[TriCore] Sometimes, after writing to cache memory using the debugger the data reverts back to the original value

2022-01-11 - Comments (0) - TriCore

This FAQ applies to AURIX devices (TC2xx and TC3xx).

When the debugger accesses the memory through the system bus this will update the memory but not the data cached by the CPU and this might cause cache coherency issues.

The debugger can overcome this in different ways. One possibility to access the cache by the debugger is to MAP the cache data and tags via MTU. Check that the option **SYStem.Option MAPCACHE ON** (default) is set.

Due to the internal architecture of the CPU (e.g. usage of the store buffers), stores might not become visible immediately in the data cache and the memory. This introduces another level of incoherency (e.g. between the content of the store buffers and the data caches).

Consider setting **SYStem.Option DSYNC ReadWrite**. This will execute the dsync instruction on the first read or write to ensure that the CPU writes all data back to caches and memory.

Consider also setting SYStem.Option DCREAD ON to display the correct cached data when using the access class "D". In this case, e.g. **Var.Watch** / **Var.View** window will display variables from the CPUs point of view.

Notes:

- It is not possible to access cached memory areas while the CPU is running. In this case, physical memory is accessed.
- Early AURIX devices (TC27x-Astep, TC2Dx) behave like AUDO devices and are not covered by this FAQ.
- For more details about different options and use-cases for the cache handling by the debugger, please refer to the paragraph "Accessing Cached Memory Areas and Cache Inspection" in TriCore Debugger and Trace.