

Article Overview

Effective debugging of large core switches involves multicore trace tools, on-chip debug systems, conditional breakpoints, and parallel debugging frameworks to manage thousands of cores efficiently.

Multicore Debugging Tools

For complex multicore systems, tools like TRACE32(R) provide deep insights into both hardware and software layers. TRACE32 supports debugging across SMP (Symmetric Multiprocessing), AMP (Asymmetric Multiprocessing), iAMP (Integrated Asymmetric MultiProcessing), and manycore architectures with synchronous run-control of all cores. It allows tracing multiple cores simultaneously, including heterogeneous cores, and supports debugging of applications, operating systems, and hypervisors without disrupting execution .

On-Chip Debug Systems

Many modern switches and SoCs include On-Chip Debug Systems (OCDS) and Multi-Core Debug Solutions (MCDS). These provide hardware breakpoints, dedicated interrupt resources, and trace logging with timestamps. This approach enables developers to monitor program execution and performance without affecting timing behavior, which is critical for high-speed networking environments .

Parallel Debugging Techniques

For very large systems, such as HPC clusters or switches with hundreds to thousands of cores, GDB's non-stop mode is effective. It allows individual processes to continue running while others are examined, preventing the entire system from halting. Techniques include attaching GDB to specific ranks, setting conditional breakpoints for selected cores, creating process groups, and using Python scripts to analyze communication patterns and memory access across cores .

Graphical and Event-Based Debugging

GUI-based frameworks like ManyGUI help visualize many-core behavior by collecting high-level events related to computation, memory, and communication. This approach provides a comprehensive view of packet paths, memory access statistics, and energy usage, which is particularly useful for debugging network-on-chip (NoC) architectures in large core switches .

Practical Recommendations

Debugging Methods for Large Core Switches

- o Use hardware-assisted tracing to minimize performance impact.
- o Set conditional breakpoints to isolate issues in specific cores or processes.
- o Leverage GUI tools for visualizing communication and memory patterns.
- o Combine software and hardware debugging for full-system insight.
- o Automate analysis with scripts to handle large-scale data from thousands of cores. By combining these methods, engineers can efficiently debug large core switches, identify bottlenecks, and optimize both hardware and software performance.

Master embedded systems debugging with proven techniques, tools, and best practices to find bugs faster and build

FreeSWITCH is a Software Defined Telecom Stack enabling the digital transformation from proprietary telecom switches to a

With high performance, large capacity, and high reliability, Core Switches offer a wide range of features and play a

In large and unfamiliar codebases, developers often rely on error-message debugging, hypothesis-testing, and

Debugging embedded systems requires a blend of hardware tools (JTAG/SWD probes, oscilloscopes, logic analyzers) and software

Debugging--the process of identifying, localizing and fixing bugs--is a key activity in software development . Due to

Our goal is to provide a system that provides a network environment (especially one that uses reconfigurable hardware) with the

With multiple cores and support structures often embedded within a shared SoC architecture, conventional piecemeal

Easy-to-use User interface guides to setup process Predefined configurations for Architectures, Devices, Boards Select cores for

The debugger as a user interface for the complex debug hardware must support the user in its work the find mistakes

This article gives a practical, high-signal overview of JTAG and SWD debugging techniques--from wiring and

level

To debug a digital chip with a scan-based debug methodology, the chip is stopped at a certain point in time in the

One of the most accessible debugging methods is strategic use of serial output. Research on embedded systems security indicates

New Concepts for Multicore Debugging For the past three years, the Lauterbach TRACE32 ICD In-Circuit Debugger has supported

Introduction This application note describes Core8051s debug operation in Axcelerator®; and RTAX-S devices. Core8051s is a high

To help you with the process of design debug and verification, new tools are required to help debug your design while it is running at

Web: <https://www.kremgroup.co.za>

