



[Knowledgebase](#) > [FAQs by core architecture](#) > [TriCore](#) > [\[TriCore\] Why does SYStem.Mode Go cause abnormal behavior when the Boot Code enables HSM and triggers an LBIST reset?](#)

[TriCore] Why does SYStem.Mode Go cause abnormal behavior when the Boot Code enables HSM and triggers an LBIST reset?

2026-09-18 - [Comments \(0\)](#) - [TriCore](#)

This abnormal behavior is caused by a conflict between the hard reset initiated by the boot code (following LBIST execution sequence) and the debugger's rest recovery sequence.

During typical sequence, the debugger resets the target, initializes the debug interface, and then releases the CPU to run.

If the boot code generates another hard reset during initialization, the debugger will follow the behavior configured via `SYStem.Option.RESetBehavior` to restore the debug connection.

Briefly halting the TriCore at the reset vector can lead to issues such as loss of synchronisation with the HSM.

To prevent this, set `SYStem.Option.RESetBehavior` to `RunRestore` to allow the CPU to proceed with its boot sequence while the debugger catches up in the background.

Using this reset behavior:

- Reduces interference between debugger recovery and the Boot Code initialization sequence.
- The CPU continues running after reset: It is not halted after the hard reset to restore debug resources.
- Debug resources are restored in the background. This may cause breakpoints that are intended to hit during the restoration window to be missed.

For more details, refer to the section “**Debugging through Resets and Power Cycles**” in [TriCore Debugger and Trace](#) manual.