Cortex™-M用インテリジェントデバッグ＆トレース

μTraceは、組込み分野で急速に普及が拡大しているCortex-Mプロセッサに対応するオールインワンデバッグ＆トレースソリューションです。Cortex-Mファミリ専用モジュールとすることで高性能でかつ低価格を実現しています。

ETMとITMのトレースデータを結合

Cortex™-M3/M4プロセッサでは、トレース情報は２つの異なるソースから生成されます。(図3参照)
ETMv3　は実行された命令の情報を生成し、ITM　はData-Watchpoint and Trace Unit (DWT)から得られたread/writeアクセス結果を生成します。

read/writeアクセスのITMトレースパケットには次の情報が含まれます：データアドレス、データ値およびプログラムカウンタ。プログラムカウンタを解析することで、これらの隔てられたトレース結果を、シームレスにインストラクションフロー結果に統合することが可能となります(図1参照)。

このことは、エラー発生位置の素早い特定につながります。もし、ライトアクセスがインストラクションフロー結果全体に埋め込まれていれば、あるアドレスに不正なデータをWriteするようなエラーによる原因を特定しやすくなります。

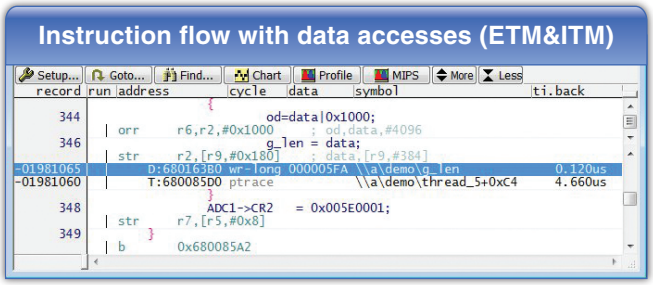

図1: ETM とITMトレースデータを結合することで、Read/Writeアクセス結果をインストラクションフローにシームレスに統合

OS認識トレース

Cortex™-M3/M4ターゲットにおいてオペレーティングシステムが動作している場合、タスクスイッチ情報がトレース解析上、必要不可欠なものになります。タスクスイッチの情報を含めるため、次のような手法がとられています：　ITMは、OS上の識別変数に対しその時点のカレントタスクの識別子データを書き込むWriteサイクルのトレース情報を生成することができます。先の記載の通り、ライトアクセス情報をシームレスに命令フロートレース情報へ統合することができるため、トレースリストの読みやすさが向上します(図2を参照)。 また、インストラクションフローへのタスクスイッチの統合化は、実行時間解析のためのベースを形成します。

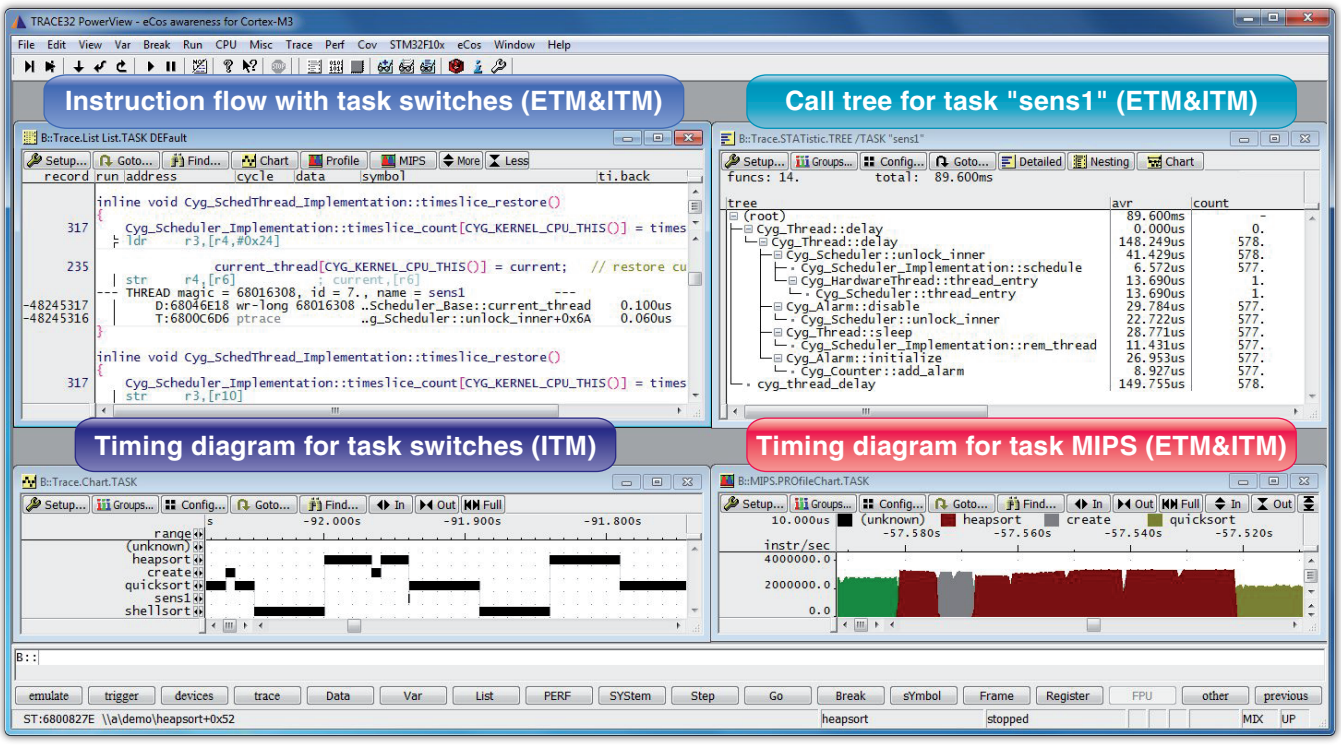

図2: ETMとITMトレースデータの結合を通して、RTOS環境下での広範なトレース解析機能が提供される

LAUTERBACH
DEVELOPMENT TOOLS

3つのレコーディングモード

Cortex™-M3/M4 プロセッサから生成されたトレース情報を記録するため、μTraceは3つのモードを用意しています：

- **FIFO モード**
  TRACE32　μTrace上の256MByteトレースメモリに先入れ先出しでトレース情報を蓄積

- **STREAM モード**
  USB3を通して最大100MBytes/sのスピードでトレース情報をホストパソコンのハードディスクに保存

- **リアルタイムプロファイリング**
  トレース情報はホストパソコンにストリームされ、プログラム実行しながら解析

最初の2つのレコーディングモードでは、まず、トレース情報の採取、保存が完了した後に、解析が行われます。

何れのレコーディングモードにも、それぞれ強みがあります。FIFOモードは最も一般的な使い方で、エラー発生場所の特定や実行時間解析など、必要に応じて、素早く、何度も使用できます。

STREAMモード：：Corex™-M3/M4プロセッサに装備されているETMv3には、トリガやトレースフィルタが内蔵されておらず、トラブルシューティングに必要な特定のプログラムセグメントだけのトレースを採ることができません。そのため、解析に必要な領域をカバーするよう、比較的長い期間のトレースデータの採取が必要になる可能性があります。このような場合、STREAMモードが最適な選択となります。
しかしながら、STREAMモードでのトレース情報の記録には、次のようなデバッグ環境が必要になります：

- ストリーミングによる大量のデータを取り扱うためTRACE32の64-bit版ソフトウェアの使用が必要.

- μTraceとホストパソコン間の転送レートは、全トレースデータをロスなくストリームするのに十分な速度が必要.　μTraceの256MByteトレースメモリはトレースポート(TPIU)からのロードピークのバッファのために使用.

リアルタイムプロファイリングは、テストによる解析結果がすぐに画面上で確認できるため、オブジェクトレベルのステートメント/ブランチカバレッジの解析に、特に適しています(図3参照)。„ok“とマークされた行は既に実行済み、“not　exec“とマークされた行はさらなるテストを要する可能性があります。

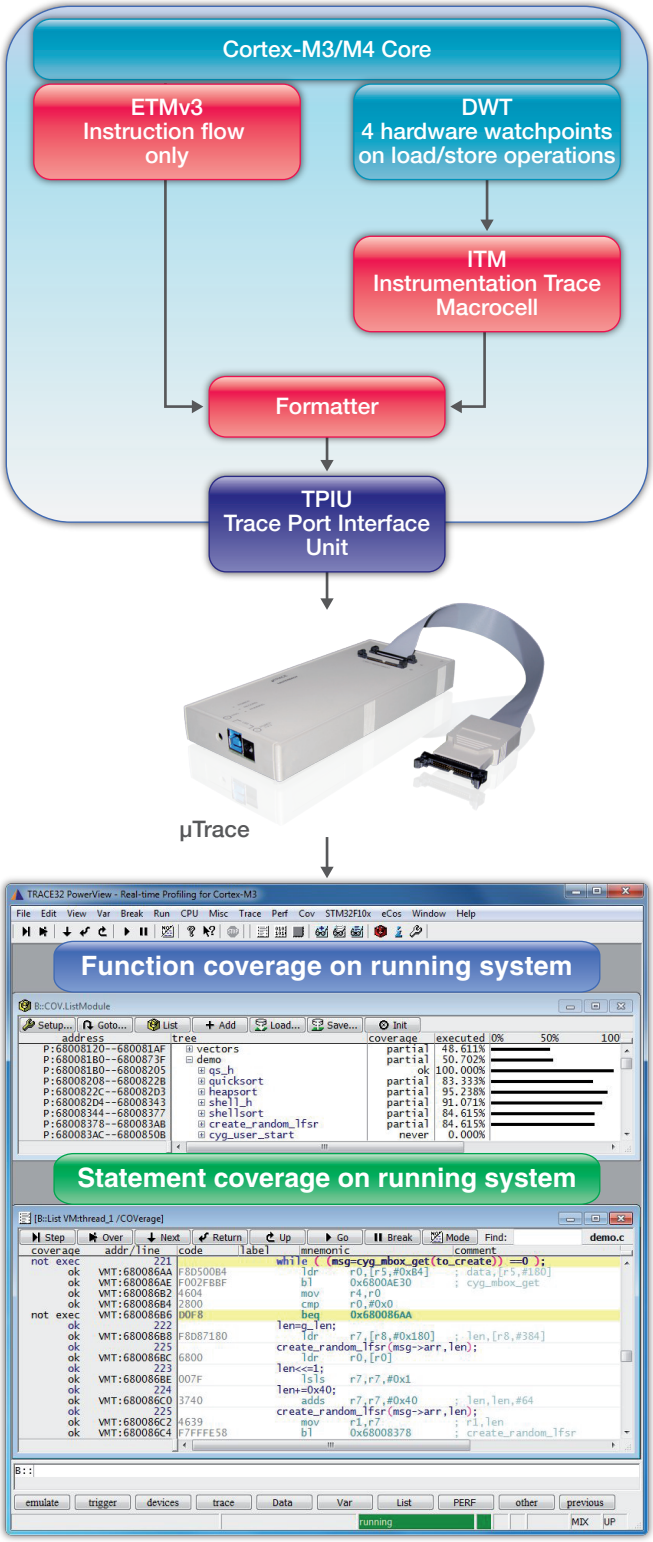

図3: リアルタイムプロファイリングで、実行しながらコードカバレッジ解析を確認