2025 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION INDIA

Remote and Probeless Debug Methodology for Data Center Silicon Debugs

Mamatha Kallimatada, Google IT Services India Private Limited
Sasidhar Sunkari, Google IT Services India Private Limited
David Kapchits, Google Israel Limited

***Abstract*-Given the scale of modern data centers, maintaining hardware reliability and performance presents significant challenges. Data center operations are characterized by high complexity, scale, and the constant pressure of maintaining uptime. Return Material Authorization (RMA) processes, involves the diagnosis and replacement of faulty hardware components. We process a significant volume of RMAs per month with suppliers. A small number of these units are screened with CPU failure signatures which triggers the Failure Analysis (FA) flow. A significant challenge in maintaining these systems is the debugging of hardware failures. Traditional debugging methods, reliant on physical access and intrusive probes, exacerbate these challenges. This methodology introduces a novel, remote debugging method that eliminates the need for physical hardware access. Within the context of this methodology, the successful implementation hinges on the prerequisite that all Baseboard Management Controllers (BMCs) involved are equipped with a functional JTAG controller. This methodology leverages the BMC's existing JTAG controller, which provides the industry-standard access required for deep silicon failure analysis. While high-level BMC channels are effective for initial triage, JTAG remains the indispensable tool for deep silicon failure analysis, especially in critical hangs where other interfaces are unresponsive. Utilizing the BMC's JTAG controller, the proposed debugger enables efficient troubleshooting of SoCs in data center RMA processes and offers several key benefits.**

Keywords—Remote Debugging, Probeless Debugging, Data Center, Failure Analysis, Silicon Debug, Baseboard Management Controller (BMC), JTAG (Joint Test Actions Group), RDDI (Remote Debug Device Interface), Network Latency, Hardware Diagnostics.

I. Introduction

The operational complexity and sheer scale of modern data centers demand high levels of hardware reliability and minimal downtime. The Return Material Authorization (RMA) process, essential for managing hardware failures, often involves intricate failure analysis (FA) to diagnose root causes, particularly for components exhibiting CPU failure signatures. Traditional hardware debugging methodologies, heavily reliant on physical system access and the use of intrusive JTAG probes, present substantial logistical and economic challenges in the data center context [1][3]. These methods often require significant technician time, specialized equipment, and potentially extended system downtime, impacting service availability and operational expenditure.

The need for efficient, scalable, and non-intrusive diagnostic techniques is paramount. This paper proposes a novel remote, "probeless" debugging methodology designed specifically for data center environments. The core principle involves utilizing the JTAG [2] (Joint Test Action Group, IEEE Std 1149.1) controller commonly integrated within the Baseboard Management Controller (BMC) found on server motherboards. By leveraging this existing hardware capability, remote debugging sessions can be established without requiring physical probes or direct access to the system under test (SUT). A critical prerequisite for this methodology is the availability of a functional JTAG controller accessible via the BMC firmware/software stack.

The proposed methodology offers several key advantages:

- Reduced Downtime: Faster diagnosis and resolution of hardware issues minimize service interruptions.
- Improved Efficiency: Eliminates the need for physical access and intrusive probes, streamlining the debugging process.
- Cost Savings: Reduces the expenses associated with physical debugging equipment and on-site technician deployments.
- Enhanced Scalability: Supports debugging operations across multiple systems and geographical locations, making it suitable for large-scale data centers.

This paper details the evolution of this methodology, starting from an initial architecture and progressing to an optimized approach that overcomes critical performance limitations. Section II describes the initial methodology and its performance bottleneck. Section III presents the optimized architecture. Section IV discusses the performance improvements. Section V explores the broader applicability and benefits, and Section VI concludes the paper.

II. INITIAL REMOTE DEBUG ARCHITECTURE AND LIMITATIONS

The initial implementation followed a conventional remote debugging paradigm. In this model, a user on a host computer initiates debugging commands using ARM-DS or debugging utilities (scandump, memdump or a jtag application). These tools communicate with the RDDI (Remote Debug Device Interface) server which is a protocol for communicating between host computer and target device for debugging purposes. This server communicates with the Debug ProbeAPI which translates RDDI commands to RPC(Remote Procedure Calls). On the BMC side, there is a RPC server listening to these messages, communicates with the target hardware via BMC's JTAG interface, and establishes a debugging session. The resulting data is returned to the RDDI server, processed, and forwarded back to the debug tool for user visualization.

This is depicted conceptually below:

Debug Tool (Host) -> RDDI Server (Host) -> Probe API (Host) -> Network (RPC) -> BMC (RPC Server -> JTAG Driver) -> Target SoC

While functionally capable, performance evaluations revealed this architecture to be approximately 30 times slower than traditional JTAG debugging using a physical probe directly connected to the host. The root cause was identified as the excessive network traffic generated between the host machine and the BMC. High-level debug commands (e.g., read a block of memory) are decomposed by the RDDI server into a large volume of individual JTAG Shift-IR and Shift-DR operations. Each of these low-level operations, when executed via the described architecture, incurred the latency of a network round-trip (Host -> BMC -> Host). The cumulative effect of this latency for complex debug sequences rendered the approach impractical for typical FA workflows.

III. OPTIMIZED BMC-RESIDENT DEBUG ARCHITECTURE

To mitigate the severe performance bottleneck caused by network latency, a significant architectural modification was implemented. The core change involved relocating the components responsible for translating high-level debug requests into low-level JTAG transactions from the host machine onto the BMC itself.

The revised architecture is depicted conceptually below:

Debug Tool (Host) -> Network -> BMC (RDDI Server -> Probe API -> JTAG Driver) -> Target SoC

In this optimized model, the user on the host computer issues the same high-level debug commands using their preferred tool. These high-level commands are transmitted over the network directly to the BMC. The RDDI server and the Probe API logic now reside and execute on the BMC. The BMC-resident RDDI server receives the high-level command (e.g., "read memory address 0x1000"). The RDDI server, running locally on the BMC, translates this into the necessary sequence of low-level JTAG Shift-IR and ShiftDR operations. These low-level JTAG operations are passed directly (via local inter-process communication or function calls within the BMC's software environment) to the BMC's JTAG driver. The JTAG driver interacts with the Target SoC. Only the final result of the high-level command (e.g., the memory content read) is packaged and sent back over the network to the host debug tool.

IV. PERFORMANCE EVALUATION OF OPTIMIZED ARCHITECTURE

The architectural shift described in Section III yielded substantial performance improvements. By executing the RDDI server and Probe API on the BMC, the translation from high-level debug commands to low-level JTAG scans occurs locally. Consequently, the high-frequency Shift-IR/Shift-DR operations no longer traverse the network. Network communication is limited primarily to the transmission of the less frequent, high-level debug commands

and their corresponding results. Performance measurements of this optimized architecture demonstrated command execution speeds nearly equivalent to those achieved with traditional, probe-based debugging setups. The elimination of network round-trip latency for the vast majority of JTAG transactions was the key factor in achieving this performance parity. This restored the practical viability of the remote, probeless debug methodology for data center FA tasks.

V. DISCUSSION

The success of the optimized architecture highlights the critical impact of network latency on remote JTAG operations and demonstrates the effectiveness of localizing protocol intelligence. Moving the JTAG transaction generation logic (in this case, the RDDI server) to the edge device (the BMC) closest to the target hardware is crucial for achieving acceptable performance. While this implementation utilized the RDDI protocol, the underlying principle is broadly applicable. The core requirement is a mechanism to: 1. Transmit high-level debug requests from a remote host to the BMC. 2. Execute logic on the BMC that translates these requests into the appropriate low-level hardware (JTAG) interactions. 3. Utilize the BMC's JTAG controller to interact with the target SoC. 4. Return results to the remote host. Alternative implementations could potentially employ different protocols or middleware. For instance, a custom server application on the BMC could be developed to interpret commands from various debuggers. Integration with other industry-standard tools, such as Lauterbach TRACE32, might be possible if appropriate server components and BMC JTAG driver interfaces were developed. The essential element remains the BMC's role as an intelligent debug agent capable of local JTAG control, rather than merely a passive physical interface proxy. The primary prerequisite remains the availability of a functional and accessible JTAG controller on the BMC platform. Ensuring robust BMC firmware and software support for JTAG operations is essential for the deployment of this methodology. The benefits realized – reduced downtime, improved efficiency, cost savings, and scalability – make this remote, probeless approach a valuable tool for managing hardware diagnostics in complex data center environments. It significantly lowers the barrier to performing detailed silicon failure analysis on RMA units, potentially leading to faster identification of systemic issues and improved hardware revisions.

VI. CONCLUSION

This paper presented a remote, probeless debug methodology tailored for data center silicon failure analysis, leveraging the JTAG controller integrated within server BMCs. An initial architecture suffered from prohibitive performance limitations due to network latency associated with transmitting low-level JTAG commands. The key innovation presented is the architectural shift relocating the debug protocol intelligence (RDDI server and Probe API) onto the BMC itself. This optimization confines high frequency JTAG transactions locally within the BMC, drastically reducing network traffic and eliminating the associated latency bottleneck. The resulting methodology achieves performance comparable to traditional probe-based debugging while retaining the significant operational advantages of remote, non-intrusive access. This approach offers substantial benefits in terms of reduced cost, improved diagnostic efficiency, and enhanced scalability, representing a significant advancement for hardware debugging practices in large-scale data center operations.

REFERENCES

[1] M. Marra, G. Polito, and E. Gonzalez Boix, "Out-Of-Place debugging: a debugging architecture to reduce debugging interference," arXiv preprint arXiv:1811.02034, 2018.

[2] IEEE Standard for Test Access Port and Boundary-Scan Architecture, IEEE Std 1149.1-2013 (Revision of IEEE Std 1149.1-2001), Dec. 2013.

[3] ARM Limited, ARM® Debug Interface Architecture Specification ADIv5.0 to ADIv5.2, ARM IHI 0031C, 2010.

[4] J. F. Guan, Y. L. Wang, and J. W. Li, "Challenges and research directions in reliable data center networking," Frontiers of Computer Science, vol. 9, no. 1, pp. 13-23, Feb. 2015.