William Wang

M: +44 7895-323-686
A: 17 Seekings Close, Cambridge, United Kingdom
H: https://wilwan01.github.io

Computer architect with 16 years experience taking innovative ideas through to CPU architectures and microarchitectures. Extensive experience across computer system stacks from workloads characterization, simulator development, to architectural extensions and microarchitectural innovations. Strong understanding of the memory subsystem, a world-leading expert on memory persistency.

Education

Master of Philosophy | ***Engineering*** — Sept. 2004 – May 2006
University of Cambridge — Cambridge, England

Bachelor of Engineering | ***Electrical Engineering*** — Sept. 1999 – July 2003
Tongji University — Shanghai, China

Work Experience

Computer Architecture Researcher — April 2006 – Present
Arm Research — Cambridge, England

- Led the systems research on non-volatile memories at Arm. Responsibilities include: exploring use cases, understanding requirements, identifying gaps, and driving solutions. Recent focus was on addressing the persistent memory programming challenges with architectural and microarchitectural support in CPUs. Past investigations include: a) addressing the NVM endurance challenges, b) intermittent computing with NVM, and c) NVM as LLC.
- World-class impact both internally and externally. Internally I advised and transferred the technologies to Arm's chief architect for adoption into Arm architectures. Externally I am recognised as a world-leading expert on memory persistency. Served on the technical program committees of HPCA'20, NVMW'20, and NVMW'21 conferences. Served as reviewers for ACM TACO, IEEE CAL, and IEEE TETC journals. Invited talks/keynotes at Persistent Programming In Real Life (PIRL) 2020, Vail Computer Elements Workshop (VCEW) 2021, and Dagstuhl Seminar 21462 Foundations of Persistent Programming.
- Generated two dozen patents, covering processor instruction set architecture, processor microarchitecture, and memory systems. Published a dozen plus papers at top venues, including ISCA, HPCA, PLDI, FAST, SPAA, DAC, ISPASS, and IISWC. Language support for memory persistency was selected as IEEE Micro Top Picks 2019.
- Led a team of ten engineers and line managed a few. Co-supervised two PhD students at the University of Southampton and the University of Edinburgh. Collaborated with professors in computer architecture at leading universities across the Atlantic, including the University of Michigan. Led Arm participation in EU Horizon 2020 Project SAGE2 and SRC JUMP CRISP tasks.

Projects

Spatial architecture | ***Architecture, μArch, Workloads*** — 2021-2022

- Exploring spatial architecture for general-purpose acceleration to provide 10x performance density over CPU with Scalable Matrix Extension. The exploratory architectural features include dataflow instruction set architecture, decoupled access and execution, reconfigurable distributed memory, and upcoming support for sparsity. The spatial architecture comprises of many ISA-based processing elements.
- Proved 10x performance density for use cases in infrastructure such as 5G and AI. Handcrafted kernels such as dense matrix multiplication and cholesky decomposition in co-designed triggered assembly, applied techniques such as tiling, vectorizing, and pipelining to spatially map the triggered assembly onto the spatial architecture.
- Worked in close collaboration with team on ISA development, spatial architecture design space exploration, simulator extension, hardware implementation, and performance density evaluation.

Architectural support for persistent memory | *Architecture, Microarchitecture* 2018-2021

- World-leading research on memory persistency models. Architectural support for a spectrum of persistency models, ranging from the more relaxed release persistency and strand persistency to the less relaxed strict persistency and sequential persistency. Comprehensive coverage of persistency models across the vertical stacks too at the programming language, ISA, system architecture, and microarchitecture levels.
- World-class impact externally with more than a dozen papers published at top venues, including PLDI'18, IEEE Micro Top Picks 2019, SPAA'19, ISCA'20, and HPCA'21. Invited talks/keynotes at PIRL'20, VCEW'21, and Dagstuhl Seminar 21462.
- Industry-leading impact internally with more than a dozen IPs patented and such IPs are impacting Arm architectures. Influenced Arm's persistence support across various interconnects, including Arm's AMBA CHI protocols and industry wide protocols such as NVDIMM-P, Gen-Z, CCIX, PCIe, and CXL.

Exploring NVM as last-level cache | *Microarchitecture* 2019

- Explored on-chip uses of NVM as a) part of application processor cache hierarchy, b) part of on-chip memory in MCUs, and c) part of on-chip memory in accelerators.
- Recommendations and learnings disseminated to Arm's internal physical design IP division as well as CPU design divisions, with quantified numbers on power, performance, area, endurance and retention.
- Led a team of four engineers across architecture and silicon.

Addressing the endurance challenge of NVM | *Microarchitecture, Architecture, Software* 2015-2017

- Proposed software managed wear-levelling. Explored NVM writes reduction and distribution techniques, including compression between LLC and NVM to reduce NVM writes, replacement policies that keep dirty data in caches for longer to reduce NVM writes, and architectural support for dead writes elimination to reduce unnecessary NVM writes. Knowledge was transferred to CeRAM/MRAM teams within Arm. Some results were presented at FAST'19 and MEMSYS'17.

Pushing Arm servers into healthcare | *Software, Architecture* 2016-2017

- Led the technical efforts across Arm ecosystem for pushing Arm servers into the healthcare sector. Worked closely with Arm marketing/engineering, Arm server CPU vendors, Arm server vendors, and customers in the pharmaceutical and biotechnology sectors.
- Optimized genomics software pipelines on Arm, including optimising: a) glibc function implementations on Arm, b) bioinformatics kernels such as Banded Smith-Waterman and Hidden Markov Model with Arm SIMD, as well as with customised accelerators in partner SoCs such as for compression, c) tools such as Arm HPC Compiler for auto-vectorization with NEON/SVE. This work was selected for demonstration at Arm's Annual Partners Meeting (APM) 2017.
- Explored scalable vector ISA extension for genomics workloads. The bit shuffle instructions were added to SVE2 as part of Armv9.
- Worked with Arm's New Business Ventures group on acquisitions in this space. Worked on the business case presented to the Arm Board with business strategy, market sizing, budgeting, recruiting, and return on investment scenario analysis.

Data profiling tool for exposing data movements in memory | *Software, Microarchitecture* 2013-2015

- Data profiling tool for exposing and then optimizing data movements to improve systems performance and energy efficiency. I developed a data profiler in C++ to expose how software interacts with CPU data caches, enabling software developers to identify hot data structures, reorganize data structures or data access patterns to reduce data cache misses. The project was presented at ARM's Global Engineering Conference 2014, and won the Best Poster Award at MSPC 2014 workshop. The tool prototype was integrated into ARM's flagship performance analysis tool DS-5 Streamline.

Android Dalvik virtual machine performance analysis and optimizations | *Runtime, Compilers*2012-2013

- System software competitive performance analysis and compiler optimizations. This study answered the question whether Arm should look at optimizing the performance of Dalvik virtual machine on Arm. I analyzed the Android Dalvik code base in C++ to identify gaps between Dalvik implementation and Java SE Embedded C1/C2 implementation, and recommended Dalvik JIT compiler optimizations to stakeholders.

gem5 simulator development & mobile systems research | *Simulator, Workloads & µArch* 2010-2012

- gem5 is an open source full-system simulator for computer systems research. I developed modules in C++ to extend the simulator support for Arm CPUs and systems, including adding the Arm GDB client, VNC (X11) client, Armv8 architecture verification and bug-fixing, Armv8 Linux kernel bring-up, classic memory system restructuring, performance optimizations, and the Android workload image creation. I studied the impact of memory subsystem on the performance of smartphone applications with the simulation infrastructure.
- gem5 supported running realistic Android workloads on Armv8 systems. Identified many microarchitectural areas for improvement which were not shown by conventional benchmarks, e.g., the large instruction footprint of Android workloads and the need to reduce instruction cache misses. Some of this work was presented in IISWC'13, MEMSYS'15, and arXiv'20.

Coarse-grained reconfigurable architecture | *Workloads & Design Space Exploration* 2008-2010

- The project aimed at evaluating the feasibility of designing a coarse-grained reconfigurable architecture that could compete with bit-configurable FPGA in terms of PPA.
- Workloads development and design space exploration. I developed a hardware benchmark suite with 25 applications for evaluating coarse-grained reconfigurable architectures. The applications ranged from quantitative finance to wireless communication, for instance, Black-Scholes, FFT and Viterbi Decoder were implemented in Verilog and C for customized hardware and CPU.
- The benchmark suite successfully aided in the CGRA architectural design space exploration, by showing how CGRA designs compare to CPU, FPGA, and ASIC in terms of PPA across a diverse spectrum of workloads in a fully automated fashion.

Razor: dynamic voltage scaling based on timing speculation | *SoC Prototyping & Verification* 2006-2008

- The Razor idea was originally developed at the University of Michigan that aimed at shaving voltage margins to save power. The original work received the 2021 Micro Test of Time award. The low-power technology was developed and proved further by us at Arm Research.
- Razor SoC design, prototyping and OS bring-up. Designed the Razor prototype SoC based on ARM1176JZF-S and implemented the SoC in FPGA. Wrote low-level software in C and Arm assembly to verify the prototype, and ran benchmarks to measure the performance. Built the Linux Kernel for the platform and brought up workloads on the platform.
- The prototype contributed to proving the viability of the Razor technology with Arm CPUs. The Razor technology was then transferred to Arm CPU product groups.

Publications and Presentations

Dagstuhl Seminar 21462	*Invited Talk:* Architectural Support for Persistent Programming
VCEW'21	*Invited Talk:* Architectural Support for Persistent Memory
HPCA'21	BBB: Simplifying Persistent Programming Using Battery-Backed Buffers
PIRL'20	*Invited Keynote:* Architectural Support for Persistent Programming
arXiv'20	The gem5 Simulator: Version 20.0+
ISCA'20	Relaxed Persist Ordering Using Strand Persistency
ISPASS'20	Fused: Closed-loop Performance and Energy Simulation of Embedded Systems
SPAA'19	Persistent Atomics for Implementing Durable Lock-Free Data Structures for Non-Volatile Memory
IEEE Micro Top Picks 2019	Language Support for Memory Persistency
FAST'19	Software Wear Management for Persistent Memories
DAC'19	Efficient State Retention Through Paged Memory Management for Reactive Transient Computing
PLDI'18	Persistency for Synchronization-Free Regions
MEMSYS'18	Quantifying the Performance Overheads of PMDK
MEMSYS'17	Composing Lifetime Enhancing Techniques for Non-Volatile Main Memories
MEMSYS'15	Inefficiencies in the Cache Hierarchy: A Sensitivity Study of Cacheline Size with Mobile Workloads
IISWC'13	A Structured Approach to the Simulation, Analysis and Characterization of Smartphone Applications

Patents

UK2105240.2 Apparatus and Method for Generating Debug Information

US17,129,515 Systems and Methods for Defining and Enforcing Ordered Constraints

US17,069,057 Draining Operation to Cause Store Data to be Written to Persistent Memory

UK2014440.8 Draining Operation for Draining Dirty Cache Lines to Persistent Memory

US16,882,402 Hardware Bitvector Techniques

UK2001782.8 Task-Aware Checkpointing for Intermittent Compute

US10,956,166 Instruction Ordering

US10,866,890 Method and Apparatus for Implementing Lock-Free Data Structures

US10,445,238 Robust Transactional Memory

US11,182,106 Energy Conservation for Memory Applications

US10,831,678 Multi-Tier Cache Placement Mechanism

US11,137,919 Initialisation of a Storage Device

US10,642,743 Apparatus and Method of Handling Caching of Persistent Data

US10,964,386 Initialisation of a Storage Device

US10,847,204 Control of Refresh Operation for Memory Regions

US10,719,236 Non-Volatile Buffer for Memory Operations

Honors and Awards

Language Support for Memory Persistency was selected as IEEE Micro Top Picks 2019 May 2019
Recognition for the top 12 most influential publications of the year in computer architecture

Best Poster Award at MSPC 2014 June 2014
DataProf: Exposing Data Movements Across the Memory Hierarchy

Arm Multiple Patent Grant Award - David 2020, 2021
Awarded for every four granted US patents

Services and Community

Member of HiPEAC, Member of SNIA NVM Programming TWG

Technical Program Committee of HPCA'20, NVMW'20, NVMW'21

Organizing Committee of IISWC'20

Organizing Committee, Technical Program Committee and Session Chair for Arm Research Summit 2016-2020

Reviewer for IEEE Computer Architecture Letters (CAL), IEEE Transactions on Emerging Topics in Computing (TETC), and ACM Transactions on Architecture and Code Optimization (TACO)

Organizer of Architectural Exploration with Gem5 Tutorial at ASPLOS 2017

Member of Arm Patents Review Committee (PRC)