PRESS RELEASE

Hoehenkirchen-Siegertsbrunn
August 19th, 2019

Lauterbach Announces Full Real-Time Trace Support in TRACE32® For Wind River® Helix™ Virtualization Platform

Lauterbach, the leading manufacturer of micro- processor development tools, announced that their real-time tracing solutions in TRACE32 now support the Wind River® Helix™ Virtualization Platform (Helix Platform). Helix Platform is Wind River‘s new edge compute software platform that runs multiple operating systems and consolidates mixed-criticality applications in a single platform, simplifying, securing, and future-proofing designs for the aerospace, automotive, defense, industrial, and medical markets. TRACE32 supports tracing the full program flow of such heterogeneous systems in real-time, without impact on the run-time behavior. The traced data is then analyzed in terms of timing behavior and code coverage. The support is available for ARMv8 architectures, currently released for Renesas R-Car H3, R-Car M3 and Xilinx Zynq Ultrascale+.

As a Wind River partner, Lauterbach has committed itself to support the company’s platforms, both in debugging and tracing. To this end, working in close cooperation with Wind River, Lauterbach again extended its product TRACE32 to include a comprehensive real-time tracing environment tailored to the Helix Platform.

Helix Platform delivers the VxWorks® RTOS, Wind River Linux, and a type-1 hypervisor together as a foundation to allow applications to independently execute at different safety and security criticalities ensuring real-time, deterministic behaviors are not affected by other functionality on the platform. Helix Platform also provides the capability to run any operating system unmodified as a guest, giving developers the ability to innovate in addition to re-using legacy software and preserving software investment.

The timing of such a virtualized system is crucial for the correct behavior of each guest as well as for the complete system. Therefore, the developer as well as the system designer needs the ability to record and analyze run-time of the real system. Software-based tools need instrumentation, which means you need to know and select exactly what you want to trace. Additionally, the instrumentation itself adds some overhead and changes the timing behavior.

Hardware based tools, which are very common in embedded systems, don't need any instrumentation (or very little) and are able to record anything. However, they need hardware support and some special analyses software that is able to deal with the trace stream of a fully virtualized trace stream.

Some of the Arm CPUs, such as R-Car or Zynq, are equipped with a so called "embedded trace macrocell" (ETM) that is able to collect program flow information during the program run. This information can be collected and stored either internally in memory (which in turn has some resource impacts) or externally by recording special hardware signals. TRACE32 is able to record and analyze this "trace stream". To correctly decode this stream of data, the analyses tool needs to have access to the code that was running, and it needs the information about when a task switch or guest switch happens. As all addresses within the trace are virtual addresses, the tool also needs to be able to resolve the different address translations performed by both the guest OS and by the hypervisor.

There are two big advantages of using an ETM trace instead of software based tracing.

First, (almost) no instrumentation is necessary. The hypervisor included in Helix Platform already provides everything necessary to successfully decode the hypervisor itself and the guest switches. It manages the so called "ContextID" register that exports information about hypervisor task switches and guest switches to the trace stream. There is no special action or modification required for the developer to enable this. For VxWorks guests, a very short task switch hook is required. Actually, all it has to do is write the new task id into the aforementioned ContextID. With Linux, the developer just needs to set a special kernel configuration option to enable this.

Second, the whole system is recorded, including hypervisor, all guests, processes and tasks. If the prerequisites are met, the trace really contains everything; the whole system behavior. That means that it contains the program flow of the hypervisor (e.g. interrupt routines), all guests with all processes and the tasks therein, be it VxWorks or Linux. You can record, trace and analyze all in one shot, at the same time.

Once the trace stream is recorded, there are a lot of possibilities to analyze it. To get a first glance of the run-time behavior, you can view the task timeline, where the tasks of all guests plus the hypervisor are displayed. This gives a quick overview, when which task of which guest was running. You can also generate a statistical report of the task run times, analyzing which task (or guest) took up the most computing time. This analysis can be separated by cores, to get the cpu load of each core. Going deeper into the details, the same can be done on a function level. Minimum and maximum run times, call hierarchies and timing charts show the actual software behavior and help to validate timing requirements, again for the whole system. To be able to see complete program flow, down from the hardware level through hypervisor and guest os up to the application level, is especially useful if you have to find bugs or timing issues related to cross-guest functionality.

For safety- and security-certified critical systems, code coverage is an essential feature. As the trace records the complete program flow, TRACE32 is able to build a code coverage report out of it. "Knowing" all the details about the hypervisor, the root os and the guests, this report will include, again, the complete system, not only structural parts of it. And all this without any code instrumentation, i.e. without affecting the timing at all! Using this, you can easily validate a complete event chain, even if it spans over several guests.

Using Lauterbach‘s TRACE32 together with Wind River‘s Helix Platform allows developers to record, trace and analyze any code in the system at any time giving full access to all parts, including the hypervisor and all guests. This feature is immediately available and is provided with the standard software delivery of TRACE32.

About LAUTERBACH

Lauterbach is the leading manufacturer of complete, modular and upgradeable microprocessor development tools worldwide with experience in the field of embedded designs since 1979. It is an international, well-established company with blue chip customers in every corner of the globe and has a close working relationship with all semiconductor manufacturers. At the headquarters in Höhenkirchen, near Munich, the engineering team develops and produces highly proficient and specialized Development Tools, which are utilized all over the world under the brand TRACE32®. Our branch offices exist in the United Kingdom, Italy, France, Tunisia, on the east and west coasts of the United States, Japan and China. Highly qualified sales and support engineers are also available in many other countries. For more information visit http://www.lauterbach.com