Function run-time analysis using instrumentation-based trace (LOGGER)

2026-04-03 - Comments (0) - Tracing

Instrumentation-based trace can be used as an alternative solution for function run-time analysis when no hardware-based trace is available.

The entry and exist of each function needs to be patched to include a call for the LOGGER functions, which write the trace information to a reserved buffer on the target memory using a trace format provided by LAUTERBACH.

Example

Add a fetch cycle at the function entry with the function name as second parameter:

```
T32_LoggerData (T32_FETCH, MyFunc, 0 /* unused */);
```

Add a fetch cycle at the end of the function to mark the function exit, with NULL as second parameter:

```
T32_LoggerData (T32_FETCH, NULL, 0 /* unused */);
```

Note

The function T32_TimerGet() needs to be populated with an architecture-specific timer to include timing information.

Example for TriCore using STM timer:

```
unsigned long long T32_TimerGet()
{
    return STM0_TIM0.U;
}
```

Refer for more information about LOGGER to

- Beyond Hardware-based Trace: Useful Alternatives for Run-Time Analysis
- TRACE32 Logger Trace
- Application Note for the LOGGER Trace