Roy Spliet, PhD.

Web: https://roy.spliet.org E-mail: roy@spliet.org Phone: Upon request

I am a Staff Research Engineer at Imagination Technologies, using a cycle-approximate performance model to evaluate novel ideas in the GPU architecture space. I am fascinated by the fundamental low-level building blocks that comprise a computer (operating systems, compilers and processor design), with a particular interest in data-parallel architectures like GPUs. I take a hands-on approach to problem solving, quick to acquire the practical skills required to systematically dissect a problem and prototype potential solutions.

Work experience

2020-present Staff Research Engineer, Imagination Technologies

- Implement and evaluate the performance of several research ideas using a cycle-approximate performance model.
- Add and improve support for common graphics techniques (e.g. MSAA, XFB) to the performance model, collaborating with compiler- and functional modelling teams.

Jul-Sep 2015 Intern, University of Cambridge Computer Lab

- Design lab exercises for the second year “Electronic Computer Aided Design and Architecture” course, assessing students’ understanding of Verilog, assembly and computer architecture.
- Implement a prototype logic design and Linux device driver to demonstrate the interaction between the Altera Cyclone V FPGA’s programmable logic and embedded ARM Cortex A9 core.

Mar-Jun 2015 Software Engineer, Ultimaker B.V.

- Develop commercial-grade embedded software in C++ for a prototype product, collaborating in a team of 3 to specify the interface between client software and the prototype’s firmware,
- Contribute to the bring-up of an Allwinner A20-based ARM board equipped with NAND storage using the Linux kernel and U-Boot, resulting in several upstream contributions to both.

Jan-Mar 2015 Temp. Customer Due Diligence, FGH bank

Jul-Sep 2014 Student, X.org Endless Vacation of Code

- Reverse-engineer the voltage- and frequency scaling hardware features found on NVIDIA GT2x0 graphics cards, including DRAM clock- and timing adjustment,
- Implement a static clock change mechanisms in the Nouveau open-source driver for NVIDIA hardware, resulting in an overall performance improvement for said graphics cards.

2013-2014 Research Intern, Max Planck Institute for Software Systems

- Research fast and practical implementations of common real-time synchronisation primitives, following the basic futex (fast user-space mutual exclusion) principles.

Sep-Dec 2012 Intern, ARM Ltd.

- Design and implement a prototype Linux kernel driver and trace decoding application for the CoreSight self-hosted trace and debug hardware found on most ARM-based SoCs.

2010, 2011 Part-time Developer (PHP, Drupal CMS), Sapito BV

2008-2009 Junior Developer (JAVA/ABAP), Super de Boer Supermarkets

Education

2015-2020: PhD in Computer Science, University of Cambridge (funding: lowRISC C.I.C.)

Dissertation: A SIMD architecture for hard real-time systems

Emerging safety-critical systems require high-performance data-parallel architectures that can guarantee tight and safe worst-case execution times. Given the complexity of existing architectures like GPUs, it is unlikely that sufficiently accurate models and timing analysis algorithms will emerge in the foreseeable future. In my dissertation I take a clean-slate approach to designing an efficient SIMD accelerator, implemented in SystemC as a cycleaccurate performance model, that permits derivation of safe and tight bounds on the execution time of data-parallel programs.

2010-2013: **Master of Science in Computer Engineering, Delft University of Technology**

Dissertation: A comprehensive study of Dynamic Memory Management in OpenCL kernels. (mark 9 - on a scale from 1 – very poor to 10 – outstanding).

Specialisation: General Purpose and High Performance Systems, comprising courses in architecture, compilers, parallel computing and real-time systems.

2006-2010: **Bachelor of Science in Computer Science, Delft University of Technology**

*Specialisation: S*oftware Technology, comprising courses in operating systems, embedded systems and algorithms.

2000-2006*:* *VWO Physics/Technology secondary education, Cals College*

Teaching

2016-2019: **University of Cambridge, Computer Lab**

Digital Electronics, first year undergraduate course, supervisions (teaching assistance),
Object-Oriented Programming, first year undergraduate course, supervisions,
Compiler Construction, second year undergraduate course, supervisions,
Electronic Computer Aided Design and Architecture, second year undergraduate lab assistance,
Algorithms, Sutton Trust summer school A-level pupils, supervisions.

Selected open-source contributions

Linux kernel	*Nouveau driver*: voltage- and frequency adjusting for various NVIDIA GPUs, *Allwinner SoC*: NAND storage driver improvements, *LITMUS^RT downstream real-time scheduling extensions*: ARM support and real-time futex implementations.
Ramulator	DDR3 and DDR4 timing model fixes.
GPGPU-Sim	Improvements to OpenCL support.
KMA	Code for a dynamic memory manager (malloc/free) in OpenCL C.

Selected publications

“Sim-D: A SIMD Accelerator for Hard Real-Time Systems”, R.Spliet, R.D. Mullins, in *IEEE Transactions on Computers, April 2022*

"The case for limited-preemptive scheduling for GPUs in hard real-time systems", R. Spliet, R.D. Mullins, in *Operating System Platforms for Embedded Real-Time Systems (OSPERT) 2018*

"Conquering the complexity mountain: Full-stack computer architecture teaching with FPGAs"*,* A.T. Markettos, S.W. Moore, B.D. Jones, R. Spliet, V.A. Gavrila, in *EWME 2018*

"Fast on Average, Predictable in the Worst Case: Exploring Real-Time Futexes in LITMUS^RT", R. Spliet, M. Vanga, B.B. Brandenburg, S. Dziadek, in RTSS *2014*

"KMA: A Dynamic Memory Manager for OpenCL", R. Spliet, A. Varbanescu, B.R. Gaster, L.W. Howes, in *GPGPU7 Workshop 2014*

Core skills

Programming	C, C++, Java, Assembly, HTML/CSS, PHP, SQL	
Parallel programming	OpenCL, pThreads, OpenMP	
Hardware modelling	SystemVerilog, SystemC, VHDL	
Revision control	GIT, Perforce, SVN	
Languages	*Dutch*	native
	English (UK)	fluent, ECRTS 8.0/CEFR C1
	French	basic understanding, CEFR B2