[RISC-V] [HW-Designer] Which core registers (CSRs) are needed by TRACE32?

2024-09-25 - [Comments (0)](#) - [RISC-V](#)

The debugger does not only need access to debug registers of the Debug Module. In order to function properly, several core registers (Control and Status Registers, CSRs) need to be accessible/implemented as well. The following summary lists CSRs that stand in the context of debugging, and mentions whether they are needed or not:

Core Debug Registers

- **Debug Control and Status (dcsr)**

  1. Mandatory register. Without it, debug support is not possible.

- **Debug PC (dpc)**

  1. Mandatory register. Without it, debug support is not possible.

- **Debug Scratch Register 0/1 (dscratch0/1)**

  1. Optional register. The Lauterbach debugger does currently not use these registers.

Debug/Trace Registers

- **Debug/Trace trigger register select (tselect):**

  1. [debug spec v0.13.x]: Mandatory, no matter if triggers are implemented or not

  2. [debug spec v0.14.x and later]: Mandatory if any triggers are implemented, otherwise optional

- **First Debug/Trace trigger data register (tdata1):**

  1. [debug spec v0.13.x]: Mandatory, no matter if triggers are implemented or not

  2. [debug spec v0.14.x and later]: Mandatory if any triggers are implemented, otherwise optional

- **Second Debug/Trace trigger data register (tdata2):**

  1. Mandatory if any triggers are implemented, otherwise optional

- **Third Debug/Trace trigger data register (tdata3):**

1. Optional register (from debugger perspective). Currently, the debugger does only support match control triggers, for which tdata3 is not used. Might however be used for other trigger types in the future.

- **Trigger Extra (textra32/textra64):**

    - Optional register (from debugger perspective). Only needed if the features in this trigger are needed for mcontrol2 or mcontrol6.

- **Trigger Info (tinfo):**

    1. For rules whether this register is optional or mandatory, see RISC-V debug specification.

Machine-Level Control and Status Registers (Machine-Level CSRs)

- **Machine ISA Register (misa):**

    1. This register helps the debugger to determine the capabilities of the core under debug.

        It helps to determine the base ISA of the core. Also, it helps to determine supported ISA extensions such as e.g. floating-point ("F"/"D"), for which the debugger needs to access and display additional floating-point registers.

        If this register is not implemented, the debugger has to fall back to less reliable and intrusive try & error methods to determine the information mentioned above. In some cases, an automatic detection of the respective information may not be possible at all. Therefore, it is **highly recommended implementing** this register, so that the debugger can work in a reliable and minimally-intrusive way.

- Tags
- Hardware Designer
- RISC-V