



## THE NEW HEADQUARTERS IN GERMANY

### ローターバッハの30年 - さらにイノベーションのために

ローターバッハは成功の歴史の中で30周年を迎え、2008年12月には本社を Arcone Technology Center に移しました。エルゴノミクスに配慮したこの新社屋は、ローターバッハ技術者チームの能力拡大を促し、TRACE32 製品が持つ創造性に富んだキャラクターとローターバッハの優れた技術サポートをより高いレベルへと引き上げ、顧客サービスのいっそうの向上に貢献することをお約束します。

1979年にたった1人で設立されたローターバッハも、今や世界中で事業を展開する企業へと成長し、ドイツ国内で70人、さらに全世界の各支社で30人の従業員を抱えるまでになりました。そのため、大きい施設への移転は、ローターバッハが今後も成長を続けるうえで不可欠なことでした。

できます。また共有のオープンスペースも豊富なためスタッフ同士での斬新なアイデアの交換を促し、生産性の向上を目指します。

ローターバッハでは職業訓練コースの拡充を予定しているため、それに適した施設空間の提供も重要であり、計画段階からこの要求が盛り込まれていました。顧客向けのトレーニング、会議、その他のイベントといった各種の用途に応じた部屋が用意され、どの部屋にも最新のメディア機器が配備されています。

独立した食堂も完備しており、訪れたお客様に有機食メニューを提供しています。

»

### Arcone Technology Center

Arcone Technology Center は、ミュンヘン近郊の Höhenkirchen-Siegertsbrunn にローターバッハが2年間をかけて建設した施設で、ローターバッハの他にも中規模のテクノロジー企業が入居できるオフィスビルとして設計されました。新しい本社の拠点として Höhenkirchen-Siegertsbrunn を選定するにあたっては、ミュンヘンと空港に近いことが決定的な要因でした。また、国内の各アウトバーンにもアクセスしやすく、公共機関の便も良い立地です。

Arcone Technology Center には、ローターバッハのスタッフが活躍するスペースが十分に確保されており、個々人の業務と必要性に応じて高い自由度で職場環境を整えることが

### コンテンツ

ETMv3 長時間トレース	
• 実装	3
• コードカバレッジ	7
• プロファイリング	8
新サポートプロセッサ	9
シリアル GigaBit トレースインタフェース	10
ARM CoreSight デバッグ	11
256 メガレコードロジックアナライザ	14
RTOS デバッグの最新動向	15

## 新しいコーポレートデザイン

移転に伴って、ローターバッハはコーポレートデザインの変更にも踏み切りました。コーポレートデザインの刷新とそれに伴う活動は、すべて2009年3月末までに完了する予定です。今後のカンパニーイメージは新たにデザインされた現代的な配色の独自ロゴに特徴付けられます。旧ロゴデザインに似た三角形の印象的な新ロゴが既に使用されています。このデザインでは、伝統だけではなく "Leading through Technology" のスローガンのもとに組み込み向け開発ツール



図1:心地良い空間を作り出すモダンで採光の良いワークスペース



図2:新しい発想を出し合う、定例あるいは即席の打ち合わせの場としての共用オープンスペース

市場における世界企業としてあり続ける未来志向企業であることを表現しています。

昨年の2008年も勿論、躍進を続けた1年でしたが、この2009年のニュースレターでも、昨年から取り組んできた技術的な進展を紹介します。開発環境の効率化をもたらすこれらの新機能へのご興味のきっかけになることを期待しています。

## TRACE32フロントエンドのフローティングライセンス

2008年9月から、ローターバッハはTRACE32フロントエンド製品にフローティングライセンスを提供しています。ターゲットハードウェアでの従来型のデバッグに加えて、組み込みソフトウェアのテストには仮想プロトタイプの利用

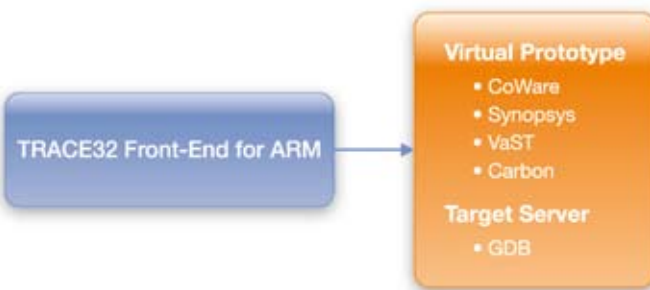


図3: "ARM対応TRACE32フロントエンド"は各種の仮想プロトタイプおよびGDBをサポート

が増えていきます。JTAG インタフェースの代わりに、デバッグ機能にソフトウェアインタフェース (API) を利用できます。ローターバッハの最新のデバッグソフトウェアフロントエンドを使用すれば、洗練されたTRACE32 GUIを使用してデバッグすることが可能です。

ソフトウェアフロントエンドは、仮想プロトタイプだけでなく、他のソフトウェアソリューションでのデバッグもサポートするように設計されており、半導体メーカーが提供

するコアシミュレータや、GDBのようなターゲットサーバーもその対象になっています。

ローターバッハのTRACE32フロントエンドは、一般的な組み込みプロセッサアーキテクチャのすべてに対応しており、たとえばARMおよびCortexプロセッサにはTRACE32 Front-End for ARMが用意されています。すべてのバージョンには、対応する最新のデバッグAPIが付属します (図3を参照)。

TRACE32フロントエンドを使用するには、以下の2つのライセンスモデルがあります:

### 1. フローティングライセンス

Reprise License Manager (RLM) を利用し、特定数のフロントエンドライセンスをネットワークで使用できます。RLMの詳細については、下記のサイトを参照してください:

[www.reprisesoftware.com](http://www.reprisesoftware.com)

### 2. USB ドングル

シングルユーザーライセンスの場合、USB ドングルを利用できます。

## ETMv3長時間トレース

# 長時間トレース

高機能なリアルタイムトレースツールを提供している世界のリーディングカンパニーである Lauterbach 社から、新たに ARM ETMv3 に対応した長時間トレース機能を紹介します。これは、TRACE32 のプロファイリングおよびコードカバレッジ機能を、非常に長い測定時間に対応させるための新技術です。

このパートでは、長時間トレース技術のコンセプトと、トレースツールおよび各ホストコンピュータに対する技術要件について説明します。

## ARM ETMv3

トレースとは、コアで実行されるプログラムコードについての詳細な情報を記録する処理です。この情報は、通常オンチップのトレースロジックによって生成されます。ARM コアの場合、このロジック要素は Embedded Trace Macrocell (ETM) として知られています。このロジックの最新バージョンである ETMv3 は、現在ほとんどの ARM11 および Cortex コアで利用されています。オンチップトレースロジックの機能はトレースデータの基礎となるものですので、まずその紹介から始めましょう。

ETMv3 は、パッケージ指向のトレースログを生成します。プログラムの実行時に以下の情報が生成され、トレースパッケージで収集されます：

- **プログラムフローパッケージ:** コアによって実行されたプログラム命令に関する情報が収められます。主に、ジャンプのターゲットアドレスや、2つのジャンプ間で実行された命令の数などが該当します。
- **データフローパッケージ:** プログラムによって読み取り/書き込みが行われたメモリアドレスや、各データの値が収められます。
- **コンテキストIDパッケージ:** オペレーティングシステムが動作中の場合に、プロセス/タスクのIDが収められます。

トレースパッケージは、オンチップトレースロジックによって、トレースポートを経由して出力されます。ETMv3 のトレースポートは、一般にトレースパッケージ用の 8 ピンまたは 16 ピンと、制御信号用の 2 ピンで構成されています。

パッケージ出力の帯域幅を最小限に抑えるために、ETMv3 はトレースパッケージを圧縮します。たとえば、アドレスはすべて特殊なアルゴリズムによって短縮されます。しかし、データ量がトレースポートの最大帯域幅よりも大きい場合には、FIFO バッファがオーバーフローし、トレースパッケージの一部が失われる可能性があります。

トレース情報を圧縮するだけでは FIFO のオーバーフロー予

防対策として万全ではありません。生成および出力するトレース情報をユーザーが簡単に定義できる ETMv3 のプログラマビリティによってトレースパッケージの数を減らすことにより、ようやく万全となります。たとえば、TRACE32 のプロファイリング機能にデータフロー情報は必要ありません。ポートでトレース量が多くなる主な原因はデータパッ

## 従来型のトレース



ケースであるため、この情報を制限することは非常に有効です。

現在、従来型のトレースは、一続きで実施される次の2つのステップで構成されます：

### 1. 記録

トレースパッケージがトレースポートでサンプリングされ、トレースメモリに格納されます。

### 2. 解析

トレースパッケージがトレースメモリからホストに転送され、そこで展開し、解析されます。

従来型のトレース手法では、保存して評価・解析できる範囲がバッファメモリに格納できるプログラムセクションに限られるという制限があります。TRACE32 トレースツールのメモリ深度は現在 1 ~ 4GB であるため、最大 3G までのトレースパッケージを記録できます。

»

記録

従来型のトレースで現実的に問題になるのが、トレースパッケージの記録です。現在の ARM コアは動作周波数が最大 1GHz であるため、高速なトレースポートがないと、すべてのトレースパッケージを損失なく出力することは保証できません。

ローターバツハの平行 ETMv3 用トレースツールは最大 275MHz DDR という周波数でのパッケージ記録をサポートするため、以下のデータ転送速度を扱うことができます (図 4 を参照) :

- トレースパッケージ用の 16 ピンの場合、8.8 GBit/s
- トレースパッケージ用の 8 ピンの場合、4.4 GBit/s



図4: 平行 ETMv3 対応のトレースツールはトレースパッケージ用 16 ピン構成で 8.8 GBit/s のデータ速度をサポート

ETMv3 用のシリアルトレースツールでは、最大 20 GBit/s のデータ転送速度で記録することが可能です。シリアルトレースツールの詳細については、10 ページを参照してください。

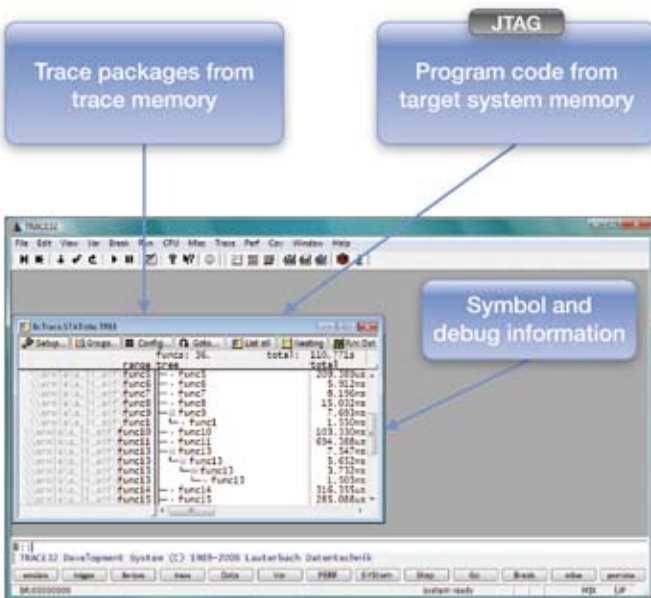


図5: シンボルレベルでプログラムフローを解析する場合、トレースパッケージ、プログラムコード及びシンボル/デバッグ情報が必要

解析

記録されたプログラムセクションを解析するには、トレースパッケージをトレースメモリからホストに転送し、圧縮解除してから評価する必要があります。

プログラムフローのトレースパッケージにはプログラムコードは含まれないため、解析する前にプログラムコードを追加する必要があります。使用されるのは以下のデータです:

- TRACE32ソフトウェアによってJTAGインタフェース上でターゲットシステムのメモリから読み取られるプログラムコード
- ユーザーによってTRACE32ソフトウェアにロードされるシンボル情報およびデバッグ情報(図5を参照)



長時間トレースは、記録中および解析中に即時にトレースパッケージをホストに転送することにより実装されます。この場合、TRACE32 トレースツールのトレースメモリは基本的に単なる FIFO として使用されます。

長時間トレースでは、作成されるデータ量も膨大になるため、トレースパッケージの解析を記録と並行して行うことをお勧めします。ファイルに保存する前にトレースパッケージを圧縮する場合でも、通常 1 時間あたりに収集されるデータ量は最大 5GB です。同時に、記録の終了後には解析にかかる時間を大幅に見込む必要があります。たとえば、プログラム実行に要する時間が 2 時間のトレースパッケージを 1 つのファイルで収集した場合、その後で従来型の解析を実行するには、処理能力の高いホストでも数時間かかります。

長時間トレースで大量のデータを短時間に記録、転送、解析する場合は、以下の条件が必要になります:

- 高速なホスト
- 高速なトレースツール
- コンパクトなデータフォーマット



図6: 長時間トレースではホストと高速なP2Pインタフェースが必要

## 高速なホスト

プログラムの実行時にホストでトレースパッケージを解析するには、高速なデュアルコアコンピュータが必要です。デュアルコアであれば、1つのコアでトレースパッケージを受け取り、もう1つのコアで並行してパッケージを評価することができます。

解析の際には、トレースパッケージだけでなくプログラムコードも必要です。ARMコアの多くはCPU実行中にターゲットシステムのメモリからコードを読み取ることができないため、長時間トレースの開始前に TRACE32 ソフトウェアにコードをコピーする必要があります。

## 高速なトレースツール

従来型のトレースで説明したように、トレースツールは高速なトレースポートで損失なくトレースパッケージをサンプリングする必要があります。そして、トレースパッケー

ジを高速でホストに転送することが長時間トレースの新たな必要条件です。その条件を達成するために、TRACE32 トレースツールにはギガビット Ethernet インタフェースが用意されています。トレースツールをピアツーピアでホストに接続すると、500MBit/s を超える転送速度を得ることができます (図6を参照)。

現在のところ、ホストへの最大転送速度が長時間トレースのボトルネックになっています。つまり、長時間トレースが機能するのは、トレースポートでの平均データ転送速度がホストへの最大転送速度を超えない場合に限りです (図7を参照)。

ハイピーク時の負荷はトレースメモリによってバッファされるため、問題にはなりません。

## コンパクトなデータフォーマット

ホストへの最大転送速度には限界があるため、データ量をできるだけコンパクトにすることが重要です。データ量は、以下の2つの要因に左右されます:

1. ETMv3 の最適なプログラミング設定
2. トレースパッケージのコンパクトなバッファ処理

### 1. ETMv3の最適なプログラミング設定

解析に関連する情報についてのみトレースパッケージを生

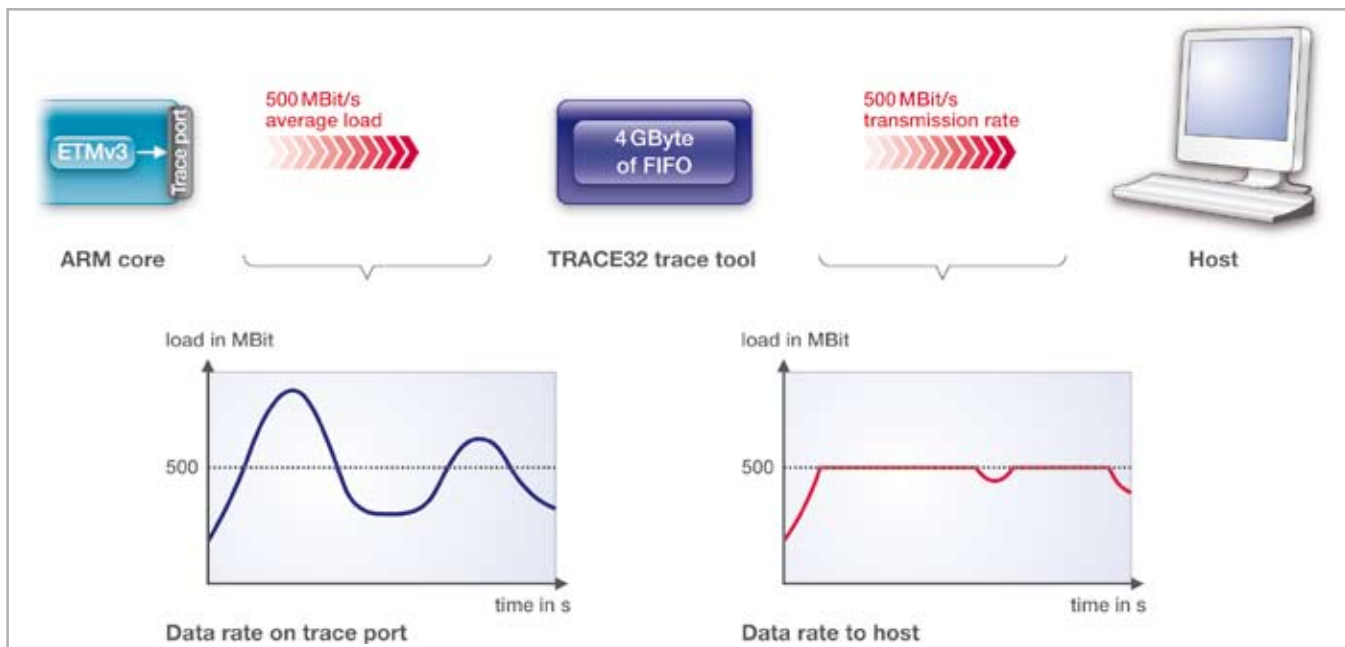


図7: 長時間トレースの動作例(トレースポートの平均負荷が500 MBit/s 以下の場合)

Software	携帯端末	浮動小数点演算	HDD コントローラ
トレース情報 /1 命令	0.8 Bit	2.2 Bit	4.3 Bit
コア	Cortex-A	ARM11	ARM9
コア動作周波数	500 MHz	300 MHz	450 MHz
トレースポート周波数 DDR	166 MHz	75 MHz	150 MHz
RTOS	Linux	—	—
トレースポートの 平均データ転送速度	340 MBit/s	406 MBit/s	798 MBit/s

成するように ETMv3 をプログラムすれば、トレースポートでのデータ転送速度に直接的に影響を与えることができます。トレースポートに対する高負荷の原因となっているデータフローパッケージは、通常プロファイリングやコードカバレッジに必要ではありません。

その他、以下のような要素もトレースポートでの平均データ転送速度に影響しますが、これらは残念ながら変更不可とお考えください：

**ARM コアの周波数：**ARM コアの周波数が高くなるほど、1秒あたりのトレースデータも多くなります。

ターゲットシステム上のソフトウェア：何度もジャンプを実行してキャッシュでデータや命令を検索するようなソフトウェアプログラムの場合、連続する多数の命令を処理してデータや命令が利用可能になるまで待機することの多いソフトウェアプログラムに比べて、1秒あたりに生成されるトレースパッケージが多くなります。本ページの表は、トレースポートでの平均データ転送速度の測定値の例を示したものです。データ転送速度は、コアで実行されるソフトウェアによる影響がきわめて大きいことがわかります。それに比べると、コアの周波数やアーキテクチャによる影響はさほど大きくありません。

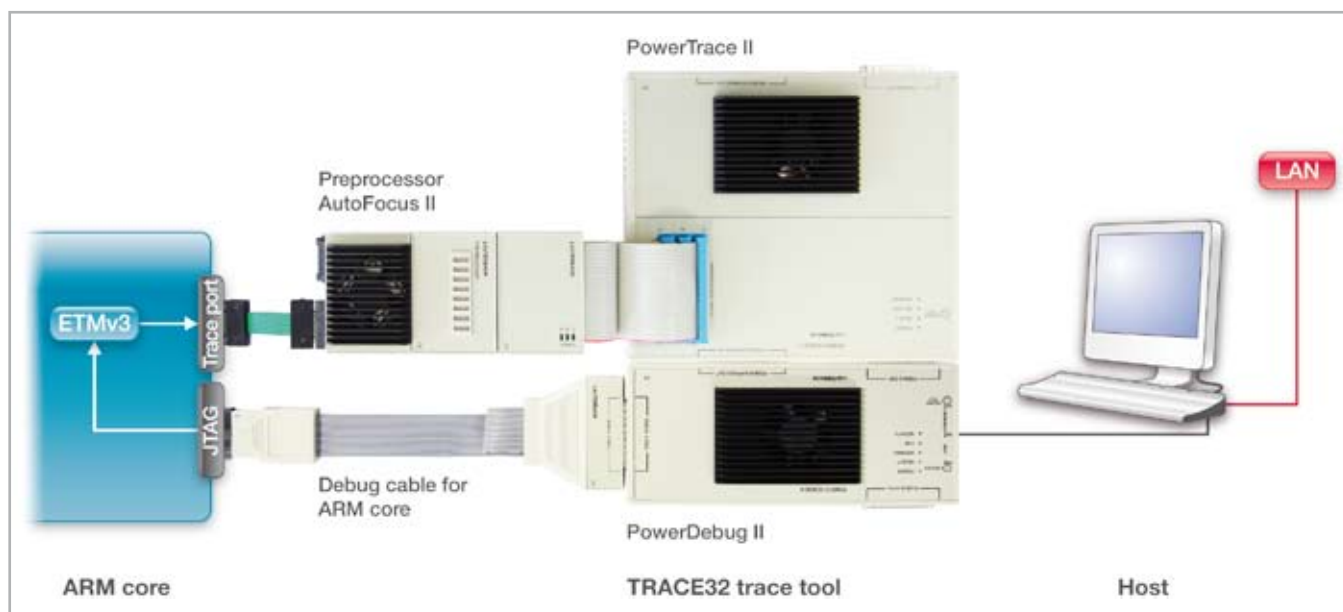


図8: ローターパツハのETMv3 長時間トレース対応ツール

## 2. コンパクトなバッファ処理

TRACE32 トレースツールのファームウェアは、トレースパッケージ出力用の 8 ピンでトレースメモリのパッケージの記録密度が最適になるように強化されています。

## まとめ

TRACE32 ソフトウェアでは、長時間トレースの設定および解析は Real-Time Streaming(RTS) という名前で実施されます。

パラレル ETMv3 の長時間トレース用のローターバッハトレースツールは、以下の TRACE32 製品で構成されます (図 8 を参照) :

PowerDebug II: ホストへのギガビット Ethernet インタフェースを提供し、トレースパッケージを転送します。

ARM コア用のデバッグケーブル: JTAG インタフェース上で ETMv3 をプログラムします。

PowerTrace II: トレースパッケージを保存します。現在の最大トレース深度は 4GB です。

プリプロセッサ AutoFocus II: パラレルトレースポートでトレースパッケージをサンプリングし、トレースメモリに転送します。

長時間トレースによって、ローターバッハはプログラム実行をほぼ無制限に解析できるトレース技術に向けて重要な一歩を踏み出しました。

- ホストマシンの高い処理能力はプログラム実行時においてのトレース情報の解析のために使われる。
- ハードディスク容量は記録時間の大きく延ばすために使われる。

ホストプロセッサ能力とハードディスク容量は、何れも数年後の間に増大していると予想されますので、さらにより複雑な解析機能を実現できることを期待しています。

## ETMv3 長時間トレース

# コードカバレッジ解析と長時間トレース

長時間トレースの用途の一つとして、システムテスト中にすべてのプログラムコードが処理されるかどうかのチェックする機能があります。TRACE32 ソフトウェアでは、このコードカバレッジ解析を提供しています。

コードカバレッジ解析のために、TRACE32 ソフトウェアでは、全関数リストとそれらのコードカバレッジを提供しています。加えて、条件命令実行の統計サマリも表示可能です。(図 9 を参照)

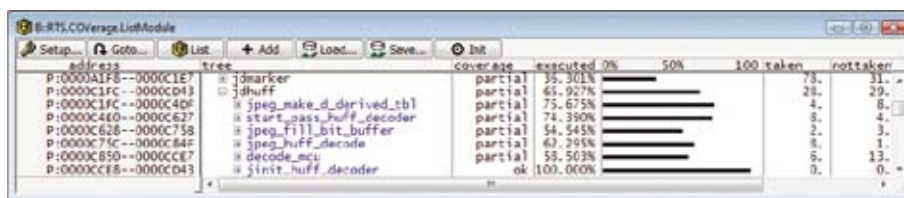


図9: 関数とモジュールのコードカバレッジ表示リスト

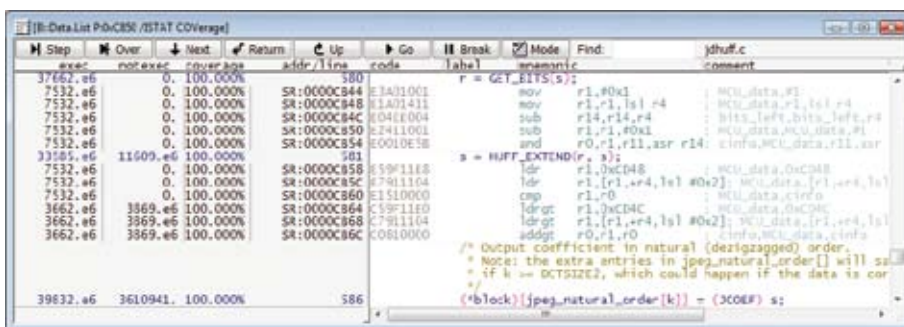


図10: 詳細な解析結果を表示(何回コマンドが実行されたのか(exec), スキップされたのか(notexec))

関数名をダブルクリックすると、関数の詳細情報が表示されます。未実行のコマンドは色付けされ、リニアなコードにおいては、テスト期間中にコマンドが実行された頻度を見ることができます。条件命令では、条件が満たされなかったために何度コマンドがスキップされたのかも見ることができます。(図 10 を参照)

従来型トレース機能と数秒間の観測時間では、ほんの短いプログラムセクションしか解析できませんでした。ETMv3 コアでは、長時間トレースの導入により、効率的でかつ、CPU の実行状態に影響を与えることのないコードカバレッジ解析が初めて実現されました。

ETMv3 長時間トレース

プロファイリングと長時間トレース

時間制約のある関数の場合、最大時間が定義されることも多く、システムテストでチェックする必要があります。長時間トレースの機能を使えばこのチェックを簡単に行うことができ、タイムアウトをすぐに検出することができます。

address	time	coverage	count	time	cycles	ratio	rpi
P:0000018-000001F3	61.705%	-	27.745s	4281.e6	0.018%	6.66	
P:000001F5-000001E7	35.301%	-	150.735s	33569.e6	0.402%	3.85	
P:000001FC-00000D43	65.927%	-	5.671ks	998.e9	11.977%	1.91	
P:0000021C-0000040F	75.675%	7221882	492.328s	86650.e6	1.019%	2.64	
P:0000040-00000E27	74.390%	1103847	9.763s	1715.e6	0.020%	5.37	
P:00000628-00000758	54.545%	1024.e6	82.108s	151786.e6	1.853%	1.58	
P:0000075c-0000084F	62.295%	144.e6	87.329s	15169.e6	0.184%	2.96	
P:00000850-00000c77	55.503%	180.e6	4.195ks	734270.e6	8.855%	1.91	
P:00000C18-00000D43	100.000%	1103647	4.308s	755.e6	0.009%	15.4	
P:00000D50-00000A3F	0.000%	-	0.000us	0.	0.000%	0.00	
P:00000480-00000139	80.000%	-	131.902s	23214.e6	0.278%	3.17	
P:00000140-00000D9F	19.410%	-	680.222s	119719.e6	1.436%	2.03	
P:00000800-0000040F	11.567%	-	4.497s	791.e6	0.009%	14.3	
P:0000040-00000E27	65.927%	-	51.015s	8581.e6	0.107%	5.45	
P:00000644-00000E93	100.000%	-	15.325ks	2697.e9	32.368%	1.69	

図11: モジュールおよび関数の時間挙動解析

H Step	Over	Time	Ratio	addr/line	source
23405564	265.573s	0.561%	181		if (inptr[DCTSIZE*1] == 0 && inptr[DCTSIZE*2] == 0 &&
18724760	224.138s	0.473%	183		inptr[DCTSIZE*3] == 0 && inptr[DCTSIZE*4] == 0 &&
20278559	0.503%	183		inptr[DCTSIZE*5] == 0 && inptr[DCTSIZE*6] == 0 &&	
16204910	184.146s	0.388%	184		inptr[DCTSIZE*7] == 0) {
17511830	203.543s	0.423%	186		/* AC terms all zero */
4471747	50.815s	0.107%	189		int dcvail = DEQUANTIZE(inptr[DCTSIZE*0], quantptr[DCTSIZE*0]) <<
4471747	50.815s	0.107%	190		wsprtr[DCTSIZE*0] = dcvail;
4471747	50.815s	0.107%	191		wsprtr[DCTSIZE*1] = dcvail;
4471747	50.815s	0.107%	192		wsprtr[DCTSIZE*2] = dcvail;
4471747	50.815s	0.107%	193		wsprtr[DCTSIZE*3] = dcvail;
4471747	50.815s	0.107%	194		wsprtr[DCTSIZE*4] = dcvail;

図12: プログラムラインごとの時間挙動の詳細

tree	total	avr	max	internal	external	internal%	%
decompress_onepass	21.516ks	1.318us	2.125us	611.197s	20.845ks	1.417%	
jpeg_fill_bit_buffer	257.176s	1.312us	22.727us	237.176s	-	0.500%	
jpeg_huff_decode	6.281ks	79.250us	147.727us	4.191ks	1.086ks	8.819%	
jpeg_huff_encode	996.030s	0.491us	125.000us	878.971s	117.059s	1.858%	
jpeg_huff_encode	117.059s	87.232us	133.636us	3.242s	113.817s	0.005%	
jpeg_huff_decode	113.817s	94.540us	113.856us	113.817s	-	0.240%	
jpeg_huff_encode	90.662s	0.627us	11.364us	87.325s	3.337s	0.184%	
jpeg_fill_bit_buffer	3.337s	0.552us	11.364us	3.337s	-	0.007%	
start_acu_rune	1.529s	1.529s	1.529s	1.529s	-	0.003%	
findtbl_input_pass	184.658ms	184.658ms	184.658ms	184.658ms	-	<0.001%	

```

record_run_address  cycle  data  symbol
+4498814986039
  copy  r11,pc
  ds  R:0000E64  gtrace  \\jpeg\jidctint\jpeg_idct_islow
  * Perform dequantization and inverse DCT on one block of coefficients.
  (void)
  jpeg_idct_islow(j_decompress_ptr cinfo, jpeg_component_info * comptr,
  SCOEPRR coef_block,
  JSAMPARRAY output_buf, JDIMENSION output_col)
  151
  * stndb: r11, [r0+r11,r13]
  INT32 tmp0, tmp1, tmp2, tmp3;
  INT32 tmp10, tmp11, tmp12, tmp13;
  INT32 z1, z2, z3, z4, z5;
  COEPRR inptr;
  ISLOW_MULT_TYPE * quantptr;
  int * wsptr;
  
```

図13: 必要に応じて、最長実行関数をすぐに表示

最初に行うのは、時間制約のある関数が最大時間を超えたかどうかのチェックです。そのためには、プログラムフローとコンテキスト ID に関するトレースパッケージのみを生成するように ETMv3 をプログラムする必要があります。これには、2つの理由があります:

1. このようにプログラムすると、トレースポートでのデータ転送速度を可能な限り低く抑えられる。
2. ネストした関数の正確な解析を妨げる FIFO のオーバーフローを防止できる。

長時間トレースを開始すると、TRACE32 ソフトウェアは関数の時間挙動を解析します。解析されるのは、完全テスト実行中のクロックサイクルの実行時間と回数、全体の実行時間に対して関数が占める割合、および「1命令あたりのクロック数」の平均です (図 11 を参照)。

各関数の詳細な解析には、プログラム行の時間挙動も表示されます (図 12 を参照)。

トラブルシューティング

解析結果により、定義した最大時間を一度でも超えるようであれば、その原因はつきとめなければなりません。

長時間トレースでは、プログラム実行時にトレースパッケージをファイルに保存するように設定することが可能です。5GB/h のデータ量の場合、平均的なハードディスクに約 4 日分のプログラム実行を記録することができます。

このトレース記録に対して、TRACE32 ソフトウェアの高速かつ高性能な検索機能を使用すれば、過剰な関数実行を検出し、詳細な解析結果を表示することができます (図 13 を参照)。

プログラムフローだけではタイムアウトの原因を見つけられない場合、従来型のトレースに戻り、プログラムフローだけでなく、全データフローも生成するように ETMv3 を設定することができます。



## 新サポートプロセッサ

### New Derivates

<b>Andes Technology</b>	<b>LA-3756 (ANDES)</b> • N9/N10/N12
<b>ARM</b>	<b>LA-7843 (Cortex-A)</b> • Cortex-A9 Single Core • Cortex-A9 MPCore
<b>Broadcom</b>	<b>LA-7760 (MIPS32)</b> • BCM3556 • BCM7325 • BCM471X  <b>LA-7761 (MIPS64)</b> • BCM1280/BCM1480
<b>CEVA</b>	<b>LA-3711 (CEVA-X)</b> • CEVA-X1641  <b>LA-3774 (TeakLite-III)</b> • CEVA-TL3210
<b>Freescale</b>	<b>LA-7732 (ColdFire)</b> • MCF5227x/MCF525x  <b>LA-7735 (DSP56300)</b> • DSP56720  <b>LA-7733 (MCS08)</b> • MC9S08ACx/DVx  <b>LA-7734 (MPC5200)</b> • MPC5121/MPC5123  <b>LA-7753 (MPC55xx)</b> • MPC560xx • MPC5633M • MPC5668 • MPC5674  <b>LA-7764 (PowerQUICC III)</b> • QorIQ  <b>LA-7736 (MCS12X)</b> • S12P • S12XF • S12XHZ • S12XS
<b>Infineon</b>	<b>LA-7759 (C166S V2)</b> • XC2267M-104F • XC2287M-104F • XC2387M-104F  <b>LA-7756 (TriCore)</b> • TC1736/TC1736ED • TC1767/TC1767ED • TC1797/TC1797ED

<b>Luminary Micro</b>	<b>LA-7844 (Cortex-M)</b> • LM3S3700 Series • LM3S5600 Series • LM3S5700 Series
<b>Marvell</b>	<b>LA-7742 (ARM9)</b> • 88F5082 • 88F5180N • 88F6082 • 88F6180 • 80F6281
<b>Microchip</b>	<b>LA-7760 (MIPS32)</b> • PIC32
<b>Micronas</b>	<b>LA-7760 (MIPS32)</b> • VCTH
<b>MIPS</b>	<b>LA-7760 (MIPS32)</b> • MIPS74
<b>NEC</b>	<b>LA-7765 (ARM11)</b> • NaviEngine  <b>LA-7835 (V850)</b> • V850E • V850DX3
<b>NXP</b>	<b>LA-7742 (ARM9)</b> • LPC29xx  <b>LA-7844 (Cortex-M)</b> • LPC17xx
<b>Renesas</b>	<b>LA-7758 (SH)</b> • SH4A-Multi • SH7786 • SH7722 • SH7723 • SH7763
<b>STMicro-electronics</b>	<b>LA-7844 (Cortex-M)</b> • STM32F102  <b>LA-7836 (MMDSP)</b> • Nomadik STn8820  <b>LA-7753 (MPC55xx)</b> • SPC560x • SPC563M
<b>Tensilica</b>	<b>LA-3760 (XTensa)</b> • Xtensa 7

## シリアルギガビットトレースインタフェース

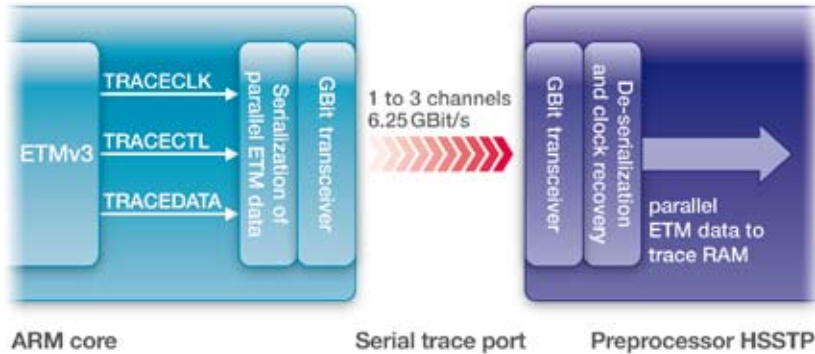


図14: ARM ETM対応HSSTPのブロック図



図15: HSSTP用プリプロセッサ

シリアルトレースインタフェースは以下の2つの問題を同時に解決します:

1. シリアル転送に必要なピン数が少なくなる。
2. ディファレンシャル伝送によってデータ転送が高速化する。

シリアル転送の性能が実際にいかに高いかを示す例として、3つのトレースチャンネルのみを使用して、DVDの内容すべてを3秒足らずで転送できることが挙げられます。

ローターバツハでは、このコンセプトの利点を確信し、2007年には技術上の試みとして意欲的に高速シリアルトレースのプロジェクトに着手しました。シリアルトレースは実用化され、2008年の中頃からお客様に提供されています。

現在、ローターバツハはARM社のHigh Speed Serial Trace Port (HSSTP) をサポートしています。また、Freescale社のQorIQ (e500 Power Architecture) 用の高速トレースポートは、すでに開発の計画段階に入っています。

プリプロセッサHSSTP(図15を参照)は、最大4つの高速チャンネルを想定して設計されています。サポートされる転送速度は以下の通りです:

- 3チャンネルまでの場合、チャンネルあたり6.25Gbit/s
- 4チャンネルの場合、チャンネルあたり3.125Gbit/s

トレースデータは、Samtec社のカスタムコネクタシステム(ERF8、40ピン)を介して提供されます。

ARM-HSSTPは、転送にXilinx社のAuroraプロトコルを使用します。パラレルトレースデータは8ビットまたは10ビットでコード化され、ARMコアでシリアル化されます。ディファレンシャルギガビットトランシーバがケーブルでローターバツハのプリプロセッサHSSTPにデータフローを送信し、プリプロセッサがシリアル転送から元のパラレルトレ

ースデータを復元します(図14を参照)。トレースデータが大量になれば、それに応じて大容量のトレースメモリが必要になることは明らかです。これを可能にするのが、4GBまでメモリを拡張できるPowerTrace IIです。

### パラレルトレースインタフェース

2008年、パラレルトレースインタフェースのためのサポートを、主にDSPなどの数種類のプロセッサアーキテクチャに対し拡張しました。次のテーブルがその一覧になります。この機能拡張は、ジョイントトレースポートを備えるヘテロジニアスマルチコアプロセッサをサポートするために必要な対応でした。12ページの"CoreSight トレース"の項にて、更なる情報を参照できます。

パラレルトレースインタフェース対応 Preprocessor AutoFocus II
Preprocessor AutoFocus II for ARM ETM
Preprocessor AutoFocus II for CEVA-X
Preprocessor AutoFocus II for MicroBlaze
Preprocessor AutoFocus II for PPC4xx
Preprocessor AutoFocus II for SHx
Preprocessor AutoFocus II for StarCore
Preprocessor AutoFocus II for TeakLite-III
Preprocessor AutoFocus II for TMS320C55x
Preprocessor AutoFocus II for TMS320C64x+

## ARM CoreSightによるデバッグ

ARM CoreSight は、異種混在マルチコアプロセッサを対象にしたデバッグおよびトレースのコンセプトの好例です。

組み込みシステムで多数のタスクを処理するために、複数のコアタイプを持つプロセッサの利用が広がっています。そうしたシステムを適切にデバッグするには、以下の2つの条件を満たす必要があります：

1. マルチコアプロセッサが、適切なオンチップデバッグおよびトレースロジックを持っていること。
2. 開発環境が、各コアのデバッグをサポートするだけでなく、インテリジェントなテストおよび解析機能によってシステム全体のデバッグもサポートしていること。

このパートでは、CoreSight のオンチップデバッグおよびトレース技術を組み合わせた TRACE32 の開発環境が、これらの必要条件をどのように満たしているかという点について説明します。

### CoreSightの概要

CoreSight は、ARM 社がマルチコアプロセッサ用に提供しているオンチップデバッグおよびトレース技術の名称です。ただし、CoreSight は固定されたロジックブロックとして設計されているのではなく、むしろ組み立てキットのように多種多様なコンポーネントを提供します。そのため、マルチコアプロセッサの設計者はデバッグとトレースのために用意する関数の範囲を定義することが可能です。CoreSight は、設定の自由度がきわめて高くなっています。通常、適切なデバッグおよびトレースオプションをプロセッサに統合するには、ツールメーカーの専門知識が必要です。ローターバハの技術者は、最新世代プロセッサの設計段階において、長年にわたって全世界の開発者にこの課題に関する助言を行ってきました。

CoreSight における組み立てキット式のコンセプトが、利用される開発ツールにも影響することは言うまでもありません。プロセッサとその CoreSight コンポーネントの設定がツールで認識されていれば、デバッグは非常に簡単に行うことができます。ただし、新たなプロセッサの場合、組み立て式のコンセプトではツールに高度な柔軟性が求められます。CoreSight の設定情報はプロセッサから読み取ることが可能ですが、プロセッサの設計者が行った実装の詳細を明確にしなければならない場合も少なくありません。

ここでは、以下のような、RISC コアの ARM11、Cortex-A、Ceva-X DSP で構成される異種混在マルチコアプロセッサの例で解説します。

- RISC core ARM11
- RISC core Cortex-A
- Ceva-X DSP

### CoreSight デバッグ

CoreSight を利用したプロセッサの場合、すべてのコアはジョイント JTAG インタフェース上でデバッグされます。代表的なプロセッサの開発環境は、以下の TRACE32 製品で構成されます（図 16 を参照）：

- USB または Ethernet インタフェース経由でホストに接続したユニバーサル PowerDebug モジュール
- ARM11、Cortex-A、および Ceva-X の各アーキテクチャのライセンスがあるデバッグケーブル

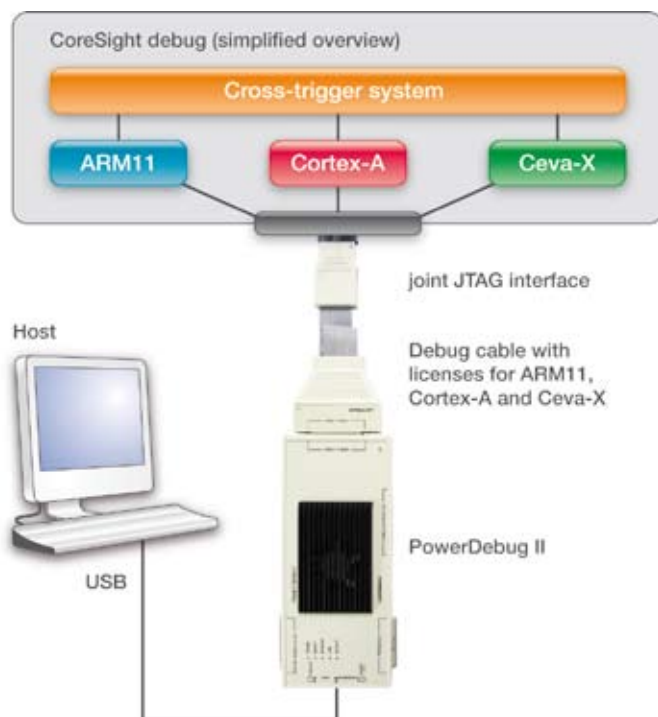


図16: CoreSight デバッグ対応TRACE32開発環境

異種混在マルチコアプロセッサでは、各コアは通常相互に独立してそれぞれのタスクを実行します。したがって、個別に TRACE32 インスタンスを起動して各コアをデバッグすることも有効です（次ページの図 17 を参照）。

しかし、コアが連携して正常に動作していることをテストするためには、コアにまたがってデバッグを実行する必要があります。そのために、CoreSight にはすべてのコアの同期デバッグが可能なクロストリガーシステムが用意されています。このシステムでは、コアがブレークポイントで停止すると、他のコアも同期して停止します。つまり、ユー

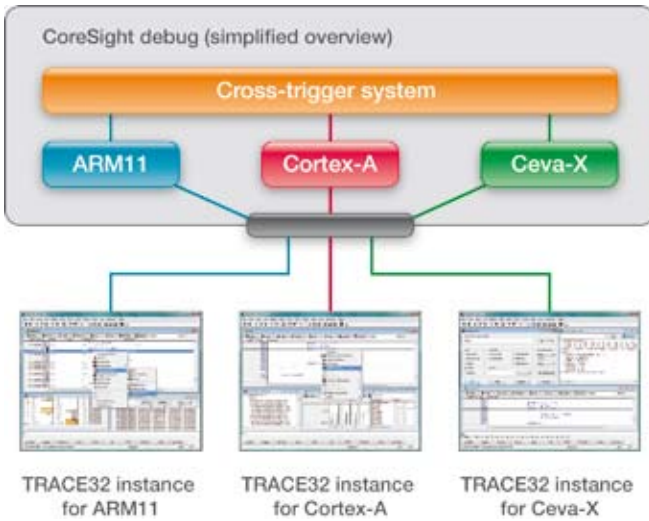


図17: 各TRACE32インスタンスが対応するコアのデバッグを開始

ザーはプログラム中の任意の場所で各コアのコンテキストを簡単に視覚化できるということです。

マルチコアデバッグに対応したこの基本機能の他にも、TRACE32にはCoreSightの設定を利用する便利なデバッグ機能が複数あります。右の囲み記事にCoreSightデバッグ用のTRACE32の全機能を紹介しています。

## CoreSight トレース

すべてのコアからのトレース情報には、共通のインターフェースも用意されています。CoreSightでは、トレース情報を生成するコンポーネントを各コアに割り当てることができます。代表的なプロセッサの場合、以下のコンポーネントがそれに該当します：

- ARM11 および Cortex-A 用の ARM ETM
- Ceva-X 用の Ceva-X ETM (図 18 も参照)

どのトレースコンポーネントも、コアが実行した命令と処理されたデータアクセスについての情報を生成します。このトレース情報をジョイントインターフェースで提供するため、Funnelによってトレースデータが単一のデータストリームに組み入れられます。これがトレースポートで出力されるか、またはオンチップトレースメモリに保存されます。

## オフチップトレースポート

18本のプロセッサピン(実際のトレース情報に16ピン、制御信号に2ピン)を使用して、すべてのコアのトレースデータを外部トレースツールに出力できます。TRACE32によるオフチップ記録および解析では、図16に示した開発環境に

## CoreSightデバッグ対応 TRACE32の主要機能

- CoreSightでマルチコアプロセッサに柔軟に対応；ARM/Cortex用だけではなく、DSP用のデバッグも幅広くサポート
- シリアル・ワイヤ・デバッグポートおよびJTAGポート接続でデバッグ
- プログラム実行中の物理メモリおよび周辺レジスタへのアクセス
- 全コアおよび周辺機能との同期デバッグ
- あるコアが省電力状態にあっても、他のコアでは影響なくデバッグ可能

以下の製品を追加する必要があります(図19を参照)：

- 4GBまでのトレースメモリに対応するユニバーサルPowerTrace IIモジュール。
- トレースポートでトレースデータにアクセスするプリプロセッサAutoFocus II。この場合、プリプロセッサAutoFocus IIにはARM ETMとCeva-X ETMのトレースライセンスが必要。

»

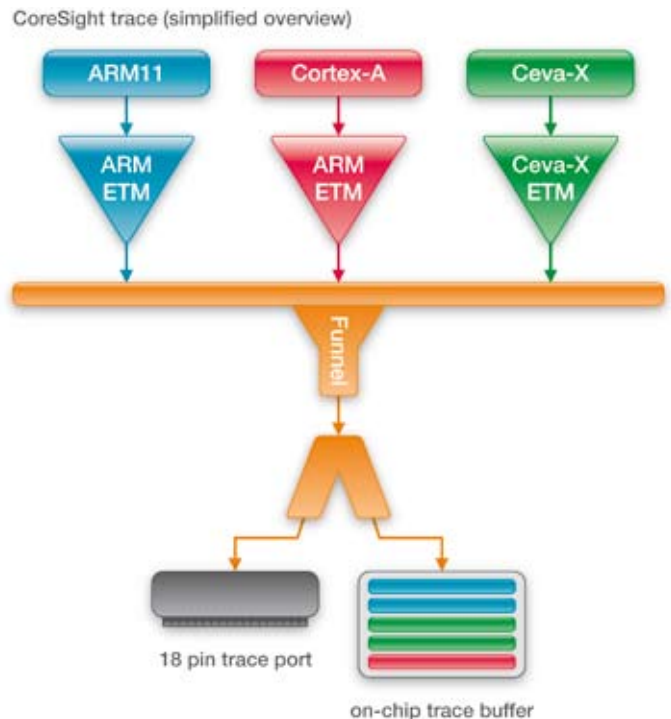


図18: 各コアはそれぞれ独自のトレース情報を生成

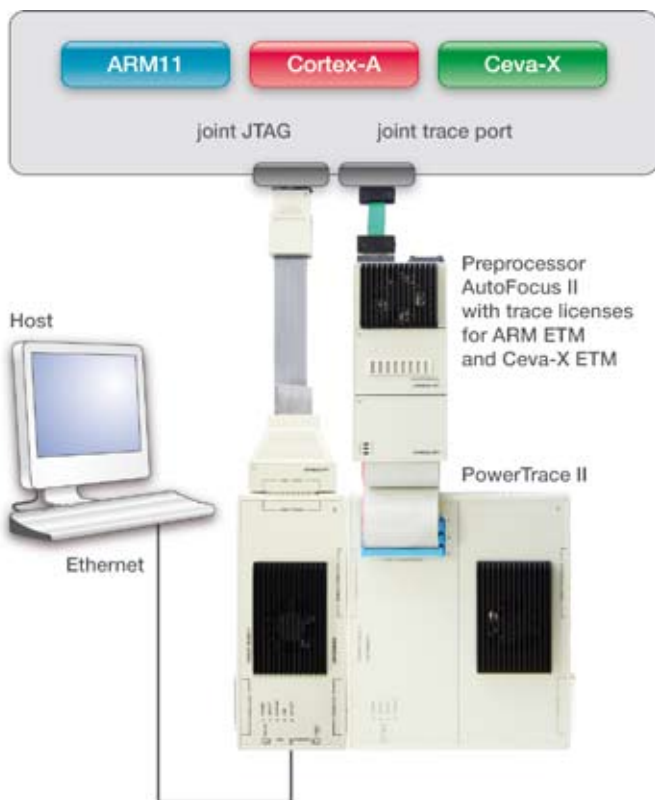


図19: CoreSightデバッグおよびCoreSightトレース対応TRACE32開発環境

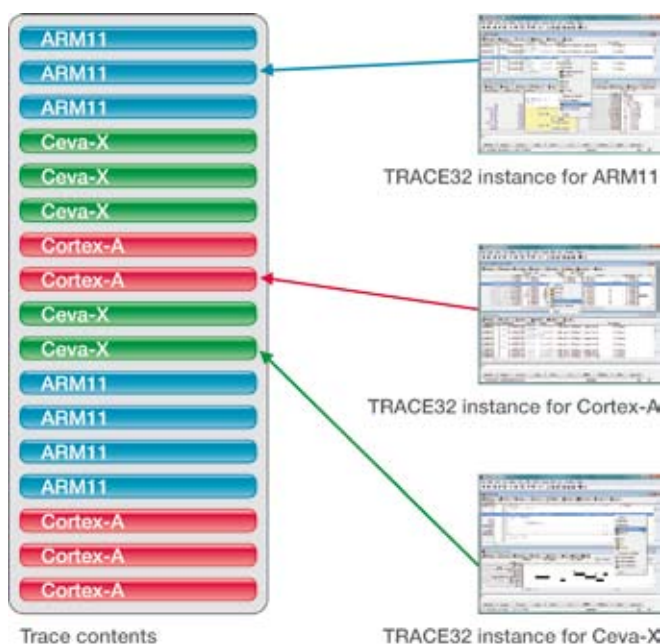


図20: 各TRACE32インスタンスで各コアのトレースデータを表示

## オンチップトレースメモリETB

トレースポートのピンを節約する別の方法として、CoreSight Embedded Trace Buffer (ETB) として知られるオンチップトレースメモリが挙げられます。ただし、その容量は外部トレースツールよりもはるかに小さく、通常は2~8KBにすぎません。

トレースデータをETBに保存してからJTAGインタフェース上で読み取る場合、図13に示したデバッグケーブルには、ARM ETBとCeva-X ETBのトレースライセンスも必要になります。

## トレース解析

記録が終了すると、開発者は個々のコアのトレースデータを表示して解析することができます。そのために、各TRACE32インスタンスはトレースデータを共通のトレースメモリから読み取ります。(図20を参照)

コアの相互動作を解析するために、トレースのディスプレイは、時間と直接関連付けて全コアのトレースエントリを表示するように設定することができます。たとえば、トレースエントリをARM11のインスタンスで選択する場合、残り2つのTRACE32インスタンスはその時点でコアによって実行された命令をマークします。

デバッグオプションの場合と同様に、TRACE32で利用可能なトレースオプションも現在のCoreSightの設定によって異なります。トレースオプションによって計画的なトラブルシューティングが容易になり、全体的なシステム性能の解析が可能になります。以下の囲み記事に各トレース機能を紹介しています。

### CoreSightトレース時のTRACE32の機能

- CoreSightによるマルチコアプロセッサの柔軟なサポート。TRACE32は、ARM ETMと各種DSP ETMのトレース情報の解析に対応する。
- AMBA AHBバスのバスサイクルのトレース。
- Instrumentation Trace Macrocell (ITM) を利用した、アプリケーションのデータ出力のトレース。
- トレースポートまたはオンチップトレースメモリのストレージでのトレースデータの出力。
- クロストリガーシステムを通じた、トレースデータ生成コンポーネントの相互有効化。
- コアごとのトレースデータの時間相関での視覚化。
- コードカバレッジおよび総合的な実行時の解析。

## 256メガレコード ロジックアナライザ



図21: PowerIntegrator II ロジックアナライザ

PowerIntegrator II は、ローターバッチのロジックアナライザでさらにもっと長い時間の記録を取りたいというユーザーから頻りに寄せられていたリクエストに応えた製品です。最大 4 GByte のメモリを装備し、102 チャンネル / 256M レコードもの容量を利用できます。使用チャンネル数を少なくし 1G レコード容量として使用できるようにするオプション機能の計画も進んでいます。

今、ローターバッチのロジックアナライザ製品シリーズは、PowerProbe, PowerIntegrator, および新 PowerIntegrator II の3つからなります。何れのロジックアナライザもトレースレコード数、入力チャンネルおよび接続方法が異なります。(詳細はテーブルを参照)

すべてのデバイスに記録制御用のトリガーユニットが装備されています。3つのカウンタと4つのトリガーレベルがあるため、複雑なトリガー条件の定義が可能です。もちろん、ローターバッチのデバッグまたはリアルタイムトレースツールによるクロストリガー処理もサポートしています。これにより、定義されたトリガー条件を検出したデバイスがどれであるかに関係なく、すべてを同期した状態で開発環境全体を停止することも簡単です。

すべてのロジックアナライザでは、以下の機能が利用可能です:

- 記録した生データをプロトコル解析用にフォーマット変換

(CAN, FlexRay, LIN).

- 複数のチャンネルをメモリバスを形成するように結合可能。より理解し易くするために、デバッグのシンボル情報でシンボルを使ってメモリアドレスを表示。
- 電流、電圧測定によりアプリケーションのエネルギー消費状況を解析。
- スティミュリジェネレータ機能でターゲットシステム用のテスト信号を生成。

ロジックアナライザを含む典型的なローターバッチ開発環境の構成は以下になります:

- 高速で洗練されたデバッグ
- プログラム / データフローのリアルタイムトレースツール
- アプリケーションに関連のあるデジタル / アナログ信号を記録するロジックアナライザ

リアルタイムトレースとロジックアナライザの記録は、時間相関させることも可能です。これにより、アプリケーションのあらゆる詳細を一目でチェックすることができます。

新製品の PowerIntegrator II は、長時間の記録が必要な場面での使用を想定して設計されています。代表的な用途は、シリアルプロトコルの解析です。PowerIntegrator II のホストには大量のデータを転送しなければならないため、PowerDebug II のギガビット Ethernet インタフェースを使用する必要があります。

	PowerProbe	PowerIntegrator	PowerIntegrator II
トレース深度	256 K records	512 K records	256 000 K records
チャンネル数	64/32/16	204/102	102/51
タイミングモードサンプリング	100/200/400 MHz	250/500 MHz	250/500 MHz
ステートモードサンプリング	100 MHz	200/400 MHz	200/400 MHz
接続	Clip set	Mictor, Samtec, clip set, standard header	Mictor, Samtec, clip set, standard header
その他	Stimuli generator, FPGA trace	—	Stimuli generator

## RTOSデバッグの最新動向

TRACE32 デバッグ環境には設定変更可能な RTOS デバッガが含まれており、リアルタイムオペレーティングシステムでのシンボリックデバッグ機能を利用することが可能です。代表的なオペレーティングシステムにはすべてロータリーバツハの標準デバッグ環境で対応済みであり、追加コストは不要です。

今年にはオペレーティングシステムに関して多くのニュースがありますが、2008 年の最も重大なイノベーションが組み込み設計における SMP オペレーティングシステムの利用増だったことは間違いありません。サポートされるマルチコアプロセッサのリストについては、左下の表を参照してください。

毎年多くの新しいオペレーティングシステムが市場に登場し、既存のオペレーティングシステムも更新されています。これについては、右の表を参照してください。

### Linux: 実行/停止モードデバッグ

以下の機能も利用可能になりました。

- TRACE32 GUI での JTAG と GDB のデバッグ切り替えが、SH アーキテクチャでもサポートされるようになりました。
- ARM アーキテクチャでは、デバッグ通信チャンネル (DCC) も使用して、GDB デバッグと Linux のターミナルウィンドウを同時に使用できるようになりました。

SMP オペレーティングシステムサポート	
Linux, QNX, Symbian, ThreadX	
For the Following Processors	
ARM11 MPCore	ARM
Cortex-A9 MPCore	ARM
MIPS34K	MIPS Technologies
MPC8572	Freescale
MPC8641D	Freescale
SH7786	Renesas

### Linux: ページブレイクポイント

TRACE32 ソフトウェアと適切な Linux パッチを使用すると、まだロードされていないプログラムコードにソフトウェアブレイクポイントを設定することができます (ページブレイクポイント)。ページブレイクポイントは、現在 ARM および MIPS アーキテクチャでご利用いただけます。

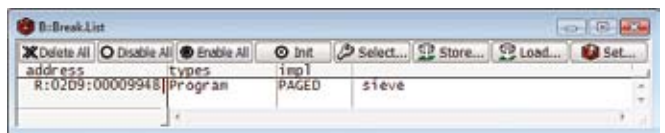


図22: ページブレイクポイント

### NetBSD: ライブラリサポート

TRACE32 RTOS デバッガが NetBSD オペレーティングシステムにも拡張され、プロセスのデバッグだけでなく、ライブラリ関数のデバッグも可能になりました。

### 新サポートRTOS

ARTX-166 for C166	available
Linux for Andes, ARC and MicroBlaze	available
LynxOS 4.0 for PowerPC	available
LynxOS 5.0 for PowerPC	planned
LynxOS-SE for PowerPC	planned
Nucleus for Andes and MicroBlaze	available
OKL4 for ARM	planned
QNX 6.4 for ARM, PowerPC, SH and XScale	planned
RTEMS for ColdFire	available
RTX-ARM for ARM	available
ThreadX for Xtensa	available
Windows CE 6.0 for MIPS	available
Xikernel for PowerPC	planned
µClinux for Blackfin	available

## STM32F1xx対応CombiProbe



図23: POWER DEBUG INTERFACE/USB 2とCombiProbe

CombiProbe は、ETM と ITM を使用する ST Microelectronics 社の STM32F1xx プロセッサにとって理想的な開発ツールです。これにより、従来は高性能な ARM プロセッサでしか利用できなかったデバッグおよびトレースのオプションが、低価格でコンパクトサイズの Cortex-M3 でも使用できるようになりました。

中速で動作する、トレースポート帯域の狭いコア用に開発された CombiProbe は特製デバッグケーブルと 128MB トレースメモリを備えています。通常、CombiProbe はユニバーサルな PowerDebug モジュールに接続して使用します（図 23 を参照）。

### CombiProbe

<b>Luminary Micro</b>	Stellaris processors with ITM
<b>Microchip</b>	PIC32 with 4-bit IFLow-Trace (program flow)
<b>NXP</b>	LPC17xx with 4-bit ETMv3 (program flow) and ITM
<b>STMicroelectronics</b>	STM32F1xx with 4-bit ETMv3 (program flow) and ITM

STM32F1xx プロセッサでは、CombiProbe は次の機能が利用できます：

- 標準的な JTAG デバッグポートおよびシリアル・ワイヤ・デバッグポートでデバッグ
- ETMv3 でプログラムフロートレース
- ITM で指定されたデータアクセスあるいは、特定アプリケーションのデータをトレース

CombiProbe の設定によって、記録中にトレースデータをホストに転送することができます。このデータは、リアルタイムで処理することもディスクに保存することも可能です。上記の表は、CombiProbe でサポートされる追加のプロセッサアーキテクチャを一覧化したものです。

### 世界の支社

#### Germany

Lauterbach GmbH  
Phone +49 8102 9876 0  
info@lauterbach.com  
www.lauterbach.de

#### USA East

Lauterbach Inc.  
Phone +1 508 303 6812  
info\_us@lauterbach.com  
www.lauterbach.com

#### USA West

Lauterbach Inc.  
Phone +1 503 524 2222  
info\_us@lauterbach.com  
www.lauterbach.com

#### France

Lauterbach S.A.R.L.  
info\_fr@lauterbach.fr  
www.lauterbach.fr

#### UK

Lauterbach Ltd.  
Phone +44 1256 333 690  
info\_uk@lauterbach.com  
www.lauterbach.co.uk

#### Italy

Lauterbach Srl  
Phone +39 02 45490282  
info\_it@lauterbach.com  
www.lauterbach.it

#### China

Suzhou Lauterbach Technologies Co. Ltd.  
Phone +86 512 6265 8030  
info\_cn@lauterbach.com  
www.lauterbach.cn

#### Japan

Lauterbach Japan Ltd.  
Phone +81 45 477 4511  
info@lauterbach.co.jp  
www.lauterbach.co.jp

**And represented by competent partners in all other countries!**

I.P.