Utilizing TRACE32 Mixed-Signal Probe to Trace Task Switches in Real-Time Operating Systems (RTOS)

2024-12-13 - Comments (0) - Trace

Today, nearly every application runs on an operating system. To effectively analyze the timing behavior of the OS, it is crucial to trace the currently running task. Unfortunately, the tracing infrastructure of the core architecture sometimes lacks both data tracing and context ID tracing. In other cases, the available on-chip trace is so small that a thorough analysis of timing behavior is not possible.

For use cases like these, where long-term task runtime analysis is required, you can leverage GPIO signals available on the target board along with a TRACE32 Mixed-Signal Probe. For example, a solution has been developed for the PolarFire SoC (RISC-V, 64-bit) running FreeRTOS on a single U54 hart. While the PolarFire SoC offers on-chip and off-chip program flow trace, it provides neither data trace nor context ID tracing. FPGA-based SoCs typically have many GPIOs, which makes implementing such a solution easier.

The basic idea is to associate certain GPIO switches with the task switch operations of an RTOS. In this case, the FreeRTOS task switch hook function is utilized to add code that writes the task number to specific GPIOs.

The GPIO signals can be recorded using a Lauterbach logic analyzer (Mixed-Signal Probe). A `DisConfig` command in TRACE32 allows the recorded GPIO signals to be mapped to the task number. To understand the following screenshots, it's important to know that the signals recorded by the Mixed-Signal Probe are displayed in TRACE32 using the TRACE32 `CIProbe` or `IProbe` main command groups. The commands `IProbe.Chart.TASK` and `IProbe.STATistic.TASK` enable now the desired long-term task runtime analysis.

Now, for those interested in a deeper understanding of the solution, here are some additional details.

FreeRTOS provides a "task num" for each created task. It's a progressive number starting from 1.

The "task num" is encoded on several GPIOs during each task switch. In this implementation, three data GPIOs are used (allowing for up to seven available tasks) along with one clock GPIO to sample valid data on the Mixed-Signal Probe. This configuration effectively functions like a bus trace with a single clock line.

Several `CIProbe/IProbe.DisConfig.CYcle` commands are configured to detect write cycles, along with the appropriate 'Strobe', 'Address', and 'Data' fields:

- The Address is the task number
- The values from the GPIOs are used as the Strobe.
- The Data is encoded to retrieve the original FreeRTOS task number based on the "task num" encoded on the GPIOs.

Please note that more than one `DisConfig` command is required—specifically, one command for each task. Each `DisConfig` command uses a distinct strobe to identify a specific task.

The information obtained from `CIProbe/IProbe.List` allows the standard FreeRTOS awareness to generate the `CIProbe/IProbe.Chart.TASK` view. This approach is designed for a single-core RTOS.

Additionally, the PolarFire SoC generates a program flow trace. Since both the PowerTrace (Analyzer), which records the program flow, and the Mixed-Signal Probe (IProbe/CIProbe), which captures the GPIOs, tag their trace records with the same - across all TRACE32 trace tools - timestamp, these recordings can be easily synchronized. The `Analyzer.Chart` Window below shows which functions are executed when the change from task SieveDemo to task StackEater occurred (blue cursor in both windows).