Justin David Smith

Verification Manager, Advanced Micro Devices
http://www.linkedin.com/in/justindavidsmithca

resume (at) chaos2.org
Phone number available on request

Objective: To solve large-scale verification problems through the development of novel formal specification, modeling and analysis techniques with a team of creative, motivated and accomplished engineers. Interests include **verification methodology**, **graphics**, **infrastructure** and **language design**.

Industry Experience

Manager, ASIC Layout/Design Sept. 2009 — present
Technology Engineering, Advanced Micro Devices, Sunnyvale

Staff Management:

- Managing a team of 9 engineers in North America and Shanghai that verifies the Radeon graphics shader core in blocklevel, IP level and SoC environments.
- Mentoring junior engineers and interns on team, helping them develop basic technical skills and higher-level skills including test planning, engagement with architects and design engineers, and advanced analysis techniques for regression failures.
- Hands-on management style — continuing to own modest technical tasks to remain connected with day-to-day engineering problems, while allowing engineering staff a wide degree of latitude in solving problems.

Project Management:

- Verification lead for Shader Core Next, a SIMD architecture supporting modern compute and graphics shader operations; this core appears in current Radeon products and in two current-generation gaming consoles. Led verification for three generations of this design, and implemented architectural specification language and assembler/disassembler tool for shader microcode. Each generation has enjoyed low ECO rates and no issues were found that triggered a respin.
- Lead for IP level infrastructure improvements, focusing on compute efficiency and ease of debug, and overseeing infrastructure team in Shanghai responsible for implementing improvements.
- Program lead for the graphics IP component of two APU chips; gave guidance on program schedule, tracked IP-level bugs, managed integrations to SoC level and coordinated PD feedback to graphics design team.
- Engaged with other leads to develop next generation test description language to replace aging internal tools. Represented the needs of our IP team to ensure the infrastructure deployed is intuitive and straightforward to use, and capable of expressing a variety of test characteristics.

Technical:

- (**perl**, **coverage**) Designed tools to analyze regression and coverage data to reduce the overall regression footprint, covering the full design with the fewest number of tests, lowest CPU time and lowest memory usage. Tools assess individual test quality and have reduced time spent in coverage reviews and debug of superfluous tests.
- (**PHP**, **SQL**, **coverage**) Developed new tracking tools for functional and code coverage with task assignment and automatic updates from regression runs.
- (**perl**, **regression**) Designed a regression environment to automatically track inflection points (where the health of a regression changes substantially) and search for changes contributing to new failures.
- (**C++**, **SystemC**, **testbench**) Made significant technical contributions to random stimulus generator and tests to support the team and ensure success. Developed significant “anti-optimizations” (instruction sequences seeking worst-case arbitration scenarios) and dataflow for random instruction generation.
- (**perl**, **p4**) Developed tools that analyze perforce submissions to monitor integration activity across multiple codelines, checking that design fixes are propagated to all relevant codelines.

Member of Technical Staff, ASIC Layout/Design Oct. 2006 — Sept. 2009
Graphics Products Group, Advanced Micro Devices, Sunnyvale

- (**C++**, **SystemC**, **testbench**) Designed from scratch block-level test infrastructure and test plan for new color blender design, and implemented a majority of the random test cases. The new code shipped with no ECOs.
- (**verification lead**) Led shader complex verification effort in one generation; the design was used in successful GPU and APU chips.
- (**C++**) Developed random tests for shader complex for three generations. The tests scaled well despite numerous large changes in shader micro-architecture in each generation.
- (**C++**, **infrastructure**) Deployed languages for constrained random stimulus generation for multiple blocks, including cache controllers and instruction pipelines.
- (**C++**, **modeling**) Developed infrastructure for performance-accurate software model of the GPU core for use in full-system simulators.

Senior Verification Engineer June 2003 — Aug. 2005
Graphics Verification Group, ATI Research, Santa Clara June — Oct. 2006 (acquired by AMD)

- (**microcode**) Designed microcode for R600-style SIMD shader processor.
- (**C**, **infrastructure**) Designed tools to automate design, implementation, and verification of ASIC block interfaces, automate block-level drivers/monitors and comparison with software models, and manage block-level connectivity.
- (**C**, **infrastructure**) Developed languages, language semantics, and compilers to simplify design and verification of ASIC chips, and to support rapid development of independent backends providing multiple views of the chip.
- (**C++**, **testbench**) Developed test infrastructure, random/directed tests and software models for SIMD shader processors.
- (**C++**) Ported native MIPS test infrastructure for console chipset, including OS code, to Solaris and Linux to simplify verification process.

Technical Skills

Actively Used Languages

- C, C++ (*proficient*)
- Perl (*proficient*)
- OCaml (*proficient*)
- Bourne, C-Shell scripting
- M4 macro system
- HTML, CSS, Javascript
- LaTeX
- Assembly (x86, others)
- PHP (*basic knowledge*)
- PostgreSQL (*basic*)
- Ruby (*learning*).

Other Languages

- Java
- CLISP, Scheme
- Tcl/Tk
- Mathematica
- Matlab.

Tools and Systems

- Experienced with gcc, bison, yacc and doxygen for code development and documentation, and use of gdb and valgrind for debug.
- Experienced with Perforce revision control system including integration management.
- Extensive experience administering custom-built personal Linux machines since 1996.
- Comfortable with Windows, Microsoft Office Suite, Visio.
- Comfortable with Synopsys simulation tools.

Recreational Programming Projects

- Extensive recreational programming in MS-DOS, MS Windows, and Linux since 1992, developing games (both game engines and graphics), graphics demos and screensavers, and specialized languages.
- Created web-based interface for managing photos, and improved flows for photo processing.

Graduate Education

M.S., Computer Science June 2001 — June 2003
California Institute of Technology GPA: 4.2

Thesis: *Fault Tolerance using Whole-Process Migration and Speculative Execution*, advised by Jason Hickey.

- Use of language semantics and compiler implementation of mechanisms to support process migration across heterogeneous architectures, and speculative execution using compiler-managed checkpoints. These mechanisms were developed to accommodate fault tolerance in distributed systems.
- Researched both theory and implementation of reliable computing in faulty distributed networks.
- Developed x86 backend for Mojave Compiler (MCC), with emphasis on whole-process migration and speculative execution primitives. Implementation included work on novel compacting garbage collector.
- Designed and implemented low-level optimizations to make MCC competitive with established compilers.
- Developed comprehensive and interactive test suite for MCC.
- Developed theory behind and models for analyzing speculative execution using nestable transactions.

Teaching Assistant for *Compilers and Operating Systems* Sept. 2001 — June 2003
Prof. Jason Hickey, Caltech CS Department, Pasadena Head T.A. Sept. 2002 — June 2003

- Tutored students on OS concepts, including kernel, memory management, filesystems, threading, and security. Also, tutored students on compiler design including lexing, parsing, program transformation, assembly code generation, and garbage collection.
- Duties included leading recitations, grading assignments, and teaching lectures periodically.
- Designed and implemented multi-course grade book and homework/lab submission system.

Undergraduate Education

B.S. with Honors, Engineering and Applied Sciences Sept. 1997 — June 2001
California Institute of Technology GPA: 3.9

- Coursework included computer graphics and vision, operating systems, compiler theory, distributed systems, digital design, machine learning, network theory, computer algorithms.
- Work study as NT Server Developer integrating NT/UNIX servers, and as Student IT Representative administering heterogeneous network of machines located in student computer labs.

SURF Summer Research June 2000 — Aug. 2000
Prof. Jim Arvo, Caltech CS Department, Pasadena

- Enhanced design of a fixed-algorithm handwriting recognition system by adding support for multiple recognition modules. Implemented new recognition algorithms, analysis software, and display algorithms.

Teaching Assistant for *Introduction to Computation* Oct. 1999 — June 2001
Prof. Jim Arvo, Caltech CS Department, Pasadena Head T.A. Sept. 2000 — June 2001

- Tutored individual students on automata theory, complexity theory, numerical analysis and graphics. Provided instruction on LISP programming.
- Duties included grading assignments and laboratories, leading recitations and giving lectures.

Other Interests

- Japanese language studies, culture and history
- Abstract mathematical puzzles
- Design-for-reliability principles in other industries, e.g. aviation
- Drawing (pencil and ink)
- Photography (primarily outdoor landscape and macro photography)
- Hiking.