# Recapturing Vanishing Visibility

**By William E. Novak**

The objective of real-time system developers has always been to debug a system as an application executes in real time. To debug an embedded application, it is important to "see," without intruding on or stopping the application, precisely what is happening as it executes. Developers would like to view data continuously streaming off an application and examine the particular data values pertinent to the application's success. Historically both emulators and logic analyzers have been used, with varying success, to provide this visibility for debugging and tuning real-time systems.

Whatever success debugging embedded applications has had in the past, it is becoming increasingly difficult to achieve today. Real-time visibility is disappearing due to several growing trends:

- Processor speeds continue to rise, with accelerating clock rates requiring ever faster data collection
- Higher system-level integration in "system-on-a-chip" solutions means more functions are moving onto the chip, further decreasing visibility
- Larger on-chip memories and caches exacerbate the problem since their values can only be accessed by the processor bus - i.e., information internal to the device is increasingly inaccessible since I/O to and from the chip runs at a third the processor speed, crippling the ability of traditional methods, such as in-circuit emulators and logic analyzers, to monitor changing values
- The increasing use of heterogeneous multiprocessing systems further compounds visibility because developers must now examine the real-time operation of two or more processors, plus peripherals, together in a system

As a result, developers are losing their ability to debug real-time embedded applications using existing emulation techniques. Real-time visibility, in short, is vanishing just as the need for it in emulation and debugging is stronger than ever.

Fortunately, aid is at hand. Semiconductor manufacturers, recognizing the need of developers to debug and tune their products, are incorporating more emulation visibility capabilities into their chips, such as adding pins that can communicate cache and memory values directly to emulators and logic analyzers. More important, however, are three advanced debug and emulation methods that have emerged to provide developers with better real-time visibility into their embedded system devices:

- With the RTDX™ data link introduced by Texas Instruments in 1998, user-selected data is transferred in real time to an emulator from which it can then be streamed to other applications for analysis and manipulation. RTDX intrudes minimally on CPU cycles and is bi-directional (host to target, or target to host).
- With Advanced Event Triggering (AET) the user can define a complex sequence of events, and then trigger an action such as halting the CPU or driving a pin. AET is essential in finding and analyzing difficult, intermittent real-time software problems for prompt correction.
- With real-time trace, a stream of data or executed instructions is extracted to a trace receiver through a trace mechanism embedded in the processor. This capability also works with AET so that when certain combinations of conditions occur, a trace can be triggered. Although real-time trace capability is completely non-intrusive, it is the most expensive.

Whichever one is used, these methods are helping to meet the needs of developers who demand maximum real-time visibility into their system's operation. Higher levels of visibility not only help to profile a system's execution, they also help tune that system for better performance. Quite simply, if you can't "see" into the system, you can't fix problems, and you certainly can't determine how to improve its performance.

*Bill Novak* is a technology manager at Texas Instruments, Inc.'s Software Development Systems group in Pittsburgh, PA.