MCDS

IPextreme

Infineon Multi-Core Debug Solution

PRODUCT BROCHURE

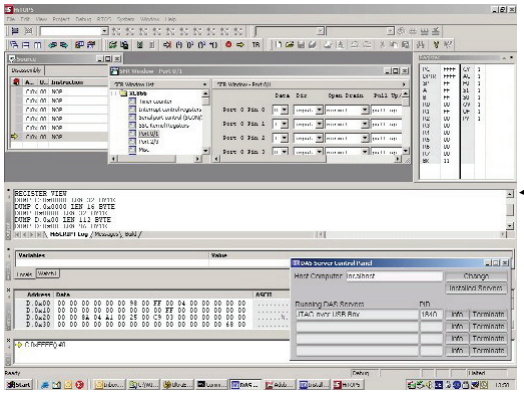
The complexity of today's SoCs has outgrown the capabilities of traditional hardware and software debug methods. A typical SoC now contains multiple processors of different types, several different buses, loads of embedded software, and a multitude of discrete signals. With only a fraction of the internal signals visible at the chip I/O, there is very little visibility into what is going on inside the chip.

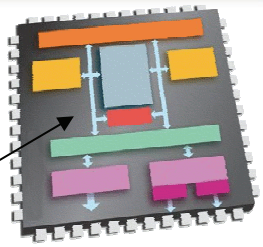
Further complicating the debug problem is the fact that these SoCs are often embedded in machinery where a failure may only occur under real-life conditions—while an engine is running or while a router is pushing packets through the Internet. Debugging these kinds of problems requires capturing the necessary data from deep within a chip at the time that the failure occurs.

Infineon Technologies has developed the Multi-Core Debug Solution (MCDS) to address the problem of debugging deeply embedded, multi-processor systems. By deploying MCDS in either a production SoC or a debug version of the SoC, you can execute real-time, cycle-accurate tracing of selected processors, buses, and signals within the chip non-intrusively, under real-life conditions, and without adding pins to the chip.

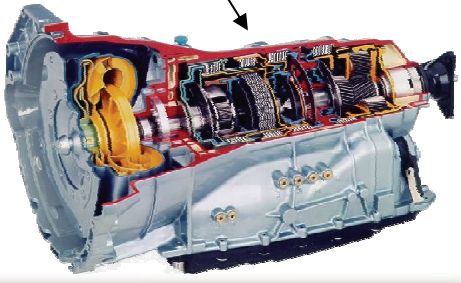
MCDS offers the benefits of:

- Debugging in the real target system: No mechanical or electrical constraints
- Full visibility: Cycle-accurate trace of multi-processor, multi-bus SoCs
- No limitation for low-pin-count, high-frequency devices
- Complex triggering modes—for example, triggering on an event not happening—allowing you to minimize the amount of trace data you collect
- Support for code profiling and performance analysis through programmable event counters
- Portability: MCDS is adaptable to any processor or bus architecture; software developers continue to use tools they are familiar with
- Low cost: No expensive hardware needed to access MCDS
- Proven implementation: Hundreds of software developers are already using MCDS-capable versions of popular Infineon controllers to develop system software





MCDS lets you debug an embedded system **here** while it is operating in real time **here**, using **tools** you already know and without adding pins to your chip.



HOW IT WORKS

MCDS works on the principle of adding logic to your chip to collect and store trace data from selected targets, then transferring that trace data to debug software for analysis. Figure 1 illustrates the concept of how data flows through an MCDS-based debug system. The trace logic shown in Figure 1 could be implemented either in the production chip or on a special debug die that is excluded from the mass production version of the chip.

MCDS supports any number of debug targets. A debug target can be a processor, a bus, or a set of selected signals. In Figure 1, the debug targets are two processors of different types, each with its own debug tool.

The trace message generation logic collects the specified trace information such as instruction pointers, addresses, data, signal values, process IDs, and so on from the debug target(s). Since you do not want to collect all of the available data all of the time, the trace qualification functions as a trace message filter, forwarding trace messages to the trace storage only when specified conditions are met.

The trace storage holds the qualified trace messages in memory where they can be accessed by a Device Access Server (DAS) over a hardware interface such as JTAG. To minimize the size needed for trace storage memory, MCDS compresses the trace messages.

The DAS is the abstraction of the physical connection, which provides a generic interface to the debug tool(s). The tool(s) do the trace reconstruction and interpretation and present the results to the user.

Finally, a small amount of execution control logic is needed in the production die of the chip, coupled to the debug target(s) to handle control such as stop/start/break from the debug tool.

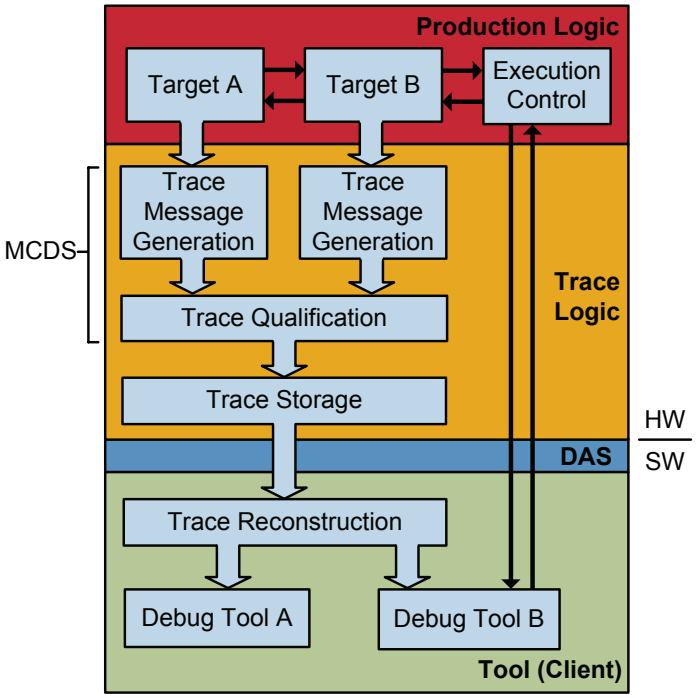


Figure 1: Data Flow through an MCDS-Based Debug System

MCDS FEATURES

- Non-intrusive debugging of embedded multi-processor systems
- Target system runs at full speed in application environment
- Access to internal buses
- Real-time, cycle-accurate tracing
- Trace capabilities for:
  - Processors: Process ID, program, data, status, watchpoint
  - Buses: Data, status, watchpoint
  - Signals: Status
- Complex trigger system including cross-target triggers
- Translates raw data into meaningful messages
- Compresses trace messages to save memory
- Trace memory can be configured as a circular buffer to collect trace messages either continuously or before and/or after a watchpoint occurs
- Implementation partitioned for easy adaptation to new cores

- Can be placed either inside the production die or outside the production die, where it can later be removed for production or kept in place for field diagnostics
- Independent of physical interface between chip and debug host (DAS)
- Security: MCDS is locked by default and can only be unlocked by system hardware
- Implementation proven in Infineon's MCDS-instrumented TriCore devices (TC1766ED and TC1796ED; where ED stands for "emulation device")

CODE PROFILING/PERFORMANCE ANALYSIS

With MCDS, you can connect counters to dedicated signals in the debug target(s)—for example, to count events such as cache hits/misses, bus transfers, instructions executed, stall cycles, interrupts acknowledged, and so on. The counters generate triggers when a programmable count limit is reached; the triggers, in turn, generate messages to the debug tool(s).

IMPLEMENTING MCDS

Implementing an MCDS-based debug system involves a combination of hardware and software. On the hardware side, MCDS is a set of building blocks assembled as needed for the system to be debugged. Figure 2 shows the hardware implementation for a dual-CPU/dual-bus MCDS system.

Each CPU and bus to be traced requires custom adaptation logic to connect the CPU or bus to the corresponding MCDS observation block:

- Processor Adaptation Logic (PAL) connects each CPU to its corresponding Processor Observation Block (POB)
- Bus Adaptation Logic (BAL) connects each bus to its corresponding Bus Observation Block (BOB)

IPextreme works with you to develop the custom adaptation logic.

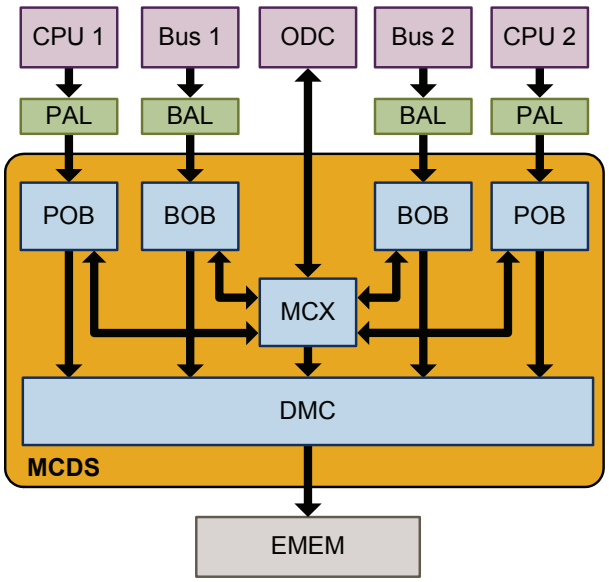


Figure 2: MCDS Hardware Example

The observation blocks (POB and BOB), Multi-Core Crossconnect (MCX), and Debug Memory Controller (DMC) are building blocks sold by IPextreme:

- The observation blocks contain trace units that collect trace information, assemble and buffer trace messages, and extract the triggers used for trace qualification. Additionally, each observation block implements the trace qualification for its internal trace units and sorts the resulting trace messages to send to the DMC.
- The MCX enables simultaneous, real-time debugging of multiple targets, rather than simply time-sharing the debug connection. The MCX provides enhanced trace qualification using cross-connected triggers from the different observation blocks and the host system's On-chip Debug Control (ODC) logic, enabling complex trigger generation based on user-specified interdependencies. Furthermore, the trace qualification can be based on programmable state machines contained in the MCX. The MCX can also forward the associated break signals to the ODC. The performance counters are also implemented in the MCX.
- The DMC sorts messages from the observation blocks and forwards them to user-implemented emulation memory (EMEM).

MCDS connects to and is configured through a peripheral bus of the host system.

Finally, your debugging software and DAS will have to be adapted for the debug target(s). The partitioning of the DAS architecture supports straightforward adaptation to new targets.

DEBUG TOOL SUPPORT

Several toolchain providers offer MCDS-compatible products; for example:

- PLS Development Tools (www.pls-mc.com):
  - MCDS support available as add-on for UDE (Universal Debug Engine)
  - JTAG debug interface
- Hitex Development Tools (www.hitex.com): JTAG debug interface
- Lauterbach (www.lauterbach.com):
  - TRACE32-ICD Debugger with MCDS support over JTAG
  - JTAG debug interface
- Tasking (www.tasking.com)

Contact IPextreme for more information regarding MCDS tool support.

MCDS IP DELIVERABLES

- Synthesizable VHDL source code for MCDS building blocks
- Example adaptation logic for processors and buses
- Example code for dual-CPU/dual-bus MCDS implementation
- Documentation
- Design service to implement custom logic

IPextreme, Inc.
307 Orchard City Drive
Suite 202
Campbell, CA 95008
800-289-6412 (toll-free)
408-608-0421 (fax)

www.ip-extreme.com