞれ独自のアプローチを論じていましたが、全てに共通した根本的な類似点がありました。

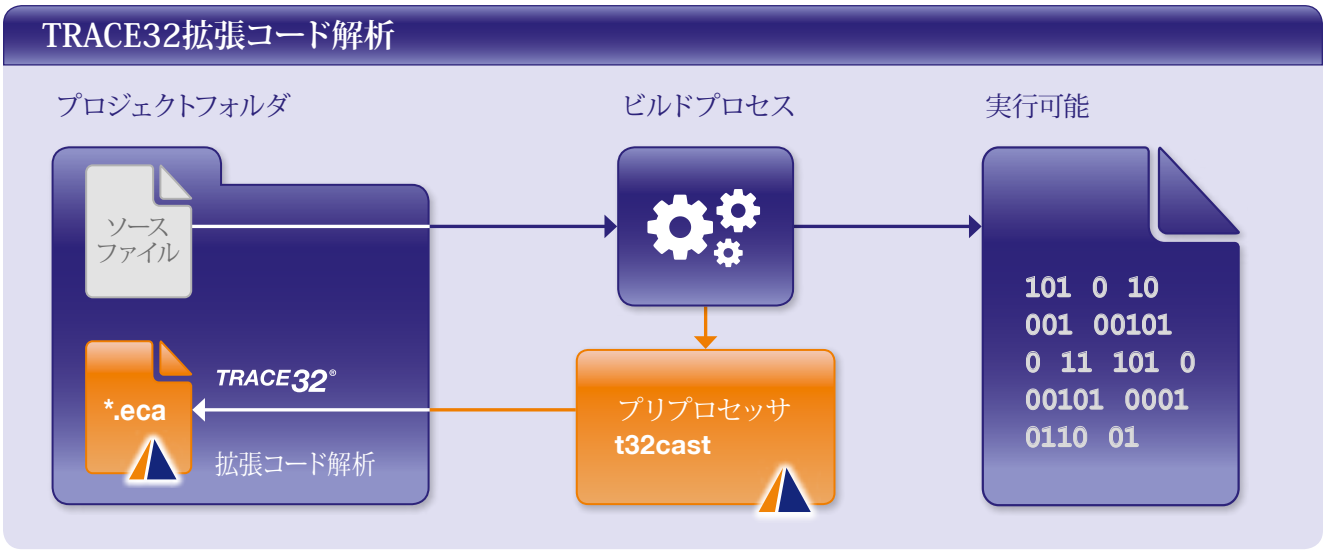
1. トレースデータを基に確実にMCDCカバレッジを証明できるようにするためには、ソースコード内の判定文の構造と位置を認識する必要がある。
2. 同時に、ソースコードの各条件は、条件の算術的表現でなく、オブジェクトコードレベルで条件付き分岐、または、条件付き命令によって表現されなければならない。
3. MCDCカバレッジ分析が実行される際、オブジェクトコードの内の判断文の構造と位置を認識する必要がある。

コードへのインスツルメンテーションなしでMCDCカバレッジ測定を実行するには、要件1によると、ソースコードの構造に関する追加情報が必要となります。この情報は現在、コンパイラが生成する情報に含まれていません。これに加えて、要件2に適合する形式でコンパイラがオブジェクトコードを生成するようにしなければなりません。

AdaCore

コンパイラとMCDCカバレッジ分析を実行するソフトウェアのプロバイダが同じ場合、MCDCカバレッジをトレース記録から簡単に証明することができます。コンパイラは各判定文の構造とソースコード内の位置に関する新しい情報をデバッグ情報に含めることができます。AdaCore社はそれが可能なソリューション([2]参照)を提供しています。また、AdaCore社はトーレステータを生成するためのターゲットエミュレーションソリューションも提供しています。

最終実装のターゲットハードウェア上でさらにMCDCカバレッジも証明する必要のあるお客様のために、AdaCore社はTRACE32で記録したトレースデータを必要な解析のためにインポートできるインターフェイスを提供しています。



t32cast

ローターバッハ社の製品をご使用のお客様は、リアルタイムトレース記録に基づき、2018年3月からTRACE32でMCDCカバレッジを証明することも可能になります。ローターバッハ社は、C/C++ソースコードを解析するための**t32cast**コマンドラインツールを提供し、MCDCカバレッジ測定実行に必要な判定文の構造に関する情報が各ソースコードファイルに対して生成されます。そのため、この情報が生成されるようにビルドプロセスを調整する必要があります。(上記のTRACE32拡張コード解析」図を参照)**t32cast**コマンドラインツールはコンパイラに依存せず、既存のビルド環境に簡単に統合することができます。

TRACE32でMCDCカバレッジ測定が開始されると、TRACE32は**t32cast**ツールが生成した、必要な全ての.ecaファイルを自動的に読み込みます。その後、TRACE32はデバッグ情報を活用して、ソースコードの判定文をオブジェクトコードにマッピングします。
しかしながら、このプロセスの間、最適化を無効にするなどをして、選択されたコンパイラがオブジェクトコードレベルではソースコード内の各条件を条件ジャンプまたは条件付命令によって表すことを保証しなければなりません。

まとめ

TRACE32　MCDCカバレッジは、コンパイラやプロセッサアーキテクチャから独立して使用することができます。安全性が重要なシステムを実装する際にもコードの最適化を断念したくない場合、トレースベースのMCDCカバレッジを使うためには最適化を減らす以外に方法はありません。

将来的には、コンパイラが選択した最適化のレベルから独立して、判定文に対してはオブジェクトコードレベルで条件付き分岐や条件付き命令のみに変換し、選択した最適化がその他の領域全体で実行されるようになったら大変歓迎すべきことです。

参照文献

[1] CAST-17 Position Paper (2003, January). Structural Coverage of Object Code
[2] Comar, C., Guitton, J., Hainque, O., & Quinot, T. (2012, May). Formalization and comparison of MCDC and object branch coverage criteria. In ERTS (Embedded Real Time Software and Systems Conference).
[3] RTCA Inc. (2011, December) RTCA/DO-178C Software Considerations in Airborne Systems and Equipment Certification

コードカバレッジライブ

トレースデータのリアルタイム処理

リアルタイムプロファイリング（RTS）は、トレースデータを記録中にホストにストリーミングして、即座に解析するトレースモードです。このモードでは、コードカバレッジ解析の結果を画面上にリアルタイムで表示できます。2017年末に、Arm-ETMv3向けに2009年から提供されているRTSモードが、さらに他のトレースプロトコルに展開されました。「RTS-サポートされているトレースプロトコル」の表では現在サポートされているトレースプロトコルが一覧で確認できます。

基本条件

次の基本条件が、サポートされている全てのトレースプロトコルに適用されます。

1. トレースデータのデコーディングには、プログラムコードが必要です。プログラムの実行中にメモリからコードを読み取ると時間がかかるので、ライブ解析を開始する前にTRACE32にコードを読み込んでください。つまり、静的プログラムに対してのみ、ライブ解析が実行できます。
2. トレースデータのライブ解析は、トレースポートの平均データ転送率がホストコンピュータへの転送速度を超えない場合のみ可能です。現在のトレースツールにはUSB 3 インターフェースが装備されているので、ホストコンピュータへの最大データ転送速度は、約180 MByte/秒です。転送速度は2009年と比較して、3倍になりました。
3. 現在は、シングルコアとSMPマルチコアトレースストリームのトレースデータのライブ解析が可能です。AMPマルチコアトレースストリームの実装は開発段階にあります。
4. ライブコードカバレッジ測定は、オブジェクトステートメントカバレッジとオブジェクトブランチカバレッジ、ステートメントカバレッジ、デシジョンカバレッジのメトリクスに使用できます。

一般に、トレースデータは解析後には必要なくなります。しかし、TRACE32では解析中にトレースデータをストリーミングファイルに保存するオプションを提供します。コードカバレッジ測定が完了した後に、もう一度トレースデータを詳しく検証することができます。従来のTRACE32コードカバレッジ測定のように、RTSモードは包括的なテストレポートを作成します。

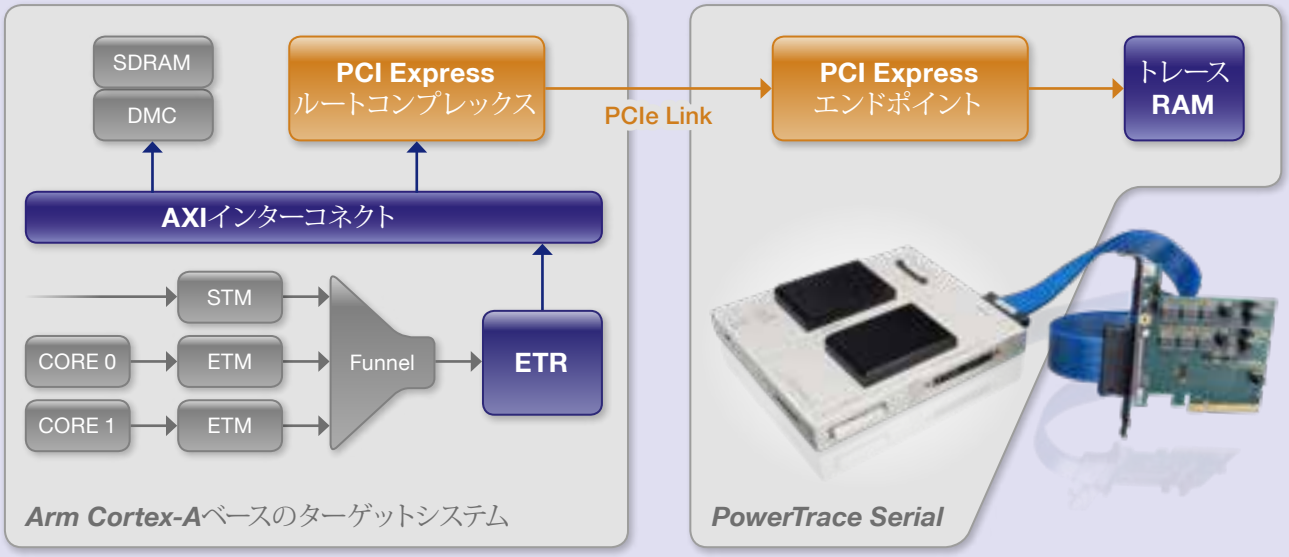
RTS – サポートされているトレースプロトコル

- ETM v3 on Arm / Cortex®
- PTM on Arm / Cortex®
- ETM v4 on Arm / Cortex®
- MCDS on Infineon TriCore™
- Nexus on NXP MPC5xxx / STM SPC5xx
- Nexus on NXP PPC QorIQ®

others on request

PCI Express経由のトレース

PCIeエンドポイントとしての TRACE32 PowerTrace Serial



チップにPCIeインターフェースが装備されている場合は、外部トレースツールを使用してトレースデータの記録と解析に使用することができます。以前からローターバッハ製品を導入しているお客様は、既にArm Cortex-A®やNXP Powerアーキテクチャーベースの QorIQ®プロセッサに対してこのトレース方法を使用しています。

PCIeエンドポイントとしてのトレースツール

まず、PCI Expressによるトレースの仕組みを簡単に説明します。TRACE32 PowerTrace SerialはPCIeエンドポイントとして動作するように設定されています。したがって、ソフトウェア（例：ブートローダ）がエンドポイントを列挙して設定し始める前に、トレースツールをターゲットに接続する必要があります。設定中に各エンドポイントにシステムメモリ内のアドレス範囲が割り当てられます。そのため、エンドポイントのデータを該当するアドレスに簡単に書き込めます。

ターゲットシステムのトレースインフラストラクチャは、TRACE32 PowerTrace Serialのエンドポイントに割り当てられたアドレス範囲にトレースデータを書き込むように設定しなければなりません。これは、ほとんどのプロセッサで実施できます。その後にトレースが始まります。

特性

可変データレート
- Gen1 レーンあたり250 MByte/秒
- Gen2 レーンあたり500 MByte/秒
- Gen3 レーンあたり984 MByte/秒

可変ポート幅 (1, 2, 4, 8 レーン)

フルサイズのカードアダプタが利用可能
mini-PCIeカードアダプタを予定

トレースメモリ：4 GigaByte
全ての標準プロトコルのトレースデコーディング

要約

PCIeを介したトレースは簡単です。専用のトレースインターフェースのないターゲットシステムの場合、この技術は大容量のトレースデータを記録するのに最適です。PCIeなどの外部インターフェースと専用トレースポートによるトレースの主な違いは、次のように要約できます。

- ターゲットソフトウェアがPCIeルートコンプレックスを構成した後にのみ、トレースの記録を開始することができます。これはオペレーティングシステムに割り当てられたタスクです。
- 同時にPCIeを介したトレースは、厳密に言うとターゲットのリアルタイム実行性に影響を与えます。システムメモリの一部を使用します。さらに、トレースはPCIバスの大域幅に関して他のエンドポイントとも競合します。

Wind RiverからTRACE32へシームレスな移行

ウインドリバー社が2014年にJTAGデバッガの提供を止めて以来、製品の管理開発のために、TRACE32に切り替えた既存のユーザが多くいます。ウインドリバー社と密に連携し、ローターバッハ社はTRACE32ソフトウェアを強化し、ユーザーの皆様が通常通り作業できるように、システムの基本的な調整はシームレスに実装されています。

すべてのプロセッサに対応

TRACE32に切り替える大多数のお客様は、Power Architecture™ ファミリを使用しています。ローターバッハ社製品は1997年以来、このアーキテクチャをサポートしており、TRACE32は実績のある信頼できるデバッグソリューションです。

スクリプトコンバータ

通常、デバッグを開始する前にプログラムコードがターゲットフラッシュにプログラムされます。TRACE32はこれにスクリプトを使用します。プロセッサ構成レジスタ、特にSDRAM構成はスクリプトの重要なコンポーネントです。ウインドリバー社は、「レジスタ構成ファイル」に必要な設定を定義しています。ローターバッハ社はこのファイルをTRACE32スクリプトに変換するために、特別なコンバータを提供しています。プロフェッショナルデバッガを使用して、何年もの間、積み重ねてきた多数の複雑なテストスクリプトをTRACE32に転送できるように、スクリプトコンバータを提供しています。

Wind River Workbench

数年もの間使用してきた実績のあるシステムから新しいデバッガに切り替えるのは、不安を伴います。そのため、ローターバッハ社は、TCFエージェントとして作動するように2015年にデバッガを強化しました。 Wind River Workbenchをデバッグ IDEとして、TRACE32デバッガをデバッグバックエンドとして使用することを可能にしたのです。

ウインドリバー社の全製品に対応

ほとんどのプロジェクトでは、例えば、Wind River VxWorks、Wind River Linux、Wind Riverハイパーバイザといったウインドリバー社のソフトウェア製品も使用されています。当然のことですが、ウインドリバー社のデバッガを使用する際に、これらの製品の包括的なデバッグソリューションが使用されていました。ローターバッハ社はこの分野で非常に豊富な経験を持つツールプロバイダであり、1996年以来、ウインドリバーVxWorksのデバッグをサポートしてきました。完全なウインドリバーLinuxのサポートは2000年に追加されました。VxWorks 7、VxWorks 653 v4、ウインドリバー、ハイパーバイザv3など、現在Workbenchでサポートされていない製品もサポートしています。

RISC-V用TRACE32デバッガ

JTAG デバッガ

ローターバッハ社は、新しいRISC-Vデバッガを2017年11月に発売しました。現在サポートされている最初のチップは、E31コアコンプレックス（32ビット）と、E51（64ビット）SiFive社のCore Complexです。

RISC-Vは、オープン命令セットアーキテクチャ（ISA）で、RISC-V　Foundation（https://riscv.org）の管理により、規定されたRISCの原則に基づいて体系化され、規定、開発されています。元々は学術研究のために設立されましたが、RISC-Vは現在、組込み市場で牽引役を果たしており、専門的なハードウェアデバッガを必要不可欠なものにしています。

ローターバッハ社のRISC-Vデバッガの実装は、「RISC-V外部デバッグサポート」というオープンソース仕様に基づいており、その仕様は2018年にRISC-V　Foundationに採用される予定です。この仕様の目的はフレキシブルな停止モードデバッグです。RISC-Vコアの各ハードウェアスレッドはリセットから直接デバッグする必要があります。

TRACE32でRISC-Vプロセッサをデバッグするためには、現在のところは、JTAG　DTM（デバッグトランスポートモジュール）が装備されている必用があります。 DTMは、独立した交換可能なモジュールであり、チップメーカーは、別の通信を介してデバッグモジュールにアクセスすることができます。ローターバッハ社は様々なデバッグ通信インターフェースにおける豊富な経験を誇る実装メーカーです。

TRACE32デバッガは、デバッグモジュール経由でプロセッサの全標準デバッグ機能にアクセスできます。
これらはデバッグレジスタ、抽象コマンドのように、仕様の一部です。さらに、この仕様により、専用デバッグ機能の設計も可能です。TRACE32　RISC-Vデバッガは、圧縮命令、浮動小数点といった異なる標準のISA拡張を既にサポートしていますが、顧客専用のISA拡張機能も追加することができます。