ISCA Tutorial Reading List

Caroline Trippel Yatin A. Manerkar Margaret Martonosi
Princeton University
{ctrippel,manerkar,mrm}@princeton.edu
June, 2017

References

[1] Sarita Adve and Kourosh Gharachorloo. Shared memory consistency models: A tutorial. *IEEE Computer*, 29(12):66–76, 1996.

[2] Sarita V. Adve and Mark D. Hill. Weak ordering—a new definition. In *Proceedings of the 17th Annual International Symposium on Computer Architecture*, ISCA '90, pages 2–14, New York, NY, USA, 1990. ACM.

[3] Jade Alglave, Mark Batty, Alastair F. Donaldson, Ganesh Gopalakrishnan, Jeroen Ketema, Daniel Poetzl, Tyler Sorensen, and John Wickerson. Gpu concurrency: Weak behaviours and programming assumptions. In *Proceedings of the Twentieth International Conference on Architectural Support for Programming Languages and Operating Systems*, ASPLOS '15, pages 577–591, New York, NY, USA, 2015. ACM.

[4] Jade Alglave, Luc Maranget, and Michael Tautschnig. Herding cats: Modelling, simulation, testing, and data mining for weak memory. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, 36(2):7:1–7:74, July 2014.

[5] ARM. ARM Cortex-A9 technical reference manual ARMv7-A, 2008-2012. http://infocenter.arm.com/help/topic/com.arm.doc.ddi0388i/DDI0388I_cortex_a9_r4p1_trm.pdf.

[6] ARM. Cortex-A9 MPCore, programmer advice notice, read-after-read hazards. ARM Reference 761319., 2011. http://infocenter.arm.com/help/topic/com.arm.doc.uan0004a/UAN0004A_a9_read_read.pdf.

[7] ARM. ARM Cortex-A series programmers guide for ARMv8-A, section 13.2.1, one-way barriers, 2015. http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.den0024a/CHDCJBGA.html.

[8] Arvind and Jan-Willem Maessen. Memory model = instruction reordering + store atomicity. In *Proceedings of the 33rd Annual International Symposium on Computer Architecture*, ISCA '06, pages 29–40. IEEE Computer Society, 2006.

[9] Mark Batty, Alastair F. Donaldson, and John Wickerson. Overhauling SC atomics in C11 and OpenCL. In *43rd Annual Symposium on Principles of Programming Languages (POPL)*, 2016.

[10] Mark Batty, Kayvan Memarian, Scott Owens, Susmit Sarkar, and Peter Sewell. Clarifying and compiling C/C++ concurrency: From C++11 to POWER. In *39th Annual Symposium on Principles of Programming Languages (POPL)*, 2012.

[11] Mark Batty, Scott Owens, Susmit Sarkar, Peter Sewell, and Tjark Weber. Mathematizing C++ concurrency. In *38th Annual Symposium on Principles of Programming Languages (POPL)*, 2011.

[12] Colin Blundell, Milo M.K. Martin, and Thomas F. Wenisch. InvisiFence: Performance-transparent memory ordering in conventional multiprocessors. In *36th Annual International Symposium on Computer Architecture (ISCA)*, 2009.

[13] Hans-J. Boehm. Threads cannot be implemented as a library. In *Proceedings of the 2005 ACM SIGPLAN Conference on Programming Language Design and Implementation*, PLDI '05, pages 261–268, New York, NY, USA, 2005. ACM.

[14] Hans-J. Boehm. Position paper: Nondeterminism is unavoidable, but data races are pure evil. In *Workshop on Relaxing Synchronization for Multicore and Manycore Scalability (RACES)*, 2012.

[15] Hans-J. Boehm and Sarita V. Adve. Foundations of the C++ concurrency memory model. In *29th Conference on Programming Language Design and Implementation (PLDI)*, 2008.

[16] Hans-J. Boehm and Brian Demsky. Outlawing ghosts: Avoiding out-of-thin-air results. In *Proceedings of the Workshop on Memory Systems Performance and Correctness*, MSPC '14, pages 7:1–7:6, New York, NY, USA, 2014. ACM.

[17] Luis Ceze, James Tuck, Pablo Montesinos, and Josep Torrellas. BulkSC: Bulk enforcement of sequential consistency. In *34th Annual International Symposium on Computer Architecture (ISCA)*, 2007.

[18] William W. Collier. *Reasoning About Parallel Architectures*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1992.

[19] M. Elver and V. Nagarajan. TSO-CC: Consistency directed cache coherence for TSO. In *20th International Symposium on High Performance Computer Architecture (HPCA)*, 2014.

[20] M. Elver and V. Nagarajan. Mcversi: A test generation framework for fast memory consistency verification in simulation. In *2016 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, pages 618–630, March 2016.

[21] Shaked Flur, Kathryn E. Gray, Christopher Pulte, Susmit Sarkar, Ali Sezgin, Luc Maranget, Will Deacon, and Peter Sewell. Modelling the armv8 architecture, operationally: Concurrency and isa. In *Proceedings of the 43rd Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, POPL '16, pages 608–621, New York, NY, USA, 2016. ACM.

[22] Kourosh Gharachorloo. *Memory Consistency Models for Shared-memory Multiprocessors*. PhD thesis, Stanford University, Stanford, CA, USA, 1996.

[23] Kourosh Gharachorloo, Anoop Gupta, and John Hennessy. Two techniques to enhance the performance of memory consistency models. In *In Proceedings of the 1991 International Conference on Parallel Processing*, pages 355–364, 1991.

[24] Kourosh Gharachorloo, Daniel Lenoski, James Laudon, Phillip Gibbons, Anoop Gupta, and John Hennessy. Memory consistency and event ordering in scalable shared-memory multiprocessors. *17th International Symposium on Computer Architecture (ISCA)*, 1990.

[25] Chris Gniady and Babak Falsafi. Speculative sequential consistency with little custom storage. In *International Conference on Parallel Architectures and Compilation Techniques (PACT)*, 2002.

[26] Chris Gniady, Babak Falsafi, and T.N. Vijaykumar. Is SC + ILP = RC? *41st International Symposium on Computer Architecture (ISCA)*, 1999.

[27] Dibakar Gope and Mikko H. Lipasti. Atomic SC for simple in-order processors. In *20th International Symposium on High Performance Computer Architecture HPCA*, 2014.

[28] Kathryn E. Gray, Gabriel Kerneis, Dominic Mulligan, Christopher Pulte, Susmit Sarkar, and Peter Sewell. An integrated concurrency and core-isa architectural envelope definition, and test oracle, for ibm power multiprocessors. In *Proceedings of the 48th International Symposium on Microarchitecture*, MICRO-48, pages 635–646, New York, NY, USA, 2015. ACM.

[29] Martonosi Research Group. Check research tools and papers website, 2017. http://check.cs.princeton.edu.

[30] Sudheendra Hangal, Durgam Vahia, Chaiyasit Manovit, and Juin-Yeu Joseph Lu. Tsotool: A program for verifying memory systems using the memory consistency model. In *Proceedings of the 31st Annual International Symposium on Computer Architecture*, ISCA '04, pages 114–, 2004.

[31] Sudheendra Hangal, Durgam Vahia, Chaiyasit Manovit, and Juin-Yeu Joseph Lu. TSOtool: A program for verifying memory systems using the memory consistency model. In *31st Annual International Symposium on Computer Architecture (ISCA)*, 2004.

[32] Derek R. Hower, Blake A. Hechtman, Bradford M. Beckmann, Benedict R. Gaster, Mark D. Hill, Steven K. Reinhardt, and David A. Wood. Heterogeneous-race-free memory models. In *19th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, 2014.

[33] ISO/IEC. Programming Languages – C. International standard 9899:2011, 2011.

[34] ISO/IEC. Programming Languages – C++. International standard 14882:2011, 2011.

[35] ISO/IEC. Programming Languages – C++, 2014.

[36] Pete Keleher, Alan L. Cox, and Willy Zwaenepoel. Lazy release consistency for software distributed shared memory. In *19th Annual International Symposium on Computer Architecture*, 1992.

[37] Ori Lahav, Nick Giannarakis, and Viktor Vafeiadis. Taming release-acquire consistency. In *Proceedings of the 43rd Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, POPL '16, pages 649–662, New York, NY, USA, 2016. ACM.

[38] Ori Lahav, Viktor Vafeiadis, Jeehoon Kang, Chung-Kil Hur, and Derek Dreyer. Repairing sequential consistency in C/C++11. *MPI-SWS*, Tech. rep. MPI-SWS-2016-011, 2016.

[39] Leslie Lamport. How to make a multiprocessor computer that correctly executes multiprocess programs. *IEEE Transactions on Computing*, 28(9):690–691, 1979.

[40] Nhat Minh Lê, Antoniu Pop, Albert Cohen, and Francesco Zappa Nardelli. Correct and efficient work-stealing for weak memory models. In *18th Symposium on Principles and Practice of Parallel Programming (PPoPP)*, 2013.

[41] Changhui Lin, Vijay Nagarajan, Rajiv Gupta, and Bharghava Rajaram. Efficient sequential consistency via conflict ordering. In *17th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, 2012.

[42] D. Lustig, G. Sethi, A. Bhattacharjee, and M. Martonosi. Transistency models: Memory ordering at the hardware-os interface. *IEEE Micro*, 37(3):88–97, 2017.

[43] Daniel Lustig, Michael Pellauer, and Margaret Martonosi. PipeCheck: Specifying and verifying microarchitectural enforcement of memory consistency models. In *47th International Symposium on Microarchitecture (MICRO)*, 2014.

[44] Daniel Lustig, Geet Sethi, Margaret Martonosi, and Abhishek Bhattacharjee. COATCheck: Verifying Memory Ordering at the Hardware-OS Interface. In *Proceedings of the 21st International Conference on Architectural Support for Programming Languages and Operating Systems*, 2016.

[45] Daniel Lustig, Caroline Trippel, Michael Pellauer, and Margaret Martonosi. ArMOR: Defending against memory consistency model mismatches in heterogeneous architectures. In *42nd International Symposium on Computer Architecture (ISCA)*, 2015.

[46] Daniel Lustig, Andrew Wright, Alexandros Papakonstantinou, and Olivier Giroux. Automated synthesis of comprehensive memory model litmus test suites. *22nd International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, 2017.

[47] Sela Mador-Haim, Luc Maranget, Susmit Sarkar, Kayvan Memarian, Jade Alglave, Scott Owens, Rajeev Alur, Milo M. K. Martin, Peter Sewell, and Derek Williams. An axiomatic memory model for POWER multiprocessors. In *24th International Conference on Computer Aided Verification (CAV)*, 2012.

[48] Yatin A. Manerkar, Daniel Lustig, Michael Pellauer, and Margaret Martonosi. CCICheck: Using μhb graphs to verify the coherence-consistency interface. In *48th International Symposium on Microarchitecture (MICRO)*, 2015.

[49] Yatin A. Manerkar, Caroline Trippel, Daniel Lustig, Michael Pellauer, and Margaret Martonosi. Counterexamples and proof loophole for the C/C++ to POWER and armv7 trailing-sync compiler mappings. *CoRR*, abs/1611.01507, 2016.

[50] Luc Maranget, Susmit Sarkar, and Peter Sewell. A tutorial introduction to the arm and power relaxed memory models (2012), 2012.

[51] Milo M. K. Martin, Daniel J. Sorin, Harold W. Cain, Mark D. Hill, and Mikko H. Lipasti. Correctly implementing value prediction in microprocessors that support multithreading or multiprocessing. In *34th International Symposium on Microarchitecture (MICRO)*, 2001.

[52] Paul E. McKenney and Raul Silvera. Example POWER implementation for C/C++ memory model, 2011. http://www.rdrop.com/users/paulmck/scalability/paper/N2745r.2011.03.04a.html.

[53] Albert Meixner and Daniel J. Sorin. Dynamic verification of memory consistency in cache-coherent multithreaded computer architectures. *IEEE Trans. Dependable Secur. Comput.*, 6(1):18–31, January 2009.

[54] Kyndylan Nienhuis, Kayvan Memarian, and Peter Sewell. An operational semantics for c/c++11 concurrency. In *Proceedings of the 2016 ACM SIGPLAN International Conference on Object-Oriented Programming, Systems, Languages, and Applications*, OOPSLA 2016, pages 111–128, New York, NY, USA, 2016. ACM.

[55] Brian Norris and Brian Demsky. Cdschecker: Checking concurrent data structures written with c/c++ atomics. In *Proceedings of the 2013 ACM SIGPLAN International Conference on Object Oriented Programming Systems Languages & Applications*, OOPSLA '13, pages 131–150, New York, NY, USA, 2013. ACM.

[56] OpenRisc Community. OpenRisc, 2016. http://openrisc.io.

[57] Scott Owens, Susmit Sarkar, and Peter Sewell. A better x86 memory model: x86-TSO. In *22nd International Conference on Theorem Proving in Higher Order Logics (TPHOLs)*, 2009.

[58] Gustavo Petri, Jan Vitek, and Suresh Jagannathan. Cooking the books: Formalizing JMM implementation recipes. In *29th European Conference on Object-Oriented Programming (ECOOP)*, 2015.

[59] Parthasarathy Ranganathan, Vijay S. Pai, and Sarita V. Adve. Using speculative retirement and larger instruction windows to narrow the performance gap between memory consistency models. In *9th Symposium on Parallel Algorithms and Architectures (SPAA)*, 1997.

[60] RISC-V Foundation. RISC-V port of Linux kernel, 2016. https://github.com/riscv/riscv-linux/blob/master/arch/riscv/include/asm/barrier.h.

[61] Susmit Sarkar, Kayvan Memarian, Scott Owens, Mark Batty, Peter Sewell, Luc Maranget, Jade Alglave, and Derek Williams. Synchronising C/C++ and POWER. In *33rd Conference on Programming Language Design and Implementation (PLDI)*, 2012.

[62] Susmit Sarkar, Peter Sewell, Jade Alglave, Luc Maranget, and Derek Williams. Understanding power multiprocessors. In *Proceedings of the 32Nd ACM SIGPLAN Conference on Programming Language Design and Implementation*, PLDI '11, pages 175–186, New York, NY, USA, 2011. ACM.

[63] David Seal. *ARM Architecture Reference Manual*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2nd edition, 2000.

[64] Peter Sewell. C/c++11 mappings to processors. 2016.

[65] Abhayendra Singh, Satish Narayanasamy, Daniel Marino, Todd Millstein, and Madanlal Musuvathi. End-to-end sequential consistency. In *39th International Symposium on Computer Architecture (ISCA)*, 2012.

[66] Daniel J. Sorin, Mark D. Hill, and David A. Wood. *A Primer on Memory Consistency and Cache Coherence*. Morgan & Claypool Publishers, 1st edition, 2011.

[67] SPARC International. *The SPARC Architecture Manual: Version 8*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1992.

[68] SPARC International. *The SPARC Architecture Manual (Version 9)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1994.

[69] J. M. Tendler, J. S. Dodson, J. S. Fields, H. Le, and B. Sinharoy. POWER4 system microarchitecture. *IBM Journal of Research and Development*, 46(1):5–25, January 2002.

[70] Linus Torvalds et al. Linux kernel, 2016. https://github.com/torvalds/linux/blob/master/arch/alpha/include/asm/barrier.h.

[71] Viktor Vafeiadis, Thibaut Balabonski, Soham Chakraborty, Robin Morisset, and Francesco Zappa Nardelli. Common compiler optimisations are invalid in the C11 memory model and what we can do about it. In *42nd Symposium on Principles of Programming Languages (POPL)*, 2015.

[72] Viktor Vafeiadis and Chinmay Narayan. Relaxed separation logic: A program logic for C11 concurrency. In *28th International Conference on Object Oriented Programming Systems Languages and Applications (OOPSLA)*, 2013.

[73] Andrew Waterman, Yunsup Lee, David A. Patterson, and Krste Asanovic. The RISC-V instruction set manual, volume I: User-level ISA, version 2.1. Technical Report UCB/EECS-2016-118, EECS Department, University of California, Berkeley, May 2016.

[74] Thomas F. Wenisch, Anastasia Ailamaki, Babak Falsafi, and Andreas Moshovos. Mechanisms for store-wait-free multiprocessors. In *34th International Symposium on Computer Architecture (ISCA)*, 2007.

[75] John Wickerson, Mark Batty, Tyler Sorensen, and George A Constantinides. Automatically comparing memory consistency models. *44th Symposium on Principles of Programming Languages (POPL)*, 2017.

[76] Meng Zhang, Jesse D. Bingham, John Erickson, and Daniel J. Sorin. Pvcoherence: Designing flat coherence protocols for scalable verification. *2014 IEEE 20th International Symposium on High Performance Computer Architecture (HPCA)*, pages 392–403, 2014.

[77] Meng Zhang, Alvin Lebeck, and Daniel Sorin. Fractal consistency: Architecting the memory system to facilitate verification. *IEEE Comput. Archit. Lett.*, 9(2):61–64, July 2010.

[78] Meng Zhang, Alvin R. Lebeck, and Daniel J. Sorin. Fractal coherence: Scalably verifiable cache coherence. In *Proceedings of the 2010 43rd Annual IEEE/ACM International Symposium on Microarchitecture*, MICRO '43, pages 471–482, Washington, DC, USA, 2010. IEEE Computer Society.

[79] Sizhuo Zhang, Arvind, and Muralidaran Vijayaraghavan. Taming weak memory models. *CoRR*, abs/1606.05416, 2016.