



Bibliography

- [ccs,] Creative commons sharealike license. <https://creativecommons.org/licenses/by-sa/4.0/legalcode>. Accessed on 5th Feb 2019.
- [Abadi et al., 2016] Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al. (2016). Tensorflow: A system for large-scale machine learning. In *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, pages 265–283.
- [Abts et al., 2003] Abts, D., Scott, S., and Lilja, D. J. (2003). So many states, so little time: Verifying memory coherence in the cray x1. In *Parallel and Distributed Processing Symposium, 2003. Proceedings. International*, pages 10–pp. IEEE.
- [Advanced Micro Devices, 2017] Advanced Micro Devices (2017). *Software Optimization Guide for AMD Family 17h Processors*.
- [Adve, 1993] Adve, S. V. (1993). *Designing memory consistency models for shared-memory multiprocessors*. PhD thesis, University of Wisconsin-Madison.
- [Adve and Gharachorloo, 1996] Adve, S. V. and Gharachorloo, K. (1996). Shared memory consistency models: A tutorial. *Computer*, 29(12):66–76.
- [Agarwal et al., 2009] Agarwal, N., Krishna, T., Peh, L.-S., and Jha, N. K. (2009). Garnet: A detailed on-chip network model inside a full-system simulator. In *2009 IEEE international symposium on performance analysis of systems and software*, pages 33–42. IEEE.
- [Aho, 2003] Aho, A. V. (2003). *Compilers: principles, techniques and tools (for Anna University)*, 2/e. Pearson Education India.
- [Aho and Ullman, 1977] Aho, A. V. and Ullman, J. D. (1977). *Principles of Compiler Design (Addison-Wesley series in computer science and information processing)*. Addison-Wesley Longman Publishing Co., Inc.
- [Akinaga and Shima, 2012] Akinaga, H. and Shima, H. (2012). Reram technology; challenges and prospects. *IEICE Electronics Express*, 9(8):795–807.

- [Akkary et al., 2003] Akkary, H., Rajwar, R., and Srinivasan, S. T. (2003). Checkpoint processing and recovery: Towards scalable large instruction window processors. In *Microarchitecture, 2003. MICRO-36. Proceedings. 36th Annual IEEE/ACM International Symposium on*, pages 423–434. IEEE.
- [Albericio et al., 2016] Albericio, J., Judd, P., Hetherington, T., Aamodt, T., Jerger, N. E., and Moshovos, A. (2016). Cnvlutin: ineffectual-neuron-free deep neural network computing. In *Proceedings of the 43rd International Symposium on Computer Architecture*, pages 1–13.
- [Alglave, 2012] Alglave, J. (2012). A formal hierarchy of weak memory models. *Formal Methods in System Design*, 41(2):178–210.
- [Alpern et al., 2005] Alpern, B., Augart, S., Blackburn, S. M., Butrico, M., Cocchi, A., Cheng, P., Dolby, J., Fink, S., Grove, D., Hind, M., et al. (2005). The jikes research virtual machine project: building an open-source research community. *IBM Systems Journal*, 44(2):399–417.
- [Alpert and Avnon, 1993] Alpert, D. and Avnon, D. (1993). Architecture of the pentium microprocessor. *IEEE micro*, 13(3):11–21.
- [Anis and Nicolici, 2007] Anis, E. and Nicolici, N. (2007). On using lossless compression of debug data in embedded logic analysis. In *2007 IEEE International Test Conference*, pages 1–10. IEEE.
- [Annaram et al., 2003] Annaram, M., Patel, J. M., and Davidson, E. S. (2003). Call graph prefetching for database applications. *ACM Transactions on Computer Systems (TOCS)*, 21(4):412–444.
- [Apalkov et al., 2013] Apalkov, D., Khvalkovskiy, A., Watts, S., Nikitin, V., Tang, X., Lottis, D., Moon, K., Luo, X., Chen, E., Ong, A., Driskill-Smith, A., and Krounbi, M. (2013). Spin-transfer torque magnetic random access memory (stt-mram). *J. Emerg. Technol. Comput. Syst.*, 9(2):13:1–13:35.
- [Arora et al., 2015] Arora, A., Harne, M., Sultan, H., Bagaria, A., and Sarangi, S. R. (2015). Fp-nuca: A fast noc layer for implementing large nuca caches. *IEEE Transactions on Parallel and Distributed Systems*, 26(9):2465–2478.
- [Arvind and Maessen, 2006] Arvind, A. and Maessen, J. (2006). Memory model= instruction reordering+ store atomicity. In *Proceedings. 33rd International Symposium on Computer Architecture*, pages 29–40.
- [Austin, 1999] Austin, T. M. (1999). Diva: A reliable substrate for deep submicron microarchitecture design. In *MICRO-32. Proceedings of the 32nd Annual ACM/IEEE International Symposium on Microarchitecture*, pages 196–207. IEEE.
- [Bakhoda et al., 2009] Bakhoda, A., Yuan, G. L., Fung, W. W., Wong, H., and Aamodt, T. M. (2009). Analyzing cuda workloads using a detailed gpu simulator. In *Performance Analysis of Systems and Software, 2009. ISPASS 2009. IEEE International Symposium on*, pages 163–174. IEEE.
- [Balasubramonian et al., 2011] Balasubramonian, R., Jouppi, N. P., and Muralimanohar, N. (2011). Multi-core cache hierarchies. *Synthesis Lectures on Computer Architecture*, 6(3):1–153.
- [Bashir et al., 2019] Bashir, J., Peter, E., and Sarangi, S. R. (2019). A survey of on-chip optical interconnects. *ACM Comput. Surv.*, 51(6):115:1–115:34.
- [Baumann, 2005] Baumann, R. C. (2005). Radiation-induced soft errors in advanced semiconductor technologies. *IEEE Transactions on Device and materials reliability*, 5(3):305–316.
- [Bekerman et al., 2000] Bekerman, M., Yoaz, A., Gabbay, F., Jourdan, S., Kalaev, M., and Ronen, R. (2000). Early load address resolution via register tracking. In *Proceedings of the 27th Annual International Symposium on Computer Architecture*, pages 306–315.

- [Bellard, 2005] Bellard, F. (2005). Qemu, a fast and portable dynamic translator. In *USENIX Annual Technical Conference, FREENIX Track*, volume 41, page 46.
- [Benini et al., 1999] Benini, L., Macii, A., Macii, E., and Poncino, M. (1999). Selective instruction compression for memory energy reduction in embedded systems. In *Proceedings of the 1999 international symposium on Low power electronics and design*, pages 206–211. ACM.
- [Bergstra et al., 2010] Bergstra, J., Breuleux, O., Bastien, F., Lamblin, P., Pascanu, R., Desjardins, G., Turian, J., Warde-Farley, D., and Bengio, Y. (2010). Theano: A cpu and gpu math compiler in python. In *Proc. 9th Python in Science Conf*, volume 1.
- [Bernick et al., 2005] Bernick, D., Bruckert, B., Vigna, P. D., Garcia, D., Jardine, R., Klecka, J., and Smullen, J. (2005). Nonstop advanced architecture. In *2005 International Conference on Dependable Systems and Networks (DSN'05)*, pages 12–21. IEEE.
- [Bienia et al., 2008] Bienia, C., Kumar, S., Singh, J. P., and Li, K. (2008). The parsec benchmark suite: Characterization and architectural implications. In *Proceedings of the 17th international conference on Parallel architectures and compilation techniques*, pages 72–81. ACM.
- [Bishop, 2006] Bishop, C. M. (2006). *Pattern recognition and machine learning*. springer.
- [Bjerregaard and Mahadevan, 2006] Bjerregaard, T. and Mahadevan, S. (2006). A survey of research and practices of network-on-chip. *ACM Computing Surveys (CSUR)*, 38(1):1.
- [Black, 1969] Black, J. R. (1969). Electromigration—a brief survey and some recent results. *IEEE Transactions on Electron Devices*, 16(4):338–347.
- [Blythe, 2008] Blythe, D. (2008). Rise of the graphics processor. *Proceedings of the IEEE*, 96(5):761–778.
- [Bodin and Sezneć, 1997] Bodin, F. and Sezneć, A. (1997). Skewed associativity improves program performance and enhances predictability. *IEEE transactions on Computers*, 46(5):530–544.
- [Bogdanov et al., 2007] Bogdanov, A., Knudsen, L. R., Leander, G., Paar, C., Poschmann, A., Robshaw, M. J., Seurin, Y., and Vikkelsoe, C. (2007). Present: An ultra-lightweight block cipher. In *International Workshop on Cryptographic Hardware and Embedded Systems*, pages 450–466. Springer.
- [Brooks et al., 2000] Brooks, D., Tiwari, V., and Martonosi, M. (2000). Wattch: a framework for architectural-level power analysis and optimizations. In *Proceedings of the 27th annual International Symposium on Computer Architecture*, pages 83–94.
- [Brown et al., 2001] Brown, M. D., Stark, J., and Patt, Y. N. (2001). Select-free instruction scheduling logic. In *Microarchitecture, 2001. MICRO-34. Proceedings. 34th ACM/IEEE International Symposium on*, pages 204–213. IEEE.
- [Budde et al., 1990] Budde, D., Riches, R., Imel, M. T., Myers, G., and Lai, K. (1990). Register scoreboard on a microprocessor chip. US Patent 4,891,753.
- [Calder and Reinman, 2000] Calder, B. and Reinman, G. (2000). A comparative survey of load speculation architectures. *Journal of Instruction-Level Parallelism*, 2:1–39.
- [Calder et al., 1999] Calder, B., Reinman, G., and Tullsen, D. M. (1999). Selective value prediction. In *Proceedings of the 26th annual international symposium on Computer architecture*, pages 64–74.
- [Callahan et al., 1991] Callahan, D., Kennedy, K., and Porterfield, A. (1991). Software prefetching. In Patterson, D. A. and Rau, B., editors, *ASPLOS-IV Proceedings - Forth International Conference on Architectural Support for Programming Languages and Operating Systems, Santa Clara, California, USA, April 8-11, 1991*, pages 40–52. ACM Press.

- [Champagne and Lee, 2010] Champagne, D. and Lee, R. B. (2010). Scalable architectural support for trusted software. In *HPCA-16 2010 The Sixteenth International Symposium on High-Performance Computer Architecture*, pages 1–12. IEEE.
- [Chandran et al., 2017] Chandran, S., Panda, P. R., Sarangi, S. R., Bhattacharyya, A., Chauhan, D., and Kumar, S. (2017). Managing trace summaries to minimize stalls during postsilicon validation. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 25(6):1881–1894.
- [Chen, 2016] Chen, A. (2016). A review of emerging non-volatile memory (nvm) technologies and applications. *Solid-State Electronics*, 125:25–38.
- [Chen et al., 1997] Chen, I., Bird, P., and Mudge, T. (1997). The impact of instruction compression on i-cache performance. Technical Report CSE-TR-330-97, Computer Science and Engineering, University of Michigan.
- [Chen et al., 2014] Chen, T., Du, Z., Sun, N., Wang, J., Wu, C., Chen, Y., and Temam, O. (2014). Dianao: a small-footprint high-throughput accelerator for ubiquitous machine-learning. In *Proceedings of the 19th international conference on Architectural support for programming languages and operating systems*, pages 269–284.
- [Chen et al., 2012] Chen, Y., Chen, T., Li, L., Li, L., Yang, L., Su, M., and Hu, W. (2012). Ldet: Determinizing asynchronous transfer for postsilicon debugging. *IEEE Transactions on Computers*, 62(9):1732–1744.
- [Chen et al., 2016] Chen, Y.-H., Krishna, T., Emer, J. S., and Sze, V. (2016). Eyeriss: An energy-efficient reconfigurable accelerator for deep convolutional neural networks. *IEEE journal of solid-state circuits*, 52(1):127–138.
- [Choi et al., 2013] Choi, J. W., Bedard, D., Fowler, R., and Vuduc, R. (2013). A roofline model of energy. In *2013 IEEE 27th International Symposium on Parallel and Distributed Processing*, pages 661–672. IEEE.
- [Chrysos and Emer, 1998] Chrysos, G. Z. and Emer, J. S. (1998). Memory dependence prediction using store sets. In *Proceedings of the 25th annual international symposium on Computer architecture*, pages 142–153.
- [Clos, 1953] Clos, C. (1953). A study of non-blocking switching networks. *Bell System Technical Journal*, 32(2):406–424.
- [Coffin Jr, 1954] Coffin Jr, L. F. (1954). A study of the effects of cyclic thermal stresses on a ductile metal. *Transactions of the American Society of Mechanical Engineers, New York*, 76:931–950.
- [Constantinides et al., 2008] Constantinides, K., Mutlu, O., and Austin, T. (2008). Online design bug detection: Rtl analysis, flexible mechanisms, and evaluation. In *2008 41st IEEE/ACM International Symposium on Microarchitecture*, pages 282–293. IEEE.
- [Cooperstein, 2015] Cooperstein, B. (2015). *Advanced linear algebra*. CRC Press.
- [Cormen et al., 2009] Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. (2009). *Introduction to Algorithms*. MIT Press, third edition.
- [Corporation, 2014a] Corporation, N. (2014a). Nvidia geforce gtx 1080. *White paper, NVIDIA Corporation*.
- [Corporation, 2014b] Corporation, N. (2014b). Nvidia’s next generation cuda compute architecture: Kepler GK110/210. *White paper, NVIDIA Corporation*.

- [Coskun et al., 2008] Coskun, A. K., Rosing, T. ., Whisnant, K. A., and Gross, K. C. (2008). Static and dynamic temperature-aware scheduling for multiprocessor socs. *IEEE Trans. VLSI Syst.*, 16(9):1127–1140.
- [Costan and Devadas, 2016] Costan, V. and Devadas, S. (2016). Intel sgx explained. *IACR Cryptology ePrint Archive*, 2016(086):1–118.
- [Cover and Thomas, 2013] Cover, T. M. and Thomas, J. A. (2013). *Elements of Information Theory*. Wiley.
- [Culler et al., 1998] Culler, D., Singh, J. P., and Gupta, A. (1998). *Parallel Computer Architecture: A Hardware/Software Approach*. The Morgan Kaufmann series in Computer Architecture Design. Morgan Kaufmann.
- [Dally and Towles, 2004] Dally, W. J. and Towles, B. P. (2004). *Principles and practices of interconnection networks*. Elsevier.
- [Dan and Towsley, 1990] Dan, A. and Towsley, D. (1990). An approximate analysis of the lru and fifo buffer replacement schemes. In *Proceedings of the 1990 ACM SIGMETRICS conference on Measurement and modeling of computer systems*, pages 143–152.
- [Danilak, 2017] Danilak, R. (2017). Why energy is a big and rapidly growing problem for data centers. <https://www.forbes.com/sites/forbestechcouncil/2017/12/15/why-energy-is-a-big-and-rapidly-growing-problem-for-data-centers>. Accessed on May 15th 2019.
- [David et al., 2013] David, T., Guerraoui, R., and Trigonakis, V. (2013). Everything you always wanted to know about synchronization but were afraid to ask. In *Proceedings of the Twenty-Fourth ACM Symposium on Operating Systems Principles*, pages 33–48. ACM.
- [Diaconis et al., 1983] Diaconis, P., Graham, R., and Kantor, W. M. (1983). The mathematics of perfect shuffles. *Advances in applied mathematics*, 4(2):175–196.
- [Dice et al., 2006] Dice, D., Shalev, O., and Shavit, N. (2006). Transactional locking ii. In *International Symposium on Distributed Computing*, pages 194–208. Springer.
- [Eden and Mudge, 1998] Eden, A. N. and Mudge, T. (1998). The yags branch prediction scheme. In *Proceedings of the 31st Annual ACM/IEEE International Symposium on Microarchitecture*, pages 69–77.
- [Eisenbarth et al., 2007] Eisenbarth, T., Kumar, S., Paar, C., Poschmann, A., and Uhsadel, L. (2007). A survey of lightweight-cryptography implementations. *IEEE Design & Test of Computers*, 24(6):522–533.
- [Elmore, 1948] Elmore, W. C. (1948). The transient response of damped linear networks with particular regard to wideband amplifiers. *Journal of applied physics*, 19(1):55–63.
- [Ergin et al., 2004] Ergin, O., Balkan, D., Ponomarev, D., and Ghose, K. (2004). Increasing processor performance through early register release. In *Computer Design: VLSI in Computers and Processors, 2004. ICCD 2004. Proceedings. IEEE International Conference on*, pages 480–487. IEEE.
- [Ersoy, 1985] Ersoy, O. (1985). Semisystolic array implementation of circular, skew circular, and linear convolutions. *IEEE transactions on computers*, 34(2):190–196.
- [Eyre and Bier, 2000] Eyre, J. and Bier, J. (2000). The evolution of dsp processors. *IEEE Signal Processing Magazine*, 17(2):43–51.

- [Farabet et al., 2011] Farabet, C., Martini, B., Corda, B., Akselrod, P., Culurciello, E., and LeCun, Y. (2011). Neufow: A runtime reconfigurable dataflow processor for vision. In *Cvpr 2011 Workshops*, pages 109–116. IEEE.
- [Farber, 2011] Farber, R. (2011). *CUDA Application Design and Development*. Morgan Kaufmann.
- [Farooqui et al., 2011] Farooqui, N., Kerr, A., Diamos, G., Yalamanchili, S., and Schwan, K. (2011). A framework for dynamically instrumenting gpu compute applications within gpu ocelot. In *Proceedings of the Fourth Workshop on General Purpose Processing on Graphics Processing Units*, pages 1–9.
- [Federovsky et al., 1998] Federovsky, E., Feder, M., and Weiss, S. (1998). Branch prediction based on universal data compression algorithms. In *Proceedings. 25th Annual International Symposium on Computer Architecture*, pages 62–72. IEEE.
- [Feng et al., 2010] Feng, P., Chao, C., Wang, Z.-s., Yang, Y.-c., Jing, Y., and Fei, Z. (2010). Nonvolatile resistive switching memories-characteristics, mechanisms and challenges. *Progress in natural science: Materials international*, 20:1–15.
- [Ferdman et al., 2011] Ferdman, M., Kaynak, C., and Falsafi, B. (2011). Proactive instruction fetch. In *Proceedings of the 44th Annual IEEE/ACM International Symposium on Microarchitecture*, pages 152–162. ACM.
- [Fujitsu Semiconductor Limited, 2010] Fujitsu Semiconductor Limited (2010). Fram guide book. <https://www.fujitsu.com/downloads/MICRO/fme/fram/fram-guide-book.pdf>. Accessed on 20th November, 2019.
- [Gabbay and Mendelson, 1997] Gabbay, F. and Mendelson, A. (1997). Can program profiling support value prediction? In *Proceedings of the 30th annual ACM/IEEE international symposium on Microarchitecture*, pages 270–280. IEEE Computer Society.
- [Gabis and Koudil, 2016] Gabis, A. B. and Koudil, M. (2016). Noc routing protocols-objective-based classification. *Journal of Systems Architecture*, 66:14–32.
- [Gaur et al., 2011] Gaur, J., Chaudhuri, M., and Subramoney, S. (2011). Bypass and insertion algorithms for exclusive last-level caches. In *Proceedings of the 38th annual international symposium on Computer architecture*, pages 81–92.
- [Geer, 2005] Geer, D. (2005). Taking the graphics processor beyond graphics. *Computer*, 38(9):14–16.
- [Gharachorloo, 1995] Gharachorloo, K. (1995). Memory consistency models for shared-memory multiprocessors, phd thesis. *Computer System Laboratory, Stanford Univ.*
- [Glendinning and Helbert, 2012] Glendinning, W. B. and Helbert, J. N. (2012). *Handbook of VLSI microlithography: principles, technology and applications*. William Andrew.
- [Goldreich and Ostrovsky, 1996] Goldreich, O. and Ostrovsky, R. (1996). Software protection and simulation on oblivious rams. *Journal of the ACM (JACM)*, 43(3):431–473.
- [Goodfellow et al., 2016] Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep learning*. MIT press.
- [Gropp et al., 1999] Gropp, W., Thakur, R., and Lusk, E. (1999). *Using MPI-2: Advanced features of the message passing interface*. MIT press.
- [GTX, 2014] GTX, N. G. (2014). 980: Featuring maxwell, the most advanced gpu ever made. *White paper, NVIDIA Corporation*.

- [Guerraoui and Kapalka, 2008] Guerraoui, R. and Kapalka, M. (2008). On the correctness of transactional memory. In *Proceedings of the 13th ACM SIGPLAN Symposium on Principles and practice of parallel programming*, pages 175–184. ACM.
- [Guerraoui and Kapalka, 2010] Guerraoui, R. and Kapalka, M. (2010). Principles of transactional memory. *Synthesis Lectures on Distributed Computing*, 1(1):1–193.
- [Gulli and Pal, 2017] Gulli, A. and Pal, S. (2017). *Deep learning with Keras*. Packt Publishing Ltd.
- [Guo and Solihin, 2006] Guo, F. and Solihin, Y. (2006). An analytical model for cache replacement policy performance. *ACM SIGMETRICS Performance Evaluation Review*, 34(1):228–239.
- [Gutsche et al., 2005] Gutsche, M., Avellan, A., Erben, E., Hecht, T., Hirt, G., Heitmann, J., Igel-Holtzendorff, T., Jakschik, S., Kapteyn, C., Krautheim, G., Kudelka, S., Link, A., Lützen, J., Sängler, A., Schroeder, U., Seidl, H., Stadtmüller, M., and Wiebauer, W. (2005). DRAM Capacitor Scaling. Technical report, Infineon.
- [Gwennap, 2019a] Gwennap, L. (2019a). Snapdragon 865 dis-integrates. Microprocessor Report.
- [Gwennap, 2019b] Gwennap, L. (2019b). Zen 2 boosts ryzen performance. Microprocessor Report.
- [Hachman, 2019] Hachman, M. (2019). Inside the snapdragon 865: Qualcomm reveals the features you’ll find in 2020’s best android phones. <https://www.pcworld.com/article/3482244/inside-the-snapdragon-865-qualcomm-android.html>. Accessed on 10th August, 2020.
- [Halfhill, 2008] Halfhill, T. R. (2008). Intel’s tiny atom. *Microprocessor Report*, 22(4):1.
- [Halfhill, 2019] Halfhill, T. R. (2019). Intel’s tremont: A bigger little core. Microprocessor Report.
- [Harris et al., 2010] Harris, T., Larus, J., and Rajwar, R. (2010). Transactional memory. *Synthesis Lectures on Computer Architecture*, 5(1):1–263.
- [Harris et al., 2006] Harris, T., Plesko, M., Shinnar, A., and Tarditi, D. (2006). Optimizing memory transactions. In *Proceedings of the ACM SIGPLAN 2006 Conference on Programming Language Design and Implementation, Ottawa, Ontario, Canada, June 11-14, 2006*, pages 14–25.
- [Hazucha and Svensson, 2000] Hazucha, P. and Svensson, C. (2000). Impact of cmos technology scaling on the atmospheric neutron soft error rate. *IEEE Transactions on Nuclear science*, 47(6):2586–2594.
- [Helkala et al., 2014] Helkala, J., Viitanen, T., Kultala, H., Jääskeläinen, P., Takala, J., Zetterman, T., and Berg, H. (2014). Variable length instruction compression on transport triggered architectures. In *Embedded Computer Systems: Architectures, Modeling, and Simulation (SAMOS XIV), 2014 International Conference on*, pages 149–155. IEEE.
- [Henning, 2006] Henning, J. L. (2006). Spec cpu2006 benchmark descriptions. *ACM SIGARCH Computer Architecture News*, 34(4):1–17.
- [Herlihy and Shavit, 2012] Herlihy, M. and Shavit, N. (2012). *The Art of Multiprocessor Programming*. Elsevier.
- [Hinton et al., 2001] Hinton, G., Sager, D., Upton, M., Boggs, D., et al. (2001). The microarchitecture of the pentium® 4 processor. In *Intel Technology Journal*.
- [Hong and Kim, 2009] Hong, S. and Kim, H. (2009). An analytical model for a gpu architecture with memory-level and thread-level parallelism awareness. In *Proceedings of the 36th annual international symposium on Computer architecture*, pages 152–163.

- [Horowitz, 1983] Horowitz, M. A. (1983). *Timing models for MOS circuits*. PhD thesis, Stanford University.
- [Howie, 2007] Howie, J. M. (2007). *Fields and Galois theory*. Springer Science & Business Media.
- [Huang et al., 2006] Huang, W., Ghosh, S., Velusamy, S., Sankaranarayanan, K., Skadron, K., and Stan, M. R. (2006). Hotspot: A compact thermal modeling methodology for early-stage vlsi design. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 14(5):501–513.
- [Hung et al., 2006] Hung, W.-L., Link, G. M., Xie, Y., Vijaykrishnan, N., and Irwin, M. J. (2006). Interconnect and thermal-aware floorplanning for 3D microprocessors. In *International Symposium on Quality Electronic Design (ISQED)*. IEEE.
- [Hwu and Patt, 1987] Hwu, W.-M. W. and Patt, Y. N. (1987). Checkpoint repair for high-performance out-of-order execution machines. *Computers, IEEE Transactions on*, 100(12):1496–1514.
- [Intel, 2004] Intel (2004). Enhanced speedstep® technology for the intel® pentium® m processor, white paper, march 2004. <http://download.intel.com/design/network/papers/30117401.pdf>. Accessed on 10th October 2019.
- [Jacob et al., 2007] Jacob, B., Ng, S., and Wang, D. (2007). *Memory Systems: Cache, DRAM, Disk*. Morgan Kaufmann.
- [JEDEC Solid State Technology Association, 2003] JEDEC Solid State Technology Association (2003). Double data rate SDRAM specification. Standard JESD79C, JEDEC.
- [JEDEC Solid State Technology Association, 2008a] JEDEC Solid State Technology Association (2008a). DDR2 SDRAM specification. Standard JESD79-2E, JEDEC.
- [JEDEC Solid State Technology Association, 2008b] JEDEC Solid State Technology Association (2008b). DDR3 SDRAM. Standard JESD79-3C, JEDEC.
- [JEDEC Solid State Technology Association, 2020] JEDEC Solid State Technology Association (2020). DDR4 SDRAM. Standard JESD79-4C, JEDEC.
- [Jerger et al., 2017] Jerger, N. E., Krishna, T., and Peh, L.-S. (2017). On-chip networks. *Synthesis Lectures on Computer Architecture*, 12(3):1–210.
- [Jia et al., 2014] Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., and Darrell, T. (2014). Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 675–678.
- [Jiménez, 2003] Jiménez, D. A. (2003). Fast path-based neural branch prediction. In *Proceedings of the 36th annual IEEE/ACM International Symposium on Microarchitecture*, page 243. IEEE Computer Society.
- [Jiménez, 2011a] Jiménez, D. A. (2011a). Oh-snap: Optimized hybrid scaled neural analog predictor. *Proceedings of the 3rd Championship on Branch Prediction*.
- [Jiménez, 2011b] Jiménez, D. A. (2011b). An optimized scaled neural branch predictor. In *Computer Design (ICCD), 2011 IEEE 29th International Conference on*, pages 113–118. IEEE.
- [Jouppi et al., 2017] Jouppi, N. P., Young, C., Patil, N., Patterson, D., Agrawal, G., Bajwa, R., Bates, S., Bhatia, S., Boden, N., Borchers, A., et al. (2017). In-datacenter performance analysis of a tensor processing unit. In *Proceedings of the 44th Annual International Symposium on Computer Architecture*, pages 1–12.

- [Kaeli et al., 2015] Kaeli, D. R., Mistry, P., Schaa, D., and Zhang, D. P. (2015). *Heterogeneous computing with OpenCL 2.0*. Morgan Kaufmann.
- [Kalayappan and Sarangi, 2013] Kalayappan, R. and Sarangi, S. R. (2013). A survey of checker architectures. *ACM Computing Surveys (CSUR)*, 45(4):1–34.
- [Kallurkar and Sarangi, 2017] Kallurkar, P. and Sarangi, S. R. (2017). Schedtask: a hardware-assisted task scheduler. In *Proceedings of the 50th Annual IEEE/ACM International Symposium on Microarchitecture*, pages 612–624. ACM.
- [Kanter, 2019] Kanter, D. (2019). Intel’s sunny cove sits on an icy lake. Microprocessor Report.
- [Karkar et al., 2016] Karkar, A., Mak, T., Tong, K.-F., and Yakovlev, A. (2016). A survey of emerging interconnects for on-chip efficient multicast and broadcast in many-cores. *IEEE Circuits and Systems Magazine*, 16(1):58–72.
- [Kawahara et al., 2012] Kawahara, T., Ito, K., Takemura, R., and Ohno, H. (2012). Spin-transfer torque ram technology: Review and prospect. *Microelectronics Reliability*, 52(4):613–627.
- [Kaxiras and Martonosi, 2008] Kaxiras, S. and Martonosi, M. (2008). Computer architecture techniques for power-efficiency. *Synthesis Lectures on Computer Architecture*, 3(1):1–207.
- [Keleher et al., 1994] Keleher, P., Cox, A. L., Dwarkadas, S., and Zwaenepoel, W. (1994). Treadmarks: Distributed shared memory on standard workstations and operating systems. In *USENIX Winter*, volume 1994.
- [Keltcher et al., 2003] Keltcher, C. N., McGrath, K. J., Ahmed, A., and Conway, P. (2003). The amd opteron processor for multiprocessor servers. *Micro, IEEE*, 23(2):66–76.
- [Khvalkovskiy et al., 2013] Khvalkovskiy, A., Apalkov, D., Watts, S., Chepulskii, R., Beach, R., Ong, A., Tang, X., Driskill-Smith, A., Butler, W., Visscher, P., et al. (2013). Basic principles of stt-mram cell operation in memory arrays. *Journal of Physics D: Applied Physics*, 46(7):074001.
- [Kim et al., 2003] Kim, C., Burger, D., and Keckler, S. W. (2003). Nonuniform cache architectures for wire-delay dominated on-chip caches. *IEEE Micro*, 23(6):99–107.
- [Kim and Lipasti, 2004] Kim, I. and Lipasti, M. H. (2004). Understanding scheduling replay schemes. In *Proceedings of the 10th International Symposium on High Performance Computer Architecture*.
- [Kim et al., 2007] Kim, J., Dally, W. J., and Abts, D. (2007). Flattened butterfly: a cost-efficient topology for high-radix networks. In *Proceedings of the 34th annual international symposium on Computer architecture*, pages 126–137.
- [Kim et al., 2004] Kim, N. S., Flautner, K., Blaauw, D., and Mudge, T. (2004). Circuit and microarchitectural techniques for reducing cache leakage power. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 12(2):167–184.
- [Klaiber et al., 2000] Klaiber, A. et al. (2000). The technology behind crusoe processors. *Transmeta Technical Brief*.
- [Kocher et al., 2019] Kocher, P., Horn, J., Fogh, A., Genkin, D., Gruss, D., Haas, W., Hamburg, M., Lipp, M., Mangard, S., Prescher, T., et al. (2019). Spectre attacks: Exploiting speculative execution. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 1–19. IEEE.
- [Kolli et al., 2013] Kolli, A., Saidi, A., and Wenisch, T. F. (2013). Rdpip: return-address-stack directed instruction prefetching. In *Microarchitecture (MICRO), 2013 46th Annual IEEE/ACM International Symposium on*, pages 260–271. IEEE.

- [Kong et al., 2012] Kong, J., Chung, S. W., and Skadron, K. (2012). Recent thermal management techniques for microprocessors. *ACM Computing Surveys (CSUR)*, 44(3):1–42.
- [Krick et al., 2000] Krick, R. F., Hinton, G. J., Upton, M. D., Sager, D. J., and Lee, C. W. (2000). Trace based instruction caching. US Patent 6,018,786.
- [Krishna et al., 2008] Krishna, T., Kumar, A., Chiang, P., Erez, M., and Peh, L.-S. (2008). Noc with near-ideal express virtual channels using global-line communication. In *2008 16th IEEE Symposium on High Performance Interconnects*, pages 11–20. IEEE.
- [Kroft, 1981] Kroft, D. (1981). Lockup-free instruction fetch/prefetch cache organization. In *Proceedings of the 8th annual symposium on Computer Architecture*, pages 81–87. IEEE Computer Society Press.
- [Kuhn et al., 2011] Kuhn, K. J., Giles, M. D., Becher, D., Kolar, P., Kornfeld, A., Kotlyar, R., Ma, S. T., Maheshwari, A., and Mudanai, S. (2011). Process technology variation. *IEEE Transactions on Electron Devices*, 58(8):2197–2208.
- [Kung and Picard, 1984] Kung, H. and Picard, R. (1984). One-dimensional systolic arrays for multidimensional convolution and resampling. In *VLSI for Pattern Recognition and Image Processing*, pages 9–24. Springer.
- [Kung and Song, 1981] Kung, H. and Song, S. W. (1981). A systolic 2-d convolution chip. Technical Report CMU-CS-81-110, Carnegie Mellon University, Department of Computer Science.
- [Kung, 1982] Kung, H.-T. (1982). Why systolic architectures? *IEEE computer*, 15(1):37–46.
- [Kwan and Okullo-Oballa, 1990] Kwan, H.-K. and Okullo-Oballa, T. (1990). 2-d systolic arrays for realization of 2-d convolution. *IEEE transactions on circuits and systems*, 37(2):267–233.
- [Kwon et al., 2018] Kwon, H., Chatarasi, P., Pellauer, M., Parashar, A., Sarkar, V., and Krishna, T. (2018). Understanding reuse, performance, and hardware cost of dnn dataflows: A data-centric approach. *arXiv preprint arXiv:1805.02566*.
- [Lam, 1988] Lam, M. (1988). Software pipelining: An effective scheduling technique for vliw machines. In *Proceedings of the ACM SIGPLAN 1988 conference on Programming Language design and Implementation*, pages 318–328.
- [Lam, 2012] Lam, M. S. (2012). *A systolic array optimizing compiler*, volume 64. Springer Science & Business Media.
- [Lavenier et al., 1999] Lavenier, D., Quinton, P., and Rajopadhye, S. (1999). Advanced systolic design. *Digital Signal Processing for Multimedia Systems*, pages 657–692.
- [Lee et al., 2009] Lee, B. C., Ipek, E., Mutlu, O., and Burger, D. (2009). Architecting phase change memory as a scalable dram alternative. In *Proceedings of the 36th Annual International Symposium on Computer Architecture*, ISCA '09, pages 2–13.
- [Lee, 2013] Lee, R. B. (2013). Security basics for computer architects. *Synthesis Lectures on Computer Architecture*, 8(4):1–111.
- [Lefurgy et al., 1997] Lefurgy, C., Bird, P., Chen, I.-C., and Mudge, T. (1997). Improving code density using compression techniques. In *Microarchitecture, 1997. Proceedings., Thirtieth Annual IEEE/ACM International Symposium on*, pages 194–203. IEEE.
- [Leibholz and Razdan, 1997] Leibholz, D. and Razdan, R. (1997). The alpha 21264: A 500 mhz out-of-order execution microprocessor. In *Compcon'97. Proceedings, IEEE*, pages 28–36. IEEE.

- [Leighton, 2014] Leighton, F. T. (2014). *Introduction to parallel algorithms and architectures: Arrays-trees-hypercubes*. Elsevier.
- [Leng et al., 2013] Leng, J., Hetherington, T., ElTantawy, A., Gilani, S., Kim, N. S., Aamodt, T. M., and Reddi, V. J. (2013). Gpuwattch: enabling energy optimizations in gpgpus. In *Proceedings of the 40th Annual International Symposium on Computer Architecture*, pages 487–498.
- [Leng et al., 2015] Leng, J., Zu, Y., and Reddi, V. J. (2015). Gpu voltage noise: Characterization and hierarchical smoothing of spatial and temporal voltage noise interference in gpu architectures. In *2015 IEEE 21st International Symposium on High Performance Computer Architecture (HPCA)*, pages 161–173. IEEE.
- [Leng et al., 2014] Leng, J., Zu, Y., Rhu, M., Gupta, M., and Reddi, V. J. (2014). Gpuvoltage: Modeling and characterizing voltage noise in gpu architectures. In *Proceedings of the 2014 international symposium on Low power electronics and design*, pages 141–146.
- [Lenoski et al., 1990] Lenoski, D., Laudon, J., Gharachorloo, K., Gupta, A., and Hennessy, J. (1990). The directory-based cache coherence protocol for the dash multiprocessor. In *[1990] Proceedings. The 17th Annual International Symposium on Computer Architecture*, pages 148–159. IEEE.
- [Li et al., 2009] Li, S., Ahn, J. H., Strong, R. D., Brockman, J. B., Tullsen, D. M., and Jouppi, N. P. (2009). Mcpat: an integrated power, area, and timing modeling framework for multicore and manycore architectures. In *Proceedings of the 42nd Annual IEEE/ACM International Symposium on Microarchitecture*, pages 469–480. ACM.
- [Li, 2012] Li, X. (2012). *Survey of Wireless Network-on-Chip Systems*. PhD thesis, Auburn University.
- [Lin, 2011] Lin, M.-B. (2011). *Introduction to VLSI Systems: A Logic, Circuit, and System Perspective*. CRC Press.
- [Lindholm et al., 2008] Lindholm, E., Nickolls, J., Oberman, S., and Montrym, J. (2008). Nvidia tesla: A unified graphics and computing architecture. *Micro, IEEE*, 28(2):39–55.
- [Lipasti et al., 1996] Lipasti, M. H., Wilkerson, C. B., and Shen, J. P. (1996). Value locality and load value prediction. In *Proceedings of the seventh international conference on Architectural support for programming languages and operating systems*, pages 138–147.
- [Lipp et al., 2018] Lipp, M., Schwarz, M., Gruss, D., Prescher, T., Haas, W., Fogh, A., Horn, J., Mangard, S., Kocher, P., Genkin, D., et al. (2018). Meltdown: Reading kernel memory from user space. In *27th {USENIX} Security Symposium ({USENIX} Security 18)*, pages 973–990.
- [Logan, 1986] Logan, D. L. (1986). *A First Course in the Finite Element Method*. PWS Engineering.
- [Lu et al., 2017] Lu, W., Yan, G., Li, J., Gong, S., Han, Y., and Li, X. (2017). Flexflow: A flexible dataflow accelerator architecture for convolutional neural networks. In *2017 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, pages 553–564. IEEE.
- [Luk et al., 2005] Luk, C., Cohn, R. S., Muth, R., Patil, H., Klauser, A., Lowney, P. G., Wallace, S., Reddi, V. J., and Hazelwood, K. M. (2005). Pin: building customized program analysis tools with dynamic instrumentation. In *Proceedings of the ACM SIGPLAN Conference on Programming Language Design and Implementation*, pages 190–200.
- [Lustig et al., 2014] Lustig, D., Pellauer, M., and Martonosi, M. (2014). Pipecheck: Specifying and verifying microarchitectural enforcement of memory consistency models. In *Proceedings of the 47th Annual IEEE/ACM International Symposium on Microarchitecture*, pages 635–646. IEEE Computer Society.

- [Ma et al., 2015] Ma, S., Pal, D., Jiang, R., Ray, S., and Vasudevan, S. (2015). Can’t see the forest for the trees: State restoration’s limitations in post-silicon trace signal selection. In *2015 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pages 1–8. IEEE.
- [Mador-Haim et al., 2011] Mador-Haim, S., Alur, R., and Martin, M. M. (2011). Litmus tests for comparing memory consistency models: How long do they need to be? In *Proceedings of the 48th Design Automation Conference*, pages 504–509. ACM.
- [Mahapatra and Parihar, 2018a] Mahapatra, S. and Parihar, N. (2018a). A review of nbti mechanisms and models. *Microelectronics Reliability*, 81:127–135.
- [Mahapatra and Parihar, 2018b] Mahapatra, S. and Parihar, N. (2018b). A review of nbti mechanisms and models. *Microelectronics Reliability*, 81:127–135.
- [Malhotra et al., 2014] Malhotra, G., Goel, S., and Sarangi, S. R. (2014). Gputejas: A parallel simulator for gpu architectures. In *High Performance Computing (HiPC), 2014 21st International Conference on*, pages 1–10. IEEE.
- [Malhotra et al., 2017] Malhotra, G., Kalayappan, R., Goel, S., Aggarwal, P., Sagar, A., and Sarangi, S. R. (2017). Partejas: A parallel simulator for multicore processors. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 27(3):1–24.
- [Manson, 1953] Manson, S. S. (1953). *Behavior of materials under conditions of thermal stress*, volume 2933. National Advisory Committee for Aeronautics.
- [Martínez et al., 2002] Martínez, J. F., Renau, J., Huang, M. C., and Prvulovic, M. (2002). Cherry: Checkpointed early resource recycling in out-of-order microprocessors. In *Microarchitecture, 2002.(MICRO-35). Proceedings. 35th Annual IEEE/ACM International Symposium on*, pages 3–14. IEEE.
- [McNairy and Soltis, 2003] McNairy, C. and Soltis, D. (2003). Itanium 2 processor microarchitecture. *IEEE Micro*, 23(2):44–55.
- [Mittal, 2016a] Mittal, S. (2016a). A survey of architectural techniques for managing process variation. *ACM Computing Surveys (CSUR)*, 48(4):1–29.
- [Mittal, 2016b] Mittal, S. (2016b). A survey of recent prefetching techniques for processor caches. *ACM Computing Surveys (CSUR)*, 49(2):35.
- [Mittal, 2018] Mittal, S. (2018). A survey of techniques for dynamic branch prediction. *CoRR*, abs/1804.00261.
- [Miyaji, 1991] Miyaji, F. (1991). Static random access memory device having a high speed read-out and flash-clear functions. US Patent 5,054,000.
- [Moolchandani et al., 2020] Moolchandani, D., Kumar, A., and Sarangi, S. R. (2020). Accelerating cnn inference on asics: A survey. *Journal of Systems Architecture*, page 101887.
- [Moore et al., 2006] Moore, K. E., Bobba, J., Moravan, M. J., Hill, M. D., and Wood, D. A. (2006). Logtm: Log-based transactional memory. In *The Twelfth International Symposium on High-Performance Computer Architecture, 2006.*, pages 254–265. IEEE.
- [Moscibroda and Mutlu, 2009] Moscibroda, T. and Mutlu, O. (2009). A case for bufferless routing in on-chip networks. In *Proceedings of the 36th annual international symposium on Computer architecture*, pages 196–207.

- [Moshovos et al., 1997] Moshovos, A., Breach, S. E., Vijaykumar, T. N., and Sohi, G. S. (1997). Dynamic speculation and synchronization of data dependences. In *Proceedings of the 24th annual international symposium on Computer architecture*, pages 181–193.
- [Moshovos and Sohi, 1999] Moshovos, A. and Sohi, G. S. (1999). Speculative memory cloaking and bypassing. *International Journal of Parallel Programming*, 27(6):427–456.
- [Muchnick et al., 1997] Muchnick, S. S. et al. (1997). *Advanced compiler design implementation*. Morgan Kaufmann.
- [Mukherjee, 2011] Mukherjee, S. (2011). *Architecture design for soft errors*. Morgan Kaufmann.
- [Muralimanohar et al., 2009] Muralimanohar, N., Balasubramonian, R., and Jouppi, N. P. (2009). Cacti 6.0: A tool to understand large caches. Technical Report HPL-2009-85, University of Utah and Hewlett Packard Laboratories.
- [Mutlu et al., 2003] Mutlu, O., Stark, J., Wilkerson, C., and Patt, Y. N. (2003). Runahead execution: An alternative to very large instruction windows for out-of-order processors. In *High-Performance Computer Architecture, 2003. HPCA-9 2003. Proceedings. The Ninth International Symposium on*, pages 129–140.
- [Narayan and Tran, 1999] Narayan, R. and Tran, T. M. (1999). Method and apparatus for five bit predecoding variable length instructions for scanning of a number of risc operations. US Patent 5,898,851.
- [Neishaburi and Zilic, 2011] Neishaburi, M. H. and Zilic, Z. (2011). Hierarchical embedded logic analyzer for accurate root-cause analysis. In *2011 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems*, pages 120–128. IEEE.
- [Ngabonziza et al., 2016] Ngabonziza, B., Martin, D., Bailey, A., Cho, H., and Martin, S. (2016). Trustzone explained: Architectural features and use cases. In *2016 IEEE 2nd International Conference on Collaboration and Internet Computing (CIC)*, pages 445–451. IEEE.
- [Nose and Sakurai, 2000] Nose, K. and Sakurai, T. (2000). Analysis and future trend of short-circuit power. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 19(9):1023–1030.
- [NVIDIA, 2018] NVIDIA (2018). Cuda toolkit documentation. <https://docs.nvidia.com/cuda/cuda-c-programming-guide/index.html>.
- [NVIDIA Inc., 2017] NVIDIA Inc. (2017). V100 gpu architecture. the world’s most advanced data center gpu. White Paper: Version WP-08608-001_v1.1, NVIDIA.
- [NVIDIA Inc., 2020] NVIDIA Inc. (2020). Cuda compiler driver nvcc. Reference Guide TRM-06721-001_v11.0, NVIDIA.
- [Ors et al., 2004] Ors, S. B., Gurkaynak, F., Oswald, E., and Preneel, B. (2004). Power-analysis attack on an asic aes implementation. In *International Conference on Information Technology: Coding and Computing, 2004. Proceedings. ITCC 2004.*, volume 2, pages 546–552. IEEE.
- [Padhye et al., 2018] Padhye, S., Sahu, R. A., and Saraswat, V. (2018). *Introduction to Cryptography*. CRC Press.
- [Palacharla et al., 1997] Palacharla, S., Jouppi, N. P., and Smith, J. E. (1997). Complexity-effective superscalar processors. In *Proceedings of the 24th annual international symposium on Computer architecture*, pages 206–218.

- [Parashar et al., 2019] Parashar, A., Raina, P., Shao, Y. S., Chen, Y.-H., Ying, V. A., Mukkara, A., Venkatesan, R., Khailany, B., Keckler, S. W., and Emer, J. (2019). Timeloop: A systematic approach to dnn accelerator evaluation. In *2019 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*, pages 304–315. IEEE.
- [Park et al., 2003] Park, I., Ooi, C. L., and Vijaykumar, T. (2003). Reducing design complexity of the load/store queue. In *Proceedings of the 36th annual IEEE/ACM International Symposium on Microarchitecture*, page 411. IEEE Computer Society.
- [Park et al., 2010] Park, J.-H., Shin, S., Christofferson, J., Shakouri, A., and Kang, S.-M. (2010). Experimental validation of the power blurring method. In *SEMI-THERM*, pages 240–244. IEEE.
- [Peterson et al., 1991] Peterson, C., Sutton, J., and Wiley, P. (1991). iWarp: a 100-MOPS, LIW microprocessor for multicomputers. *Micro, IEEE*, 11(3):26–29.
- [Petric et al., 2005] Petric, V., Sha, T., and Roth, A. (2005). Reno: a rename-based instruction optimizer. In *32nd International Symposium on Computer Architecture (ISCA'05)*, pages 98–109. IEEE.
- [Pinto and Santos, 2019] Pinto, S. and Santos, N. (2019). Demystifying arm trustzone: A comprehensive survey. *ACM Computing Surveys (CSUR)*, 51(6):1–36.
- [Powell et al., 2001] Powell, M. D., Agarwal, A., Vijaykumar, T., Falsafi, B., and Roy, K. (2001). Reducing set-associative cache energy via way-prediction and selective direct-mapping. In *Proceedings of the 34th annual ACM/IEEE international symposium on Microarchitecture*, pages 54–65. IEEE Computer Society.
- [Powell and Vijaykumar, 2003a] Powell, M. D. and Vijaykumar, T. (2003a). Pipeline damping: a microarchitectural technique to reduce inductive noise in supply voltage. In *30th Annual International Symposium on Computer Architecture, 2003. Proceedings.*, pages 72–83. IEEE.
- [Powell and Vijaykumar, 2003b] Powell, M. D. and Vijaykumar, T. (2003b). Pipeline muffling and a priori current ramping: architectural techniques to reduce high-frequency inductive noise. In *Proceedings of the 2003 international symposium on Low power electronics and design*, pages 223–228.
- [Pratt, 1995] Pratt, V. (1995). Anatomy of the pentium bug. In *TAPSOFT'95: Theory and Practice of Software Development*, pages 97–107. Springer.
- [Prvulovic, 2006] Prvulovic, M. (2006). Cord: Cost-effective (and nearly overhead-free) order-recording and data race detection. In *The Twelfth International Symposium on High-Performance Computer Architecture, 2006.*, pages 232–243. IEEE.
- [Quinn, 2017] Quinn, M. (2017). *Parallel Programming in C with MPI and OpenMP*. McGrawHill Education.
- [Qureshi et al., 2011] Qureshi, M. K., Gurumurthi, S., and Rajendran, B. (2011). Phase change memory: From devices to systems. *Synthesis Lectures on Computer Architecture*, 6(4):1–134.
- [Rashkeev et al., 2002] Rashkeev, S., Fleetwood, D., Schrimpf, R., and Pantelides, S. (2002). Dual behavior of H^+ at $Si-SiO_2$ interfaces: Mobility versus trapping. *Applied physics letters*, 81(10):1839–1841.
- [Rastegar, 1994] Rastegar, B. (1994). Integrated circuit memory device having flash clear. US Patent 5,311,477.
- [Rathnam and Slavenburg, 1996] Rathnam, S. and Slavenburg, G. (1996). An architectural overview of the programmable multimedia processor, tm-1. In *Compton'96. Technologies for the Information Superhighway Digest of Papers*, pages 319–326. IEEE.

- [Rau, 1993] Rau, B. R. (1993). Dynamically scheduled vliw processors. In *Proceedings of the 26th annual international symposium on Microarchitecture*, pages 80–92. IEEE Computer Society Press.
- [Rau, 1994] Rau, B. R. (1994). Iterative modulo scheduling: An algorithm for software pipelining loops. In *Proceedings of the 27th annual international symposium on Microarchitecture*, pages 63–74. ACM.
- [Reagen et al., 2017] Reagen, B., Adolf, R., Whatmough, P., Wei, G.-Y., and Brooks, D. (2017). Deep learning for computer architects. *Synthesis Lectures on Computer Architecture*, 12(4):1–123.
- [Reinman and Jouppi, 2000] Reinman, G. and Jouppi, N. P. (2000). Cacti 2.0: An integrated cache timing and power model. Research Report 2000/7, Compaq Western Research Laboratory.
- [Ren et al., 2017] Ren, L., Fletcher, C. W., Kwon, A., Van Dijk, M., and Devadas, S. (2017). Design and implementation of the ascend secure processor. *IEEE Transactions on Dependable and Secure Computing*, 16(2):204–216.
- [Rogers et al., 2007] Rogers, B., Chhabra, S., Prvulovic, M., and Solihin, Y. (2007). Using address independent seed encryption and bonsai merkle trees to make secure processors OS and performance-friendly. In *40th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO 2007)*, pages 183–196. IEEE.
- [Roy et al., 2003] Roy, K., Mukhopadhyay, S., and Mahmoodi-Meimand, H. (2003). Leakage current mechanisms and leakage reduction techniques in deep-submicrometer cmos circuits. *Proceedings of the IEEE*, 91(2):305–327.
- [Rumpf and Strzodka, 2006] Rumpf, M. and Strzodka, R. (2006). Graphics processor units: New prospects for parallel computing. In *Numerical solution of partial differential equations on parallel computers*, pages 89–132. Springer.
- [Rupp, 2017] Rupp, K. (2017). Moore’s law: Transistors per microprocessor. <https://ourworldindata.org/grapher/transistors-per-microprocessor>. Accessed on 11th August 2020.
- [Saini, 1993] Saini, A. (1993). Design of the intel pentium processor. In *Computer Design: VLSI in Computers and Processors, 1993. ICCD’93. Proceedings., 1993 IEEE International Conference on*, pages 258–261. IEEE.
- [Salminen et al., 2008] Salminen, E., Kulmala, A., and Hamalainen, T. D. (2008). Survey of network-on-chip proposals. *White paper, OCP-IP*, 1:13.
- [Samajdar et al., 2020] Samajdar, A., Joseph, J. M., Zhu, Y., Whatmough, P., Mattina, M., and Krishna, T. (2020). A systematic methodology for characterizingscalability of dnn accelerators using scale-sim. In *International Symposium on Performance Analysis of Systems and Software*. IEEE.
- [Sarangi, 2015] Sarangi, S. R. (2015). *Computer Organisation and Architecture*. McGrawHill.
- [Sarangi et al., 2014] Sarangi, S. R., Ananthanarayanan, G., and Balakrishnan, M. (2014). Lightsim: A leakage aware ultrafast temperature simulator. In *2014 19th Asia and South Pacific Design Automation Conference (ASP-DAC)*, pages 855–860. IEEE.
- [Sarangi et al., 2008] Sarangi, S. R., Greskamp, B., Teodorescu, R., Nakano, J., Tiwari, A., and Torrellas, J. (2008). Varius: A model of process variation and resulting timing errors for microarchitects. *IEEE Transactions on Semiconductor Manufacturing*, 21(1):3–13.
- [Sarangi et al., 2006a] Sarangi, S. R., Greskamp, B., and Torrellas, J. (2006a). Cadre: Cycle-accurate deterministic replay for hardware debugging. In *International Conference on Dependable Systems and Networks (DSN’06)*, pages 301–312. IEEE.

- [Sarangi et al., 2015] Sarangi, S. R., Kalayappan, R., Kallurkar, P., Goel, S., and Peter, E. (2015). Tejas: A java based versatile micro-architectural simulator. In *International Workshop on Power and Timing Modeling, Optimization and Simulation (PATMOS)*.
- [Sarangi et al., 2006b] Sarangi, S. R., Tiwari, A., and Torrellas, J. (2006b). Phoenix: Detecting and recovering from permanent processor design bugs with programmable hardware. In *Proceedings of the 39th Annual IEEE/ACM International Symposium on Microarchitecture*, pages 26–37. IEEE Computer Society.
- [Savage et al., 1997] Savage, S., Burrows, M., Nelson, G., Sobalvarro, P., and Anderson, T. (1997). Eraser: A dynamic data race detector for multithreaded programs. *ACM Transactions on Computer Systems (TOCS)*, 15(4):391–411.
- [Scheurich and Dubois, 1988] Scheurich, C. and Dubois, M. (1988). The design of a lockup-free cache for high-performance multiprocessors. In *Proceedings of the 1988 ACM/IEEE Conference on Supercomputing*, Supercomputing '88, pages 352–359.
- [Sehatbakhsh et al., 2020] Sehatbakhsh, N., Nazari, A., Alam, M., Werner, F., Zhu, Y., Zajic, A. G., and Prvulovic, M. (2020). REMOTE: robust external malware detection framework by using electromagnetic signals. *IEEE Trans. Computers*, 69(3):312–326.
- [Settle et al., 2003] Settle, A., Connors, D. A., Hoflehner, G., and Lavery, D. (2003). Optimization for the intel/spl reg/itanium/spl reg/architecture register stack. In *Code Generation and Optimization, 2003. CGO 2003. International Symposium on*, pages 115–124. IEEE.
- [Seznec, 1993] Seznec, A. (1993). A case for two-way skewed-associative caches. In *Proceedings of the 20th Annual International Symposium on Computer Architecture*, pages 169–178. IEEE.
- [Seznec, 2004] Seznec, A. (2004). Revisiting the perceptron predictor. Technical Report PI-1620, IRISA, France.
- [Seznec, 2007] Seznec, A. (2007). A 256 kbits l-tage branch predictor. *Journal of Instruction-Level Parallelism (JILP) Special Issue: The Second Championship Branch Prediction Competition (CBP-2)*, 9:1–6.
- [Seznec et al., 2002] Seznec, A., Felix, S., Krishnan, V., and Sazeides, Y. (2002). Design tradeoffs for the alpha ev8 conditional branch predictor. In *Proceedings 29th Annual International Symposium on Computer Architecture*, pages 295–306. IEEE.
- [Sharangpani and Arora, 2000] Sharangpani, H. and Arora, H. (2000). Itanium processor microarchitecture. *IEEE Micro*, 20(5):24–43.
- [Shivakumar and Jouppi, 2001] Shivakumar, P. and Jouppi, N. P. (2001). Cacti 3.0: An integrated cache timing, power, and area model. Research Report 2001/2, Compaq Western Research Laboratory.
- [Silberschatz et al., 2018] Silberschatz, A., Gagne, G., and Galvin, P. B. (2018). *Operating system concepts*. Wiley.
- [Själänder et al., 2014] Själänder, M., Martonosi, M., and Kaxiras, S. (2014). Power-efficient computer architectures: Recent advances. *Synthesis Lectures on Computer Architecture*, 9(3):1–96.
- [Slegel et al., 1999] Slegel, T. J., Averill, R. M., Check, M. A., Giamei, B. C., Krumm, B. W., Krygowski, C. A., Li, W. H., Liptay, J. S., MacDougall, J. D., McPherson, T. J., et al. (1999). Ibm's s/390 g5 microprocessor design. *IEEE micro*, 19(2):12–23.
- [Sloss et al., 2004] Sloss, A., Symes, D., and Wright, C. (2004). *ARM system developer's guide: designing and optimizing system software*. Elsevier.

- [Smith and Sohi, 1995] Smith, J. E. and Sohi, G. S. (1995). The microarchitecture of superscalar processors. *Proceedings of the IEEE*, 83(12):1609–1624.
- [Sorin et al., 2011] Sorin, D. J., Hill, M. D., and Wood, D. A. (2011). A primer on memory consistency and cache coherence. *Synthesis Lectures on Computer Architecture*, 6(3):1–212.
- [Sprangle et al., 1997] Sprangle, E., Chappell, R. S., Alsup, M., and Patt, Y. N. (1997). The agree predictor: A mechanism for reducing negative branch history interference. In *Proceedings of the 24th annual international symposium on Computer architecture*, pages 284–291.
- [Sridhar et al., 2010] Sridhar, A., Vincenzi, A., Ruggiero, M., Brunschweiler, T., and Atienza, D. (2010). 3d-ice: Fast compact transient thermal modeling for 3d ics with inter-tier liquid cooling. In *2010 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pages 463–470. IEEE.
- [Srinivasan et al., 2005] Srinivasan, J., Adve, S., Bose, P., and Rivers, J. (2005). Exploiting structural duplication for lifetime reliability enhancement. In *32nd International Symposium on Computer Architecture (ISCA'05)*, pages 520–531. IEEE.
- [Srinivasan et al., 2004] Srinivasan, J., Adve, S. V., Bose, P., and Rivers, J. A. (2004). The case for lifetime reliability-aware microprocessors. In *Proceedings of the 31st annual international symposium on Computer architecture*, ISCA '04, pages 276–.
- [Stallings, 2006] Stallings, W. (2006). *Cryptography and network security, 4/E*. Pearson Education India.
- [Stefanov et al., 2013] Stefanov, E., Van Dijk, M., Shi, E., Fletcher, C., Ren, L., Yu, X., and Devadas, S. (2013). Path oram: an extremely simple oblivious ram protocol. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, pages 299–310.
- [Stenstrom, 1990] Stenstrom, P. (1990). A survey of cache coherence schemes for multiprocessors. *Computer*, 23(6):12–24.
- [Suggs and Bouvier, 2019] Suggs, D. and Bouvier, D. (2019). Zen 2. <https://www.youtube.com/watch?v=QU3PHKdj8wQ>. Accessed on 17th August, 2020.
- [Suh et al., 2005] Suh, G. E., O'Donnell, C. W., and Devadas, S. (2005). Aegis: A single-chip secure processor. *Information Security Technical Report*, 10(2):63–73.
- [Sultan et al., 2014] Sultan, H., Ananthanarayanan, G., and Sarangi, S. R. (2014). Processor power estimation techniques: a survey. *IJHPSA*, 5(2):93–114.
- [Sultan et al., 2019] Sultan, H., Chauhan, A., and Sarangi, S. R. (2019). A survey of chip-level thermal simulators. *ACM Comput. Surv.*, 52(2):42:1–42:35.
- [Sultan and Sarangi, 2017] Sultan, H. and Sarangi, S. R. (2017). A fast leakage aware thermal simulator for 3d chips. In *Design, Automation & Test in Europe Conference & Exhibition, DATE 2017, Lausanne, Switzerland, March 27-31, 2017*, pages 1733–1738.
- [Sultan et al., 2018] Sultan, H., Varshney, S., and Sarangi, S. R. (2018). Is leakage power a linear function of temperature? *arXiv preprint arXiv:1809.03147*.
- [Sze et al., 2020] Sze, V., Chen, Y.-H., Yang, T.-J., and Emer, J. S. (2020). Efficient processing of deep neural networks. *Synthesis Lectures on Computer Architecture*, 15(2):1–341.
- [Szefer, 2018] Szefer, J. (2018). Principles of secure processor architecture design. *Synthesis Lectures on Computer Architecture*, 13(3):1–173.

- [Szefer, 2019] Szefer, J. (2019). Survey of microarchitectural side and covert channels, attacks, and defenses. *Journal of Hardware and Systems Security*, 3(3):219–234.
- [Taassori et al., 2018] Taassori, M., Shafiee, A., and Balasubramonian, R. (2018). Vault: Reducing paging overheads in sgx with efficient integrity verification structures. In *Proceedings of the Twenty-Third International Conference on Architectural Support for Programming Languages and Operating Systems*, pages 665–678.
- [Tarjan et al., 2006] Tarjan, D., Thoziyoor, S., and Jouppi, N. P. (2006). Cacti 4.0. Technical Report HPL-2006-86, HP Laboratories.
- [Taub and Schilling, 1977] Taub, H. and Schilling, D. L. (1977). *Digital integrated electronics*. McGraw-Hill New York.
- [Thekkath et al., 2000] Thekkath, D. L. C., Mitchell, M., Lincoln, P., Boneh, D., Mitchell, J., and Horowitz, M. (2000). Architectural support for copy and tamper resistant software. In *Proceedings of the Ninth International Conference on Architectural Support for Programming Languages and Operating Systems*, page 168–177.
- [Thornton, 2000] Thornton, J. E. (2000). Parallel operation in the control data 6600. *Readings in computer architecture*, page 32.
- [Thoziyoor et al., 2007] Thoziyoor, S., Muralimanohar, N., and Jouppi, N. P. (2007). Cacti 5.0. Technical Report HPL-2007-167, HP Laboratories.
- [Tiwari and Torrellas, 2008] Tiwari, A. and Torrellas, J. (2008). Facelift: Hiding and slowing down aging in multicores. In *2008 41st IEEE/ACM International Symposium on Microarchitecture*, pages 129–140. IEEE.
- [Turkington, 2013] Turkington, D. A. (2013). *Generalized vectorization, cross-products, and matrix calculus*. Cambridge University Press.
- [Van Bulck et al., 2018] Van Bulck, J., Minkin, M., Weisse, O., Genkin, D., Kasikci, B., Piessens, F., Silberstein, M., Wenisch, T. F., Yarom, Y., and Strackx, R. (2018). Foreshadow: Extracting the keys to the intel {SGX} kingdom with transient out-of-order execution. In *27th {USENIX} Security Symposium ({USENIX} Security 18)*, pages 991–1008.
- [Vangal et al., 2007] Vangal, S., Howard, J., Ruhl, G., Dighe, S., Wilson, H., Tschanz, J., Finan, D., Iyer, P., Singh, A., Jacob, T., et al. (2007). An 80-tile 1.28 tflops network-on-chip in 65nm cmos. In *2007 IEEE International Solid-State Circuits Conference. Digest of Technical Papers*, pages 98–589. IEEE.
- [Vantrease et al., 2011] Vantrease, D., Lipasti, M. H., and Binkert, N. (2011). Atomic coherence: Leveraging nanophotonics to build race-free cache coherence protocols. In *High Performance Computer Architecture (HPCA), 2011 IEEE 17th International Symposium on*, pages 132–143.
- [Wang et al., 2018] Wang, C., Wu, H., Gao, B., Zhang, T., Yang, Y., and Qian, H. (2018). Conduction mechanisms, dynamics and stability in rerams. *Microelectronic Engineering*, 187:121–133.
- [Wang et al., 2005] Wang, D., Ganesh, B., Tuaycharoen, N., Baynes, K., Jaleel, A., and Jacob, B. (2005). Dramsim: a memory system simulator. *ACM SIGARCH Computer Architecture News*, 33(4):100–107.
- [Wang and Agrawal, 2008] Wang, F. and Agrawal, V. D. (2008). Single event upset: An embedded tutorial. In *21st International Conference on VLSI Design (VLSID 2008)*, pages 429–434. IEEE.

- [Wang et al., 2013] Wang, J., Tim, Y., Wong, W.-F., and Li, H. H. (2013). A practical low-power memristor-based analog neural branch predictor. In *Low Power Electronics and Design (ISLPED), 2013 IEEE International Symposium on*, pages 175–180. IEEE.
- [Wang and Franklin, 1997] Wang, K. and Franklin, M. (1997). Highly accurate data value prediction using hybrid predictors. In *Proceedings of the 30th annual ACM/IEEE international symposium on Microarchitecture*, pages 281–290. IEEE Computer Society.
- [Wang et al., 2016] Wang, Y., Li, H., and Li, X. (2016). Re-architecting the on-chip memory sub-system of machine-learning accelerator for embedded devices. In *2016 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pages 1–6. IEEE.
- [Wegman and Carter, 1981] Wegman, M. N. and Carter, J. L. (1981). New hash functions and their use in authentication and set equality. *Journal of computer and system sciences*, 22(3):265–279.
- [Wickerson et al., 2017] Wickerson, J., Batty, M., Sorensen, T., and Constantinides, G. A. (2017). Automatically comparing memory consistency models. In *Proceedings of the 44th ACM SIGPLAN Symposium on Principles of Programming Languages*, pages 190–204.
- [Williams et al., 2009] Williams, S., Waterman, A., and Patterson, D. (2009). Roofline: an insightful visual performance model for multicore architectures. *Communications of the ACM*, 52(4):65–76.
- [Williamson, 2007] Williamson, D. (2007). Arm cortex-a8: A high-performance processor for low-power applications. *Unique Chips and Systems*, page 79.
- [Wilton and Jouppi, 1993] Wilton, S. J. and Jouppi, N. P. (1993). An enhanced access and cycle time model for on-chip caches. Research Report 93/5, Digital Western Research Laboratory.
- [Wittenbrink et al., 2011] Wittenbrink, C. M., Kilgariff, E., and Prabhu, A. (2011). Fermi gf100 gpu architecture. *IEEE Micro*, 31(2):50–59.
- [Wong et al., 2010] Wong, H., Papadopoulou, M.-M., Sadooghi-Alvandi, M., and Moshovos, A. (2010). Demystifying gpu microarchitecture through microbenchmarking. In *Performance Analysis of Systems & Software (ISPASS), 2010 IEEE International Symposium on*, pages 235–246. IEEE.
- [Woo et al., 1995] Woo, S. C., Ohara, M., Torrie, E., Singh, J. P., and Gupta, A. (1995). The splash-2 programs: Characterization and methodological considerations. *ACM SIGARCH computer architecture news*, 23(2):24–36.
- [Wouters, 2009] Wouters, D. (2009). Oxide resistive ram (oxrram) for scaled nvm application. *Innovative Mass Storage Technologies-IMST*.
- [Wu et al., 2019] Wu, Y. N., Emer, J. S., and Sze, V. (2019). Accelergy: An architecture-level energy estimation methodology for accelerator designs. In *2019 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pages 1–8. IEEE.
- [Xu and Liu, 2010] Xu, Q. and Liu, X. (2010). On signal tracing in post-silicon validation. In *2010 15th Asia and South Pacific Design Automation Conference (ASP-DAC)*, pages 262–267. IEEE.
- [Yeh and Patt, 1991] Yeh, T.-Y. and Patt, Y. N. (1991). Two-level adaptive training branch prediction. In *Proceedings of the 24th annual international symposium on Microarchitecture*, pages 51–61. ACM.
- [Yeh and Patt, 1992] Yeh, T.-Y. and Patt, Y. N. (1992). Alternative implementations of two-level adaptive branch prediction. In *Proceedings of the 19th annual international symposium on Computer architecture*, pages 124–134.

- [Yeh and Patt, 1993] Yeh, T.-Y. and Patt, Y. N. (1993). A comparison of dynamic branch predictors that use two levels of branch history. In *Proceedings of the 20th annual international symposium on computer architecture*, pages 257–266.
- [Yiu, 2009] Yiu, J. (2009). *The definitive guide to the ARM Cortex-M3*. Newnes.
- [Yoaz et al., 1999] Yoaz, A., Erez, M., Ronen, R., and Jourdan, S. (1999). Speculation techniques for improving load related instruction scheduling. In *Proceedings of the 26th annual international symposium on Computer architecture*, pages 42–53.
- [Yu, 2016] Yu, S. (2016). Resistive random access memory (rram) from devices to array architectures. *Synthesis Lectures on Computer Architecture*, 6.
- [Yu and Chen, 2016] Yu, S. and Chen, P.-Y. (2016). Emerging memory technologies: Recent trends and prospects. *IEEE Solid-State Circuits Magazine*, 8(2):43–56.
- [Zhou et al., 2007] Zhou, P., Teodorescu, R., and Zhou, Y. (2007). Hard: Hardware-assisted lockset-based race detection. In *2007 IEEE 13th International Symposium on High Performance Computer Architecture*, pages 121–132. IEEE.

Index

- 2-level Predictor, 74
- 2D Convolution, 728
- Access Graph, 456
- Access Transistor, 536
- Accumulator, 88
- ACID Properties, 511
- ACIF Properties, 683
- Acquire Instruction, 445
- Activity Factor, 611
- Adaptive Routing, 372
- Address Space Layout Randomization, 685
- Address Translation Attack, 693
- Adiabatic, 618
- Adiabatic Boundary, 618
- Advanced Memory Buffer (AMB), 559
- AES, 672
 - hardware implementation, 677
 - key schedule, 673
 - operations, 673
- AES Block Cipher, 672
- AES Modes, 674
- Aging, 662
- Allocators, 387
- Alpha Power Law, 612, 658
- AMD Processors, 755
- Amdahl's Law, 421
- Arbiters, 387
- Arbitration, 469
- Architectural Simulation, 743
- Architectural Vulnerability Factor, 643
- ARF, 174
- ARM Memory Model, 494
- ASIC, 215
- ASIP, 215
- Asymmetric Ciphers, 678
- Asynchronous DRAM, 552
- Atomic Exchange, 485
- Atomic Operations, 479
- Atomicity, 433
- Attack Surface, 685
- Attack Vectors, 686
- Authenticity, 679
- Automated Theorem Proving, 652
- Average Memory Access Time, 268
- Axioms of Coherence, 442
- Back-to-back Execution, 120
- Backward Slice, 329
- Band-to-Band Tunneling, 614
- Bandwidth, 353
- Bank, *see* Cache Bank 291
- Bank Set, 402
- Barrier, 228
- Bartok STM, 518
- BEDO, 555
- Benes Network, 345
- Binary Modular Exponentiation, 696
- Binning, 652
- Bit Lane Steering, 562
- Bit-Complement Traffic, 408
- Bit-Reverse Traffic, 409
- Bit-Rotation Traffic, 409
- Bit-serial Multiplier, 731
- Block, 226
- Block Cipher, 672
- blockDim, 228
- Blocking, *see* Loop Tiling 308
- Branch History Register, 73
- Branch Interlock, 36
- Branch Target Buffer, 82
- Broadcast, 112
 - tag, 113
 - tag bus, 113
- BTB, 82
- Buffer Overflow Attack, 684
- Buffered DIMMs, 558
- Buffered Links, 337
- Burst Chop Mode, 558
- Burst Extended Data Out, *see* BEDO 555
- Burst Length, 558
- Busy Wait Loop, 444
- Butterfly Network, 345
- Bypass Network, 124

- Bypassing, 39, 121, 122
- Byzantine Failures, 638
- Cache, 256
 - basic operations, 266
 - replacement, 267
 - write back, 267
 - write through, 267
- Cache Bank, 291
- Cache Coherence, 425, 442
 - axioms, 443
 - race conditions, 477
 - write-invalidate protocol, 464
 - write-update protocol, 459
- Cache Coherence Problem, 426
- Cache Coherence Protocol, 426
- Cache Coherence Protocols, 459
- Cache Hierarchy, 258
- Cache Hit, 257
- Cache Index, 260
- Cache Line, 260
- Cache Miss, 257
- Cacti, 277, 289
 - bank, 292
 - mat, 292
 - subarray, 292
 - subbank, 292
- Call Graph, 321
- Call Graph History Cache, 323
- Call Graph Prefetching, 321
- CAM Array, 287
- CAM Cell, 285
- Capacity Misses, 269
- Carter-Wegman Construction, 690
- Causal Graph, 458
- Causal Ordering, 506
- Causality, 457
- CBRAMs, 592
- CDG, *see* Channel Dependence Graph 366
- Chain of Trust, 686
- Channel, 348, 710
- Channel Dependence Graph, 366
- Charge Sharing, 734
- Checkpoint, 645
- Chemical Mechanical Polishing, 657
- Cipher Block Chaining, 674
- Ciphertext, 671
- Circuit Switching, 352
- Circular Queue, 107
- Clock Cycle, 29
 - clock period, 29
 - frequency, 29
- Clock Gating, 627
- Clock Layer, 340
- Clos Network, 343
- Close-Page Access Policy, 573
- CNNs, 709
 - design, 711, 712
 - memory system, 732
 - optimizations, 730
- Coarse-grained Multithreading, 424
- Code Injection Attack, 684
- Code Reuse Attack, 684
- Coherence, 428, 442
- Cold Boot Attack, 687
- Column Multiplexers, 283
- Commit, 133
- Commit Width, 135
- Common Mode Rejection, 283
- Common Subexpression Elimination, 180, 730
- Compare and Set, 485
- Compiler Techniques, 178
- Compulsory Misses, 269
- Concurrency Control, 513
 - optimistic, 513
 - pessimistic, 513
- Conduction, 619
- Conduction Band, 615
- Conductive Filament, 591
- Confidentiality, 671
- Conflict Detection, 513
 - eager, 513
 - lazy, 513
 - validation, 513
- Conflict Misses, 269
- Conflicting Access, 465
- Conflicting Accesses, 497
- Conflicts, 512
 - detection, 512
 - occurrence, 512
 - resolution, 512
- Confusion, 672
- Consensus Number, 489
- Consensus Problem, 489
- Constant Folding, 179
- Content Addressable Memory, *see* CAM Array 287
- Content Addressable Memory , *see* CAM Array 263
- Context of a Program, 139
- Control Hazard, 35
- Control Path, 27
- Convection, 619
- Convolution Operation, 710

- Convolutional Layer, 709
- Convolutional Neural Networks, *see* CNNs 709
- Convoy Effect, 153
- Core, 13, 415
- Correct Path, 52
- Correctness of Executions, 459
- Counter Mode Encryption, 675
- Counters, 688
- Covert Channel, 696
- Credit, 348
- Credit based Flow Control, 348
- Critical Section, 480
- Critical Word First, 270
- Crossbar Switch, 385
- Cuckoo Hashing, 307
- CUDA, 225
 - block, 226
 - events, 233
 - graphs, 231
 - grid, 226
 - kernel, 226
 - streams, 231
 - thread, 226
- Current Summing, 734
- d-cache , *see* Data Cache 257
- DAG, 49
- Data Cache, 257
- Data Dependence Order, 48
- Data Flow Optimizations, 178
- Data Hazard, 33
- Data Integrity, 680
- Data Interlock, 34
- Data Path, 27
- Data Prefetching, 324
- Data Race, 497
- Data-Race-Freedom, 499
- DDR Memory, 556
- DDR2 Memory, 557
- DDR3 Memory, 557
- DDR4 Memory, 557
- Dead Code, 180
- Dead Code Elimination, 180
- Deadlock, 363, 365
- Deadlock Avoidance, 369
- Deadlock Recovery, 369
- Decode, 27
- Decoupling Capacitors, 629
- Deep Feed-Forward Networks, 709
- Deep Learning, 704
- Deep Neural Networks, *see* DNNs 708
- Delayed Branches, 37
- Delayed Selective Replay, 167
- Dennard Scaling, 14
- Dependence Check Logic, 102, 106
- Dependence Speculation, 155
- Design Faults, 651
- Design for Debug, 655
- Destructive Read, 539
- Detected but Unrecoverable Errors, 643
- DfD, *see* Design for Debug 655
- DIBL, 616
- Diffie-Hellman Key Exchange Protocol, 680
- Diffusion, 672
- Digital Signature, 679
- dim3, 226
- Dimension Slicing, 386
- Dimension-Ordered Routing, 369
- Dipole Moment, 582
- Direct Mapped Cache, 261
- Direct Tunneling, 616
- Directed Acyclic Graph, 49
- Directory, 472
- Directory Protocol, 472
- Disjoint Access Parallelism, 510
- Disjoint Lock Atomicity, 518
- Dispatch, 108
- Dispatcher, 713
- Distance Fog, 224
- Distributed Cache, 425
- Distributed Shared Memory, 417
- DMR, *see* Dual Modular Redundancy 645
- DNNs, 708
- Dopant Density Fluctuations, 657
- Double Data Rate Memory, 556
- Doubly Block Circulant Matrix, 729
- Drain Induced Barrier Lowering, 616
- DRAM Access Protocols, 551
- DRAM Address Mapping Policies, 574
- DRAM Addressing, 574
 - close page, 575
 - open page, 575
- DRAM Array, 538
 - read, 539
 - refresh, 546
 - sense amplifier, 541
 - write, 546
- DRAM Cell, 536
- DRAM Command Scheduling, 576
- DRAM Page, 538
- DRAM Scheduling, 572
 - greedy scheduling, 576

- rank round robin scheduling, 576
- bank round robin scheduling, 576
- commands, 576
- DRAMs, 536
- DRF Memory Models, 503
- Drowsy Caches, 630
- Dual Modular Redundancy, 645
- DVFS, 612
- Dynamic NUCA, 402
- Dynamic Power, 609
- Dynamic Power Management, 625
- Dynamic RAMs , see DRAMs 536
- Dynamic Voltage and Frequency Scaling , see DVFS 612
- Early Restart, 270
- Eavesdropping, 680
- EDA, 46
- EDO, 554
- EEPROM, 577
- Electrical Masking, 642
- Electromigration, 664
- Electronic Codebook, 674
- Elmore Delay Model, 293
- Embedded DRAM, 593
- Emulator, 744
- Enclave, 686, 693
- Energy Delay Square Metric, 612
- EPIC Processors, 194
- EPROM, 577
- Error, 637
- Error Masking, 642
- Event Queue, 745
- Evict+Time, 698
- Execution, 430, 435, 447
- Execution Witness, 447
- Extended Data Out, see EDO 554
- Fail-stop Failure Mode, 638
- Failure, 637
- False Sharing, 478
- Fast Page Mode, see FPM 553
- Fault, 637
- FB-DIMMs, 559
- Feature Size, 14
- Fence Instruction, 444
- FeRAM, 582
- FeRAM Cell, 583
- Fermi Level, 615
- Ferroelectric Effect, 582
- Ferroelectric RAM, see FeRAM 582
- Fetch Throttling, 628
- Filter Reuse, 716
- Fine-grained Multithreading, 424
- Finite Difference Method, 621
- Finite Element Method, 623
- FIT Rate, 638
- Flash
 - program/erase cycle, 581
- Flash Clearing, 514
- Flash Memory, 577
 - block, 580
 - floating gate transistor, 578
 - page, 580
 - read disturbance, 582
 - wear leveling, 581
- Flattened Networks, 341
- Flit, 346
- Floating Gate Transistor, 578
- Flow Control, 347
 - credit based, 348
 - on-off, 349
- Flush+Reload, 698
- Flushing the Pipeline, 138
- Flynn's Classification, 423
- Folded Bit Line Array Architecture, 541
- Folded Torus Topology, 340
- Folding (execution folding), 714
- Forward Biasing, 616
- Forward Slice, 162
- Forwarding, 39
- Forwarding Paths, 40
- Fourier's Law, 620
- Fowler-Nordheim Tunneling, 616
- FPGA, 215
- FPM, 553
- Fragment Processor, 223
- Frame, 274
- Frame Buffer, 224
- Free list, 106
- Frequency, 43
- Freshness, 683
- From-read Edge, 452
- Fully Associative Cache, 262
- Fully Buffered DIMMs, see FB-DIMMs 559
- Fully Connected Layer, 709
- Fully Mapped Scheme, 473
- GAg Predictor, 76
- GAP Predictor, 74
- Gate Oxide Tunneling, 616
- Gate-Induced Drain Leakage, 617

- GB, *see* Global Buffer 713
- GIDL, 617
- Global Buffer, 713
- Global History, 74
- Global Miss Rate, 268
- Global Wires, 338
- GPC, 234
- GPGPU, 218, 233
- GPU, 213
 - shader, 218
- Graph, 49
- Graphics Processing Cluster(GPC), 233
- Graphics Processors, 213
- Green's Functions, 624
- Grid, 226
- Ground Plane, 340
- GShare Predictor, 79
- Gustafson-Barsis's Law, 422

- Halo Pixels, 717
- Happens-before Relationship, 448
 - global, 448
 - local, 448
- Hard Disk, 272
- Hard Errors, 662
- Hardware Threads, 424
- Hardware Transactional Memory, 511, 524
- Harvard Architecture, 254
- Hazucha-Svensson Model, 640
- HCI, 663
- Head-of-Line(HOL) Blocking, 358
- Heat Sink, 618
- Heat Transfer Mechanisms, 619
- Helper Threads, 329
- High Bandwidth Memory, 593
- Hit-Miss Predictor, 161
- Home Bank, 402
- Host Process, 693
- Hot Carrier Injection, *see* HCI 663
- Hot-Carrier Injection, 616
- HP NonStop, 645
- HTM, *see* Software Transactional Memory 511,
 - see* Hardware Transactional Memory 524
- Hybrid Access Policy, 573
- Hybrid Memory Cube, 593
- Hypercubes, 341

- i-cache , *see* Instruction Cache 257
- IBM G5, 645
- ifmap, *see* Input Feature Map 708
- ILP, 47

- Induction Variables, 182
- Inductive Noise, 648
- Ineffectual Operations, 730
- Inferencing, 708
- Injection Rate, 407
- Input Buffering, 378
- Input Feature Map, 708
- Input Stationary, 717
- Instruction Cache, 257
- Instruction Commit, 135
- Instruction Compression, 88
- Instruction Decoding, 83
- Instruction Issue, 108
- Instruction Level Parallelism, 47
- Instruction Packet, 31, 62, 84
- Instruction Prefetching, 319
- Instruction Retirement, *see* Instruction Commit 135
- Instruction Select, *see* Select 114
- Instruction Status Table, 65
- Instruction Window, 108
- Integrity, 671
- Intel Processors, 749
- Intel SGX, 687
- Interposer, 593
- IPC, 43
- IS, *see* Input Stationary 717
- Isothermal Boundary, 618
- Issue, 108
- Issue Throttling, 628
- Issue Width, 108
- Itanium, 197
 - register remapping, 200
 - aggressive speculation, 207
 - instruction group, 200
 - load boosting, 207
 - multi-way branch, 199
 - predication, 205
 - register stack engine, 203
 - register stack frame, 201
 - stop bit, 200
 - virtual registers, 200
- JIT Compilation, 225

- Kerckhoff's Principle, 685
- Kernel, 226

- L1 Cache, 257
- Last Level Cache, 535
- Latency Speculation, 159
- Lateral Heat Conduction, 619

- LB, *see* Local Buffer 713
- Leakage Power, 614
- Leakage Power Management, 628
- Leakage-Temperature Feedback Loop, 608
- Legal Execution, 430
- Legal Sequential Execution, 431
- LET Rate, 639
- Line Edge Roughness, 657
- Linear Regression, 705
- Links, 337
- Livelock, 362
- LLC, 535, 546
- Load Latency Speculation, 152
- Load Linked and Store Conditional, 485
- Load-store Forwarding, 128
- Load-store Queue, *see* LSQ 125
- Load-use Hazard, 42
- Local Buffer, 713
- Local History, 74
- Local Miss Rate, 268
- Local Wires, 338
- Lock, 480
- Lock Set Algorithm, 504
- Lock-free Algorithms, 487
- Logical Masking, 642
- Lookahead Routing, 398
- Lookup Tables, 131
- Loop Fusion, 183
- Loop Invariant based Code Motion, 181
- Loop Optimizations, 181
- Loop Predictor, 72
- Loop Tiling, 308
- Loop Unrolling, 184
- Loop-Carried Dependence, 189
- Loosely Coupled Multiprocessors, 417
- LRU, 267
- LSQ, 125
 - design, 129
 - forwarding, 128
 - load queue and store queue, 129
- LVPT Table, 161
- MAC, 682, 688
- Man-in-the-middle Attack, 681
- Mapping of Loops, 713
- Marking an Instruction, 138
- Markov Prefetching, 320
- Markov Property, 320
- Matrix Arbiter, 390
- Measurement, 686
- Memory Address, 28
- Memory Consistency, 427, 446, 489
- Memory Consistency Model, *see* Memory Model 446
- Memory Controller, 571
- Memory Dependences, 126
- Memory Encryption Engine, 688
- Memory Model, 427, 446
- Memory Models, 489
- Memory Operation, 429
- Memory Wall, 535
- Merkle Tree, 690
- Mesh Topology, 340
- MESI Protocol, 467
- Message, 346
- Message Authentication Code, *see* MAC 682
- Metal Layers, 339
- Micro-operation, *see* Microinstruction 84
- Microbumps, 593
- Microcode Memory, 317
- Microinstruction, 84
- MIMD, 423
- Minimally Oblivious Routing, 371
- MISD, 423
- Miss Penalty, 270
- Miss Status Handling Register, *see* MSHR 304
- Mixed Mode Accesses, 516
- Model Checking, 652
- MOESI Protocol, 469
- Moore's Law, 14
- MPMD, 423
- MRAM, 585
- MSHR, 304
 - primary miss, 304
 - secondary miss, 305
- MTBF, 638
- MTTF, 638
- MTTR, 638
- Multicore, 415
- MultiMaster, 645
- Multiply and Accumulate Operation, 712
- MultiSlave, 645
- Multithreaded Execution, 424
- Mutual Exclusion, 482
- NBTI, 663
- Negative Bias Temperature Instability, *see* NBTI 663
- Network Diameter, 340
- Network Topology, 338
 - bus, 338
- Neural Networks, 705

- inferencing, 708
- training, 708
- Neuromorphic Circuits, 735
- Next Line Prefetching, 319
- NLR, *see* No Local Reuse Architectures 715
- No Local Reuse Architectures, 715
- No-Dep-Cycle, 458
- NoC, 335
- NoC Evaluation Metrics, 406
- NoC Simulation, 407
- Node Table based Routing, 381
- Non-blocking Caches, 304
- Non-Selective Replay, 163
- Non-uniform Cache Architectures, *see* NUCA Architectures 401
- Nonces, 683
- Nondeterminism, 649
- Nonvolatile Memory, 577
- NUCA Architectures, 401
- nvcc, 225
- NVM, 577
- Oblivious RAM, *see* ORAM 695
- Oblivious Routing, 371
- Off-Axis Illumination, 661
- ofmap, *see* Output Feature Map 708
- On-chip Network, 335
- On-Off Flow Control, 349
- One-Time Pad, 675
- OOO Processor, 48
- Opacity, 516
- Open Bit Line Array Architecture, 541
- Open Page Access Policy, 573
- OpenCL, 225
- Operand Fetch, 27
- Optical Proximity Correction, 661
- ORAM, 695
- Orphan Instructions, 167
- OS, *see* Output Stationary 717
- Out-of-order Processor, 48
- Output Feature Map, 708
- Output Feedback Mode, 675
- Output Stationary, 717
- Oxide Thickness Variation, 657
- Oxygen Vacancy, 591
- P-N Junction Reverse Bias Current, 614
- Packet, 346
- PAG Predictor, 76
- Page, 274
- Page Fault, 277
- Page Fault Snooping Attacks, 694
- Page Table, 276
- Page Walk, 276
- PAP Predictor, 78
- Parallel Execution, 432
- Parallel Reduction, 719
- Parameter Variation, 656
- Partial Store Ordering, *see* PSO 494
- Partial Sum, 712
- Partially Mapped Scheme, 476
- Pass Transistors, 283
- Path Diversity, 341, 371
- Path ORAM, 695
- Pattern History Table, 75
- PC Address, 28
- PCM, 588
- PE, 712
- Per-location Sequential Consistency, *see* PLSC 437
- Performance Equation, 43
- Phase Change Memory, *see* PCM 588
- Phase Shift Masking, 661
- Phit, 347
- Photolithography, 656
- PHT, *see* Pattern History Table 75
- Physical Addresses, 274
- Physical Memory Space, 274
- Physical Register File, 98
- Physical Registers, 98
- Physically Unclonable Function, *see* PUF 679
- PIM, 733
- Pipeline Bubble, 34
- Pipeline Damping, 648
- Pipeline Diagram, 33
- Pipelined Caches, 302
- Pixel, 708
- Pixel Engine, 224
- Placement, 46
- Plaintext, 671
- PLSC, 437, 457
- Point of View, 429
- Pointer Chasing, 327
- Poison Bit, 168
- Polarization, 582
- Polymorph Engine, 221, 234
 - attribute setup, 222
 - hull shader, 221
 - stream output, 222
 - tessellation, 222
 - vertex shader, 221
 - viewport, 222
 - viewport transformation, 222

- window, 222
- Pooling Layer, 708
 - max pooling, 708
- Port, 291
- Power Gating, 628
- Power Gating Noise, 629
- Power Grid, 340
- Power Management, 625
- Power Models, 608
- Power of Atomic Operations, 488
- Power Plane, 340
- PowerPC Memory Model, 494
- Pre-execution Techniques, 328
- Precharging, 281
- Precise Exceptions, 56, 134
- Predecoder, 84
- Predicated Execution, 195, 206, 239
- Prefetch Length, 555
- Prefetch Width, 558
- Prefetching
 - data, see Data Prefetching 324
 - instructions, see Instruction Prefetching 319
- Prime+Probe, 697
- Private Key, 678
- Process, 271, 419
- Process Variation, 656
- Processing Block, 236
- Processing Element, see PE 712
- Processing in Memory, 733
- Processor Consistency, 494
- Profiling, 162, 322
- Program Order, 48
- Program Order Edge, 450
- Program Phase, 256
- Programming GPGPUs, 225
- Properly Synchronized Programs, 502
- Pseudo-LRU, 267
- PSO, 494
- PTX, 225
- Public Key, 678
- PUF, 679
- Punchthrough Current, 617
- QDR Memory, 557
- Quad Data Rate Memory, 557
- Qualcomm Processors, 759
- Quoting Enclave, 693
- Radiation, 619
- Radiation Hardening, 641
- Random Variation, 657
- Rank, 572
- RAW Dependence, 33, 54
- RC4, 676
 - hardware implementation, 677
- RC4 Stream Cipher, 676
- Read Leveling, 566
- Read Postamble, 566
- Read Preamble, 566
- Read-from Edge, 451
- Reconvergence, 241
- Redo Log, 515
- Redox ReRAMs, 590
- Reduce Operation, 421
- Reduction, 713
- Refresh, 536
- Refresh Management, 573
- Register Alias Table, 99
- Register Release, 137
- Register Renaming, 54
- Register Spilling, 203
- Register Write-back, 28
- Registered Memory, 562
- Relations, 449
- Release Consistency, 494
- Release Instruction, 445
- Reliability, 637
- ReLU Function, 706
- Remote Attestation, 687
- Rename Table, 98, 99
- Renaming, 96
- Rendering, 219
- Reorder Buffer, 135
- Repeaters, 337
- Replay, 162
- Replay Attack, 682
- Replay Queue, 166
- ReRAM, 590
- Resample and Resync, 561
- Reservation Stations, 177
- Resistive RAM, see ReRAM 590
- Resistive Switching Phenomenon, 590
- Restoring a Read, 539
- Retiming Lemma, 726
- Retirement, 134
- Retirement Register File, 140
- Return Address Stack, 82
- Reverse Biasing, 616
- ROB, see Reorder Buffer 135
- ROB Entry, 139
- Roofline Model, 594
- Root of Trust, 686

- Round Robin Arbitration, 389
- Route Computation, 380
- Router, 337
 - design, 378
 - pipeline, 395
- Router Bypassing, 398
- Routing, 46, 361
 - dynamic, 361
 - static, 361
- Row Stationary, 717
- Rowhammer Attack, 698
- RS, *see* Row Stationary 717
- Runahead Execution, 328

- SASS, 225
- Saturating Counter, 70
- Scan Chain, 652
- Scheduling, 108
- Scoreboarding, 203
- Sealing, 687
- Secure Architectures, 684
- Security, 671
- Select, 114
 - tree based select unit, 114
- Semi-systolic Array, 725
- Semiconductor Package, 617
- Sense Amplifier, 284, 541
- Separable Allocator, 391
- Sequential Consistency, 435, 494
- Sequential Execution, 430
- Serializability, 516
- Serialization, 430
- Session Keys, 679
- Set Associative Cache, 264
- SHA, 681
- shader, 218
- Shared Cache, 425
- Shmooring, 652
- Short-Circuit Power, 613
- Shuffle Traffic, 409
- Side Channel, 696
- Side-Channel Attacks, 696
 - classification, 696
 - countermeasures, 699
- Sigmoid Function, 706
- Signals, 654
- Silent Data Corruption, 643
- Silent Stores, 181
- Silicon Traps, 663
- SIMD, 423
- SimpleRisc ISA, 741
- SIMT, 237
- Simulator, 744
- Simultaneous Multithreading, 424
- Single Cycle Processor, 30
- Single Lock Atomicity, 517
- SingleSlave, 645
- SISD, 423
- Skewed Associative Caches, 305
- Sleep State, 629
- Sleep Transistors, 629
- SM, 235
- Snoopy Protocol, 459, 464
- SoC, 755
- Soft Errors, 638
- Software Pipelining, 186
 - epilogue, 191
 - prologue, 191
- Software Transactional Memory, 511, 518
- Source Routing, 380
- Source Synchronous Bus, 557
- Space-Time Diagram, 353
- Spatial Locality, 255
- Spatial Reuse, 718
- Speculation, 152
- Speculative Execution Attacks, 699
- Speculative VC Allocation, 399
- Spin Lock, 444, 483
- Spin-Transfer Torque, 586
- Splicing Attack, 681
- SPMD, 423
- Spoofing Attack, 681
- Squashing, 162
- SRAM Array, 279
- SRAM Cell, 278
- Stacked Capacitors, 538
- Start-gap Wear Leveling, 590
- Starvation, 362, 483
- Static NUCA, 401
- Stationarity, 715
- Statistical Static Timing Analysis, 661
- Steady-state Analysis, 618
- Steady-state Thermal Analysis, 619
- STM, *see* Hardware Transactional Memory 511,
see Software Transactional Memory 518
- Stop-and-go, 632
- Store and Forward Flow Control, 354
- Store Set, 157
- Stream Cipher, 672
- Streaming Multiprocessor, 235
- Streaming Processors, 236
- Strength Reduction, 179

- Stress Migration, 664
- Strict Serializability, 516
- Stride, 155, 324
- Stride based Prefetching, 324
- Strongly Coupled Multiprocessors, 417
- STT-MRAM, 586
- Sub-resolution Assist Features, 661
- Subthreshold Leakage, 615
- Summarization, 655
 - spatial, 655
 - temporal, 655
- Superscalar Processor, 48
- Support Vector Machines, 705
- Swap Space, 272
- Switch Allocation, 382
- Switch Traversal, 384
- Symmetric Ciphers, 678
- Synchronization Edge, 453
- Synchronization Instructions, 444
- Synchronous DRAM, 555
- Synthetic Traffic Generation, 407
- System Call, 138
- Systematic Variation, 657
- Systolic Array, 725
- Systolic Arrays, 727
- Systolic Transfer, 719
- Tag Bus, 113
- Tail Recursion, 202
- TCB, *see* Trusted Computing Base 685
- TEE, *see* Trusted Execution Environment 686
- Tejas Architectural Simulator, 743
- Temperature Management, 631
- Temperature Map, 618
- Temperature Model, 617
- Temporal Locality, 256
- Temporal Reuse, 715
- Tessellation, 221
- Test and Exchange, 484
- Test and set, 485
- Texture Mapping, 223
- Thermal Cycling, 665
- Thermal Interface Material, 618
- Thermal Runaway, 608
- Thermal-electrical Analogy, 622
- Thin Air Read, 458
- Thread, 226, 329, 417
- Thread Engine, 233
- Threat Model, 686
- Throughput, 353
- Tile, 337
- Time-dependent Dielectric Breakdown, 664
- Timing Vulnerability Factor, 643
- Timing Window Masking, 642
- TL2 STM, 521
- TLB, 276
- TMR, *see* Triple Modular Redundancy 645
- Token based Replay, 170
- Tomasulo's Algorithm, 177
- Topological Sort, 454
- Tornado Traffic, 410
- Torus Topology, 340
- Total Store Ordering, *see* TSO 494
- Tournament Predictor, 79
- TPC, 235
- Trace, 314
- Trace Based NoC Simulators, 410
- Trace Buffers, 655
- Trace Cache, 313
- Training, 708
- Trans-silicon Vias, 339
- Transactional Memory, 509, 511
- Transactional Sequential Consistency, 518
- Transactions, 511, 512
 - conflict, 512
 - eager version management, 514
 - lazy version management, 515
 - read set, 512, 518
 - version management, 513
 - write set, 512, 518
- Transfers per Second, 558
- Transient Analysis, 618
- Transient Faults, 638
- Transient Thermal Analysis, 619
- Translation Lookaside Buffer, *see* TLB 276
- Transpose Traffic, 408
- Trench Capacitors, 537
- Triple Modular Redundancy, 645
- True Sharing, 478
- Trusted Computing Base, 685
- Trusted Execution Environment, 686
- Trusted Platform Module, 686
- TSO, 494
- TSVs, 339
- Tunneling Magnetoresistance, 585
- Turn Graph, 367
- Turn Model, 373
- Turns, 373
- Undo Log, 514
- Uniprocessor Access Constraint, 455
- Universal Approximator, 704

- Unlock, 480
- Valence Band, 615
- Valiant's Algorithm, 371
- Validation, 651, 652
- Value Prediction, 161
- Vault, 593
- Vector Clocks, 505
- Verification, 651
- Vertical Heat Conduction, 619
- Vias, 339
- Virtual Channel Allocation, 382
- Virtual Channels, 359
 - deadlock freedom, 375
- Virtual Cut Through Flow Control, 356
- Virtual Memory, 271, 272
- Virtually Indexed Physically Tagged Caches, 311
- VLIW, 194
- Volatile Memory, 577
- Von Neumann Architecture, 254
- Wait-free algorithms, 488
- Wakeup, 112
- WAR Dependence, 54
- Warp, 236
- Wavefront Allocator, 392
- WAW Dependence, 54
- Way Prediction, 307
- Weak Consistency, 494
- Weak Ordering , see Weak Consistency 494
- Wear Leveling, 581
- Weight Reuse, 716
- Weight Stationary, 716
- Window of Vulnerability, 163
- Workarounds for Design Faults, 654
- Working Set, 256
- Wormhole Flow Control, 357
- WP Axiom, 443
- Write Caching, 573
- Write Cancellation, 589
- Write Leveling, 568
- Write Pausing, 589
- Write Postamble, 568
- Write Preamble, 568
- Write Propagation Axiom, 443
- Write Recovery Time, 546
- Write Serialization Axiom, 443
- Write Serialization Edge, 452
- Write-Invalidate Protocol, 464
- Write-Update Protocol, 459
- Wrong Path, 36, 52
- WS, see Weight Stationary 716
- WS Axiom, 443
- X-Y Routing, 369
- Z-depth, 224