New Concepts for Multicore Debugging

For the past three years, the Lauterbach TRACE32 ICD In-Circuit Debugger has supported debugging of multicore SoCs. Lauterbach is now expanding its multicore debugging concept based on experience gained from many customer projects.

To achieve optimum functionality and performance, several cores are increasingly being integrated to create a system-on-chip (SoC). Use of RISC processors merged with one or several DSPs is widespread; however, other combinations are also in use. In multicore designs, generally only one debug interface is provided for all cores in order to save on pins and costs.

Control of several cores over a common debug interface

This new concept means that only one PowerDebug Module and one debug cable are required to debug all cores in a SoC. For this to work, the debug cable must include either individual licenses for all cores used or a multicore license.

For each core that is debugged, a separate TRACE32 application runs on the host. The common TRACE32 system software on the PowerDebug Module ensures that debug commands issued by an application are forwarded to the correct core.

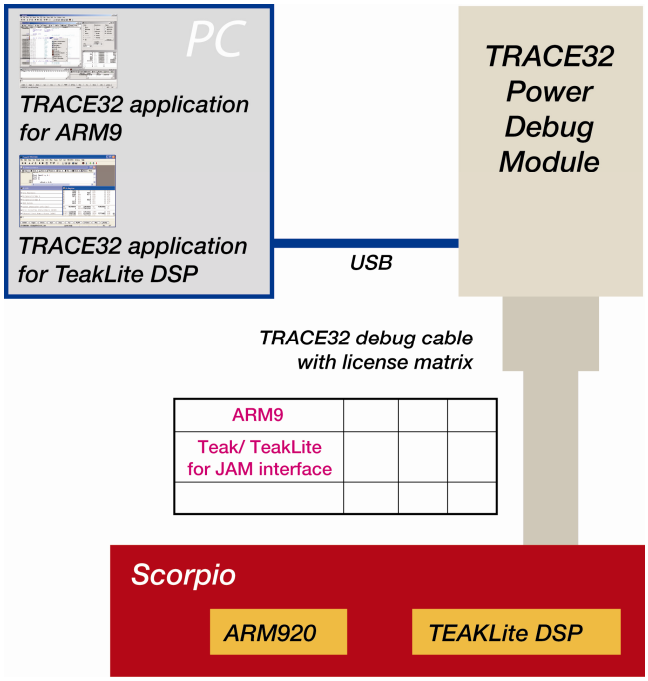

Fig. 1 Debugging ARM920 and TeakLite DSPs in Scorpio over a common TRACE32 ICD

Here are two examples that illustrate this concept: Scorpio: Scorpio from Samsung is a SoC that contains an ARM920 and a TeakLite DSP. To debug both cores at the same time, the debug cable must include a license for ARM9 and a license for TeakLite (see figure 1).

MSC8102: MSC8102 is a SoC from freescale that contains four StarCore DSPs. To debug all four DSPs at the same time, it is sufficient if the debug cable includes a StarCore license and a multicore license.

Start/Stop synchronization

Start/stop synchronization naturally plays an important role in simultaneous debugging several cores. Both processes only work if the SoC supports both synchronized starting and stopping. For example, if the individual cores are connected over a breakswitch, all cores can be made to stop almost synchronously with the clock. Furthermore, a programmable breakswitch allows users to configure which cores in the SoC should stop each other synchronously. If the SoC does not support synchronized stopping, and there is no actual physical way in which the cores can stop each other in the target system, then the debugger can only implement an approximate synchronized stop.

The same applies to start synchronizations: If the individual cores in the SoC can be started at the same time using their debugging logic (for example, using a special JTAG command), then the debugger can implement a synchronized start. Otherwise, the debugger has to start the cores individually in quick succession.

Shared trace port

Many developers who use a multicore SoC design do not want to forgo the advantages of a real-time trace. Naturally, the number of pins for the trace port has also been reduced to save costs. In the meantime, the first SoC designs in which several cores share the same trace port are available. In current designs, developers can specify which core the trace port can use exclusively. There are also plans to develop SoCs in which several cores use one trace port at the same time (for example, in the CoreSight concept from ARM or for NEXUS).

Andrea Martin, February 2005

Altlaufstraße 40 • D-85635 Höhenkirchen-Siegertsbrunn • Tel. +49 8102 9876-0 • Fax +49 8102 9876-999 • sales@lauterbach.com