Intelligent Trace Analyses for Cortex-M3/M4

Troubleshooting, performance tuning and code-coverage - all of these can be performed quickly and precisely on an embedded system if the adequate trace analysis is provided.
In 2011, Lauterbach explored new paths to enable optimized trace analyses for the Cortex-M3/M4 processors.

Combining ETM and ITM

For Cortex-M3/M4 processors, trace information can be generated from two different sources (see Figure 3). The **ETMv3** generates information about the executed instructions. The **ITM** generates information about the performed read/write accesses assisted by the Data Watchpoint and Trace Unit (DWT).

The ITM trace packages for read/write accesses contain the following information: data address, data value, program counter.

Through analysis of the program counter, the data accesses which are separately generated can be seamlessly integrated into the program sequence (see Figure 1), which in turn leads to significantly simpler error location.

The cause of an error such as an incorrect data value being written into an address can be easily found if the write accesses are embedded into the overall program trace.

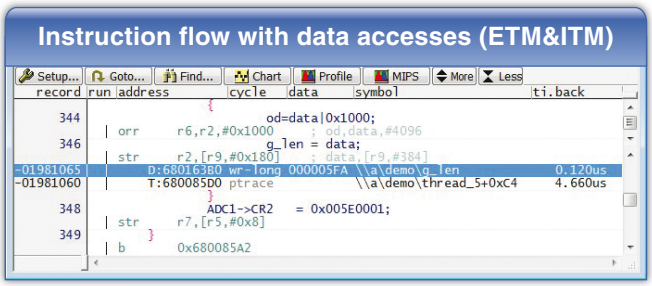

Fig. 1: By combining ETM and ITM trace data, read/write accesses can be integrated seamlessly into the program sequence.

OS-Aware Tracing

If an operating system is running on the Cortex-M3/ M4, task switch information becomes essential for the trace analysis.
In order to receive information about task switches the following method can be used: Trace information on the

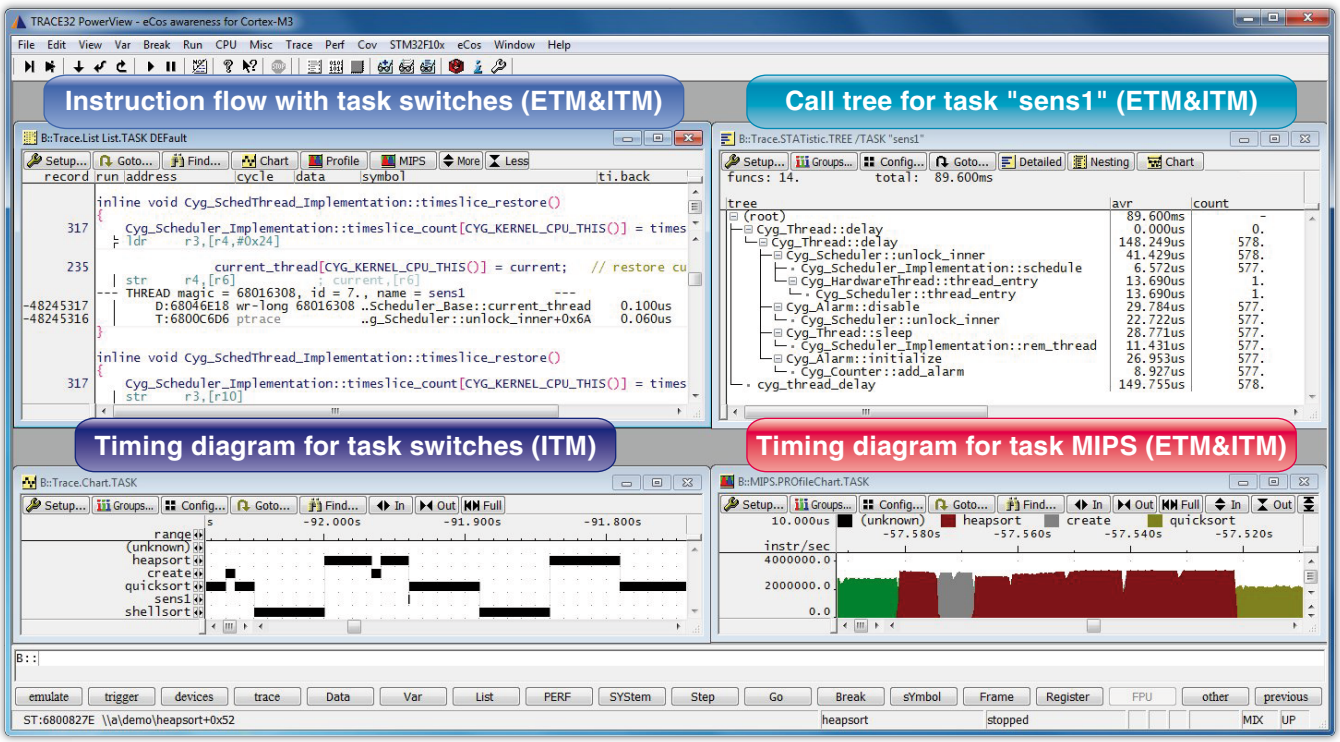

Fig. 2: Through the combination of ETM and ITM trace data, extensive trace analysis can be provided for the eCos operating system.

write cycle in which the kernel writes the identifier for the current task on the corresponding OS variable can be generated using the ITM. As described above the write access information can be integrated seamlessly into the program flow trace. This improves the readability of the trace listing (see Figure 2). The integration of the task switch into the program sequence also forms the basis for the runtime analyses shown in the figure 2.

Three Recording Modes

To record the trace information generated by the Cortex-M3/M4 processors, Lauterbach supports three modes:

- **FIFO mode:** Storing the information in the 128 MByte memory of the TRACE32 CombiProbe.
- **STREAM mode:** Streaming the information to a hard-disk on the host computer.
- **Real-time Profiling:** The trace information is streamed to the host computer and analyzed during runtime.

For the first two recording modes, the trace information is collected and the trace analysis is undertaken after recording is completed.

Each recording mode has its own features. FIFO is the most commonly used mode. It is quick and usually all that is needed for error location and the runtime analyses.

The ETMv3 implemented on Cortex-M3/M4 processors has neither a trigger nor a trace filter. It is not possible to select for recording only those program segments that are needed for troubleshooting. This can mean trace data might have to be collected for a relatively long period in order to cover the area needed for analysis. In this case the STREAM mode can be the best option. The STREAM mode, however, places high demands on the debug environment:

- The large amount of data that results from streaming requires a 64-bit TRACE32 executable. This is needed to allow the address range for the large number of trace entries that will be collected.
- The transfer rate between CombiProbe and host computer must be fast enough to stream all trace data without a data loss. The 128 MByte memory of the CombiProbe is used to cushion load peaks from the trace port (TPIU).

Real-time Profiling is particularly suitable for performing statement and condition coverage. The coverage analysis can be followed live on the screen and the test results are visible immediately (see Figure 3). “ok” marked lines are already covered.

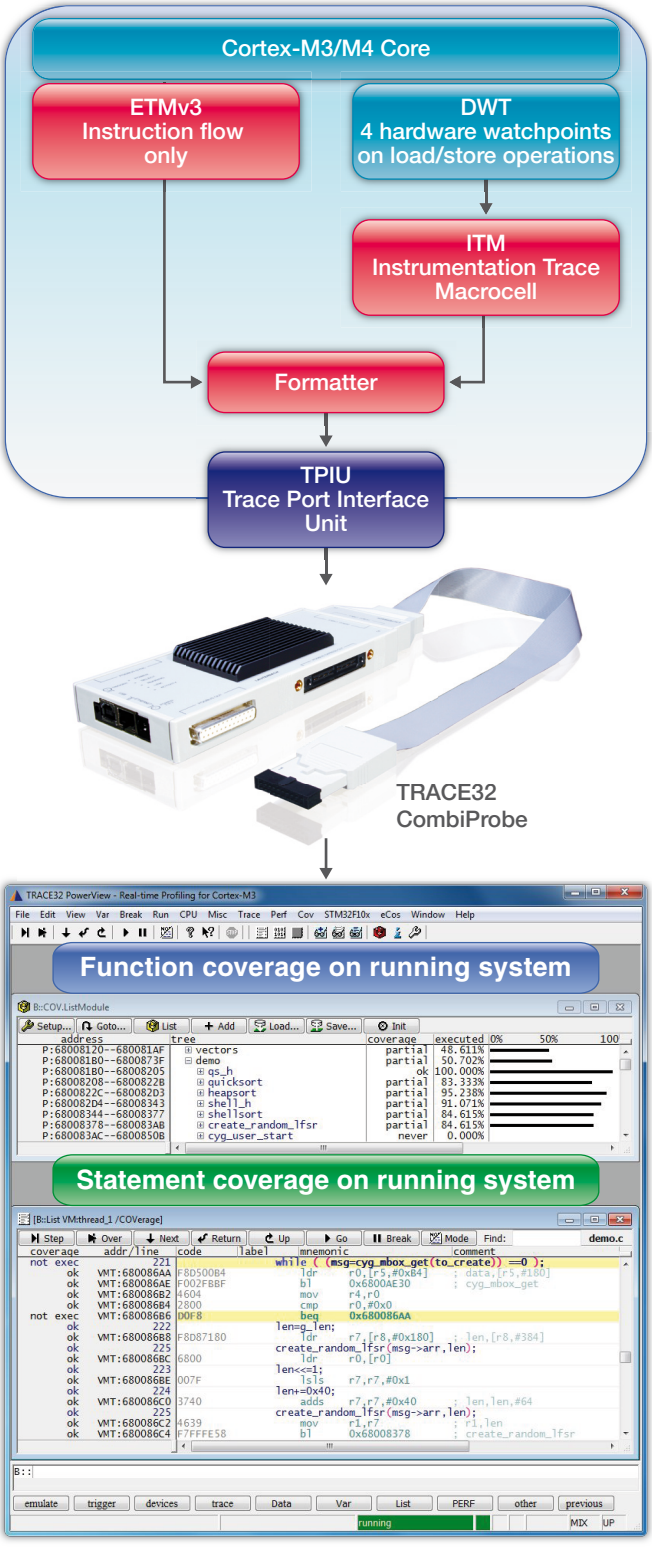

Fig. 3: Real-time profiling enables code-coverage analysis to be followed live on the screen