

Collect Trace Information on MCUs without Trace Port iSYSTEM's Slow Run gathers Debug Trace Information although the MCU has no Trace Port

iSYSTEM supports with its development and debugger software winIDEA (since version 9.12.7) the so called Slow Run Mode. Slow Run allows the capture of the complete program execution and data trace – although the MCU does not provide a debug trace port.

Why is Slow Run Mode an important feature?

When customers select a MCU, the price per unit is an important decision criterion. To save costs many manufacturers do not include an on-chip debug trace port on low-priced MCUs. If during development (and also later) problems occur, without trace port there is no way to get any detailed information on the execution of the application.

In such situations Slow Run mode is the solution.
Slow Run mode provides:

- Analysis of the complete program trace (which instruction was executed)
- Code Coverage (what code path has been executed and what are dead code paths)
- Profiling (when, how long, which function is executing) – although in Slow Run mode time measurement is not in real time

How does Slow Run Mode work?

In Slow Run mode winIDEA executes the target application step by step and gathers CPU state information. Based on that winIDEA constructs a trace file for analysis.

The collection and processing of all of these data needs time – hence the name “Slow Run”. The effective rate is at 30-50 instructions per second, dependent on the architecture used.

Prerequisites for Slow Run Mode

Prerequisites for Slow Run Mode are the usage of an iSYSTEM debugger e.g. iC5000, iC3000 and the corresponding development environment winIDEA.
The MCU only needs to provide basic run control debug - the “single step” mode.

Comparison of trace facilities without and with Slow Run Mode

We use a test application on an iSYSTEM MPC5554 evaluation board. This board offers two interfaces: a JTAG/OnCE port (14 pins) and a NEXUS class 3 port (38 pin Mictor).

As debug tool we use an iSYSTEM iC5000 on-chip debugger.

In the following tests we compare what data is available over the different interfaces.

Trace over Nexus:

Details on the Nexus interface:

The Nexus interface is available in different classes. The MPC5554 evaluation board provides a Nexus class 3 interface (38pins). Nexus class 3 offers a trace port and allows additionally to Run Control Debug also real time data trace and real time read/write of memory areas without stopping the MCU.

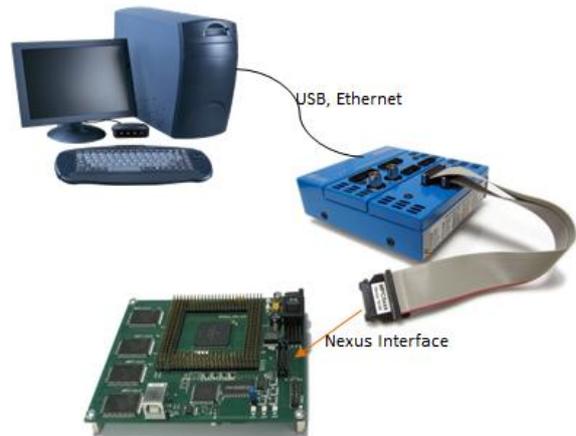


Figure 1: Test Setup over Nexus

Debug test:

Connect the iC5000 over the Nexus interface to the MPC5554 board and start the trace.

Simultaneously to program execution winIDEA shows the complete program and data trace. The data acquisition is handled in real time - without interrupting or slowing down the MCU and without effects on the application's behavior. Additionally the occurrence and reaction on external signals (e.g. AUX) can be tracked.

Based on this trace execution profiling (when, how long, which function is executing) as well as Code Coverage analysis (what source code is executed and what are "dead" code paths) is available.

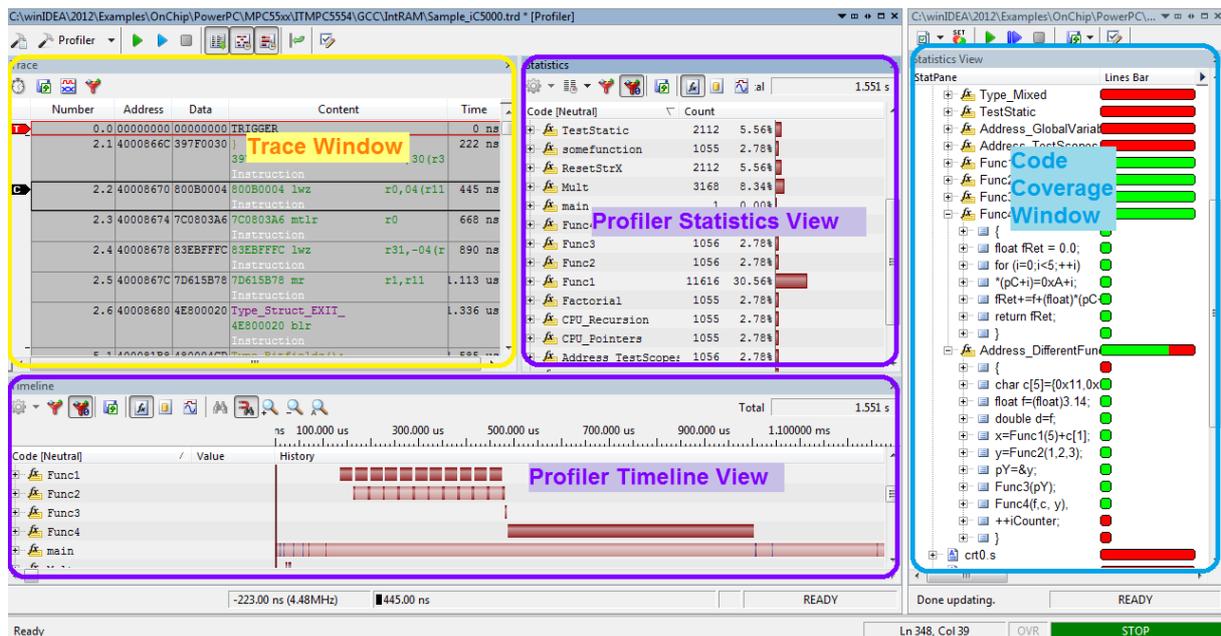


Figure 2: Trace Results over NEXUS

„Trace“ over JTAG/OnCE:

Details on the JTAG/OnCE interface:

JTAG/OnCE interface provides only Run Control debug (e.g. start/stop program execution, read/write registers, set of breakpoints and watches). A complete program execution trace is not available.

Debug test:

When the iC5000 is connected over the JTAG interface to the MPC5554 demonstration board, the trace is not available.

The only way to debug is analysis of the disassembly code and registers content while the program execution is manually stopped e.g. by breakpoints or run until.

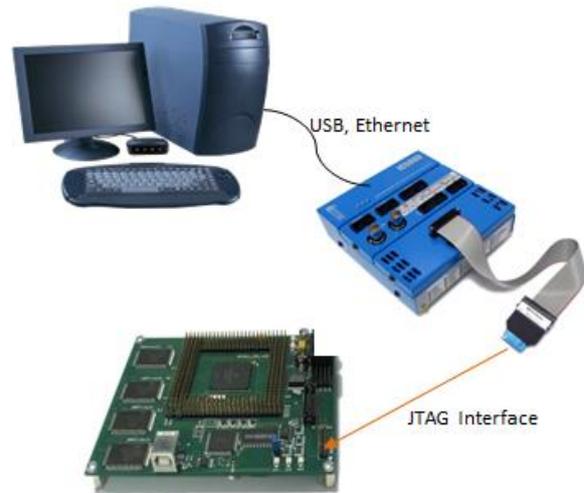


Figure 3: Test Setup over JTAG

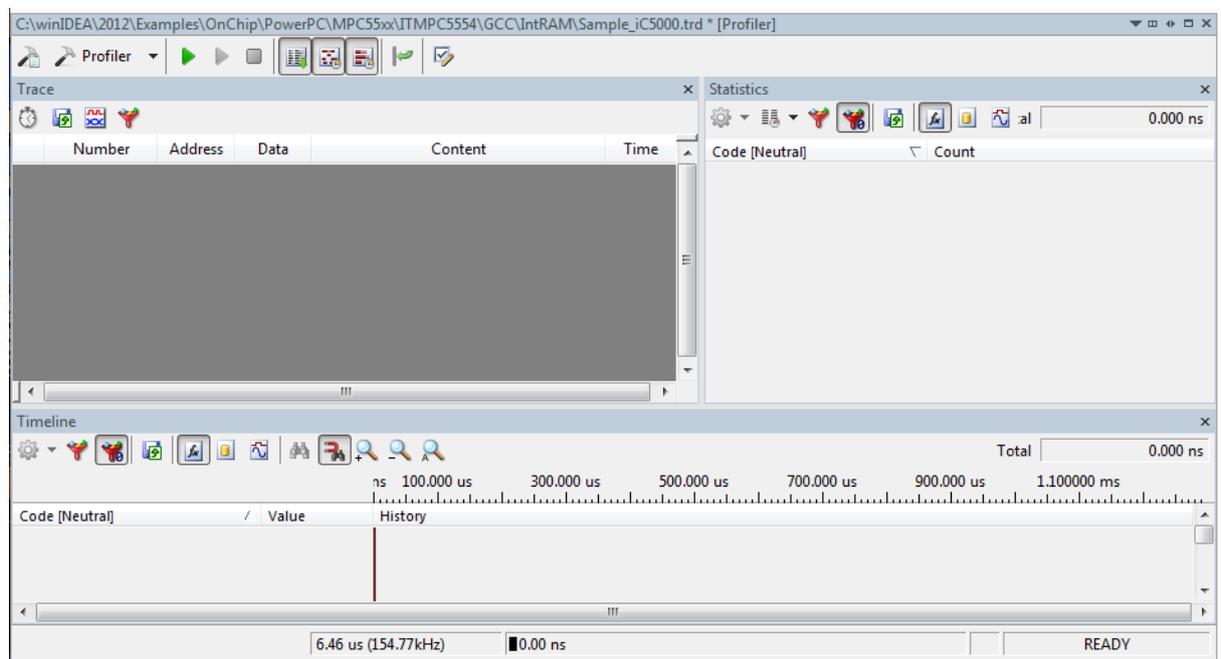


Figure 4: Trace Results over JTAG (without Slow Run)

“Trace” over JTAG/OnCE – using Slow Run Mode:

Details on the interface:

The interface is the same as described in section “„Trace“ over JTAG/OnCE”.

Debug test:

When you connect the iC5000 over the JTAG interface, select “Slow Run” (in winIDEA’s “Debug” menu) and start the trace, you get a complete program execution trace (similar to the trace over the Nexus interface).

The resulting trace file allows analysis of the program execution and data trace plus profiling and code coverage examination. The time base in this case is the number of instructions executed.

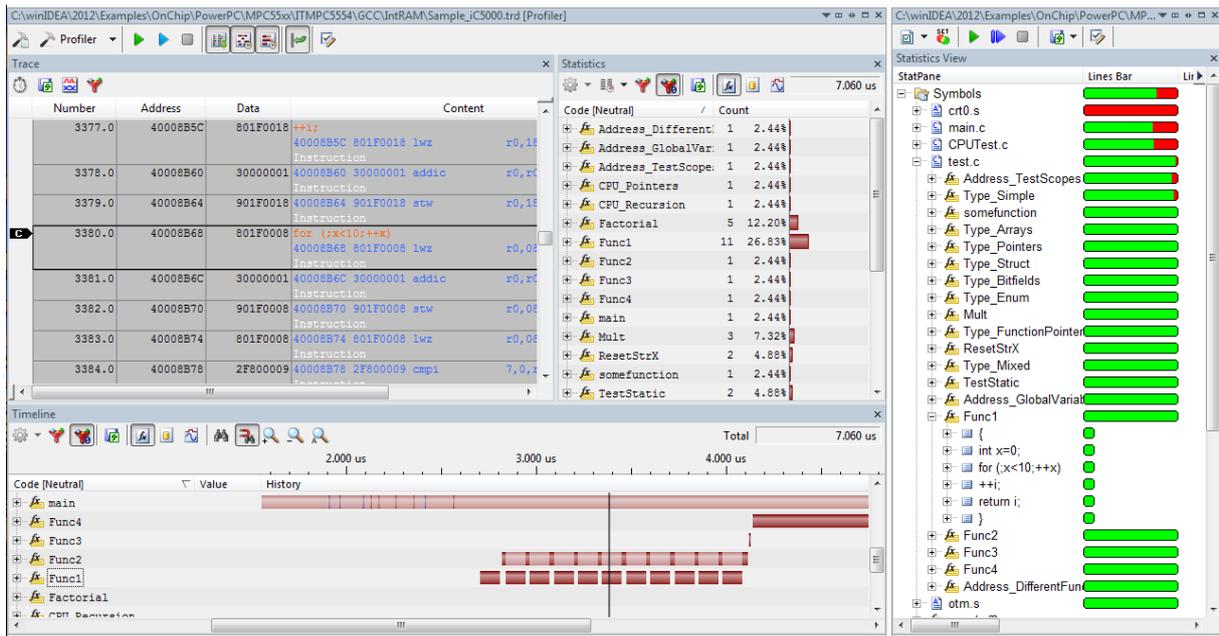


Figure 5: Trace Results over JTAG (with Slow Run)

Comparison of run times with and without Slow Run Mode

As in the previous example we use a test application on the MPC5554 demonstration board. The debug tool is again an iSYSTEM iC5000 on-chip debugger.

We track the execution duration over the Nexus 3 interface with and without Slow Run Mode.

Table 1: Time Duration with and without Slow Run

	46 instructions e.g. library call for cast operation: int i; float fRet; i = fRet;		„Step over“ over a function 7038 instructions	
Interface	without Slow Run	with Slow Run	without Slow Run	with Slow Run
Nexus 3	9,87µs + 171Kb Trace File	2s + 171Kb Trace File	522,7 µs + 190Kb Trace File	85s + 190Kb Trace File

Limitations of Slow Run

Slow Run is well-suited to trace small portions of an application.

If library functions are called (see above table), Slow Run may need a long time to return, because much code is added. Likewise the execution of a function can last very long in Slow Run mode (see above table, example “Step over”).

If a trace of a complete application is needed, it is recommended to run the data acquisition in Slow Run mode overnight.

Another limitation of Slow Run mode is the possibly changed behavior of the application. Due to step by step execution the reaction on external signals e.g. timers, interrupts, can be delayed or not happen at all, e.g. if an interrupt is lost.

Conclusion

Slow Run mode allows to debug and to analyze the complete program trace including profiling and code coverage also on MCUs without trace port.

The most important use case for Slow Run is unit testing with testIDEA. Here the tested functions are relatively short in terms of op-codes executed, but at the same time the execution coverage information is a critical measure of the quality of test vectors. If necessary, a test can be executed twice – under Slow Run to determine vector quality and under regular execution to verify the outcome also in real-time conditions.

Slow Run mode is not a substitute for powerful MCUs with trace ports and according debuggers. It is not suitable for system test or for testing modules under real-time conditions. However, it can provide insights into execution of application parts which would otherwise remain invisible.

Pictures

- Figure 1: TestSetupNexus.jpg
- Figure 2: NexusTraceProfilerCC-2.jpg
- Figure 3: TestSetupJTAG.jpg
- Figure 4: JTAGkeinTRPProf.jpg
- Figure 5: JTAGSlowRunTraceProfilerCC.jpg

About iSYSTEM - Debug & Test Tools - All under One Umbrella

iSYSTEM specializes in embedded development and test tools.

We provide debugger and analyzer solutions for more than 50 CPU architectures and their derivatives (2500+ microcontrollers). The Windows and/or Eclipse based development environment (winIDEA) is easy to learn and use. The flexible integration and application of iSYSTEM solutions within the entire development process is enabled by open and public interfaces (APIs).

iSYSTEM's new iC5000 platform is fully software configurable and adapts to a multi-functional analyzer, development and test tool for many different processors and controllers. An optional I/O module adds the capability to generate, monitor and trace digital as well as analog signals. Additionally, the iC5000 can measure the power and current consumption on target level.

Testing in general and especially according to functional safety standards is gaining more and more importance in the embedded world. iSYSTEM provides a real-time unit test tool (testIDEA) integrated in iSYSTEM's development environment winIDEA that allows execution of test cases without code instrumentation!

To qualify iSYSTEM's tool functionality we make our development and test process transparent to customers. In addition we provide access to our regression test tool suite. This enables our customers to prove that iSYSTEM tools perfectly fit in the context of, e.g., a safety project as required by a standard.

Links to iSYSTEM

www.isystem.com

www.twitter.com/isystemag

Blog: www.embedded.typepad.com

Contact

Erol Simsek

iSYSTEM AG

Sales & Marketing

Phone +49 (8138) 6971-56

Fax +49 (8138) 6971-46

e-Mail: erol.simsek@isystem.com