Leo C. Wood

Senior Computer Engineer/Software Developer

Visit *LeoCWood.com* for more details

Current Address:
100 Torreyglen Drive
Lexington, South Carolina 29072

Contact Information:
Email: lcwood@outlook.com
Mobile: (817) 443-8460

EDUCATION: **B.S., Computer Engineering.** May 2013
Oral Roberts University, Tulsa Oklahoma

LANGUAGES: C, C++, C#, Java, Python, Bash, Perl, MySQL, PHP, HTML, Verilog, VHDL, Assembly, Matlab, …

WORK EXPERIENCE:

Intel Corp, Columbia, SC Xeon Server System Reliability Validation Engineer February 2015– Present

- I am currently a system validation engineer and technical lead with primary responsibility for the features directly related to reliability, availability, serviceability, and maintainability (RASM) on Intel Xeon Server platforms. These features include corrupted data containment, out-of-band access to error logs, hardware error injection, software error handling, server cluster maintenance and debug, I/O error escalation, memory error escalation, and error recovery.
- Along with a small team of engineers, I own writing, code reviewing, executing, and debugging the software libraries responsible for these RAS features. I have developed two different validation frameworks to accomplish my goals in both emulation and OS environments.
    - For emulation:
        - My software runs on Synopsis Zebu 3 emulation servers, so the available CPU cycle count is very limited. My test suite is written in C++ and includes libraries to install interrupt handlers, detect hardware capability, initialize RAS features, and exercise all the different register access methods that Intel uses, including MMIO, MSR, CR, PCIe CFG access, funny IO, and more.
    - For post-silicon (executing on a live system booted to an OS):
        - Using Python and C++, I wrote a test suite that interfaces with the ACPI tables and Linux Kernel to verify that the UEFI configured hardware correctly, responded to error events, and that the OS logged errors correctly. The tests I wrote under this framework are regressed on server clusters all over the world.
- I spend roughly 80% of my time writing new software using C++ and python, as well as maintaining code, debugging, and engaging in collaborative code reviews. I engage in the Agile software development cycle with my team as an individual contributor. I use GIT and SVN every day to manage my software commits.
- When problems with the silicon design occur, I routinely dive into debug by reviewing source Verilog/VHDL. I've also become versed in Linux Kernel and UEFI/ACPI source debug.

Intel Corp, Chandler, AZ Microprocessor Component Design and Validation Engineer June 2013 – February 2015

- My role was the power management technical lead for cutting-edge Atom based SoCs, with responsibilities of creating power management test plans, writing test and debug software, automating tests for regression, debugging failures, and responding to customer issues with a sense of urgency. Using both pre-silicon and post-silicon tools, I validated both the architecture and firmware of embedded microcontrollers that are responsible for executing advanced power management flows. These include connected-standby, Intel Speedstep, Intel Turboboost, resets/shutdown, DDR Voltage and Frequency scaling, and suspend-to-RAM/Disk.
- Using Python I wrote the test and debug software content that proved essential for enabling validation teams during the life-cycle of these products. In order to be reusable from project to project, object oriented design was heavily laced into my code. The debug tools are still in use by hundreds of engineers and help ensure time-to-market by providing quick triage of power management related failures.
- Using C I routinely created debug patches for our embedded microcontrollers in order to experiment during critical debug efforts. I engaged in code reviews with fellow design partners and deployed patch updates to regression test suites before they went into production.

Intel Corp, Chandler, AZ Microprocessor Validation Engineering Intern January 2012 – July 2012

- I executed production ready validation stress-test suites comprised of synthetic benchmarks that were used to increase our confidence in product health and robustness before it shipped to market. Later I automated the test suite to run on platforms while performing voltage-temperature shmoo.
- I automated the process of finding manufacturing-induced silicon bugs using small-scale robotic tester tools. The number of defect products in a given sample lot was used to determine the level of customer exposure.

Oral Roberts University, Tulsa, OK Spectrum Analyst, Student Research August 2012 – May 2013

- I performed a 2.4 ISM band spectrum analysis in a local hospital using National Instruments PXIe-1082 8-Slot 3U PXI Express Chassis populated with a PXIe-5663E 6.6 GHz Vector Signal Analyzer. The aim of the project was to analyze, using Matlab and numPy, the probability of device coexistence during peak network traffic in the ICU. The findings were shared with Oklahoma University in a joint effort for IEEE publishing.

REFERENCES: References Available Upon Request