Journal 2, CS6000

Mong Sim
Computer Science Department
University of Colocrado Colorado Springs
Colorado Springs, USA
msim@uccs.edu

I. Learning Process

My paper search was from ACM and IEEE sites. I set the latest date first since I am more interested in the recent papers. My paper search is much broader because I am interested in fault tolerance on both software and hardware. Most of the articles I found by looking at the title and reading the abstract to decide if the paper is worth further reading. One inconvenience on the website is that it does not remember which document we select and on return, it always goes back to the first document of that page. Trying to find a specific paper without knowing the title can be very different to find if one is going through these sites in sequential order. One other thing I did which I find it very useful is to rename the file with the research title.

II. Raw Notes

All the papers [1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12] [13] [14] [15] [16] [17] [18] [19] [20] [21] [22] [23] [24] [25] [26] [27] [28] [29] [30] [31] [32] [33] [34] [35] [36] [37] [38] [39] [40] [41] [42] [43] [44] [45]

Hardware-Assisted Fault Tolerance [6] utilizes instruction-level redundancy for fault detection and hardware transactional memory for fault recovery. Issues are amplified in the new processor architectures that are continuously boosting performance with higher circuit density using ever-shrinking transistor sizes. A Fault-Tolerant Non-Volatile Main Memory File System [22] proposes solutions for the unique challenges in adding fault tolerance to an NVMM file system. Atomic updates to implement features such as complex atomic data and metadata updates, but doing so requires different data structures and algorithms than block-based file systems have employed. Let's shock our IoT's heart: ARMv7-M under (fault) attacks [28] explore howElectromagnetic fault injection (EMFI) can be used to create vulnerabilities in sound software. Modulo operations are utilized to reduce the hardware overhead. The proposed method can detect faults with more than 70% accuracy and more than 87% fault coverage within an acceptable test time. A 1GHz Fault Tolerant Processor with Dynamic Lockstep and Self-recovering Cache for ADAS SoC Complying with ISO26262 in Automotive Electronics [35] present a processing platform that implements DMR with separate clock and power sources. Fault tolerant processor contains three key features: 1) dynamic lockstep (DLS) with separate clock and power sources to reduce dependent failures and have the high performance, 2) the cache with self-recovering function to reduce transient faults, 3) reconfigurable function to reduce permanent faults. TMR Group Coding Method for Optimized SEU and MBU Tolerant Memory Design [38] proposes a memory rich micro-coded processor. Triple Module Redundancy (TMR) and Error Correcting Code (ECC) are two commonly used fault tolerant approaches. TMR can ensure the right output in the case of single module fault, but it requires high hardware redundancy. The hardware cost for ECC is less while its ability to detect and correct soft errors is limited. Processor checkpoint recovery for transient faults in critical applications [39] an approach to implement checkpoint recovery on FPGA with application-level transparency. Dual Modular Redundancy (DMR) and CR configuration is used in order to find the optimal checkpoint selection. An integrated design environment of fault tolerant processors with flexible HW/SW solutions for versatile performance/cost/coverage tradeoffs [43] provides for the designer to select the options and the IDE automatically generates the hardware Verilog code and the modified embedded software. It injects faults by modifying the C code or assembly code. EDFI analyzes and duplicates the control flow to inject the faults. A fault-tolerant real-time microcontroller with multiprocessor architecture [45] proposes working in TMR and some spare processors. Leverage the hardware redundancy of a multiprocessor architecture, doing work together three processors at TMR configuration and using the remaining processors as spare parts to tolerate the permanent faults.

References

[1] L. Su and N. H. Vaidya, *Fault-Tolerant Multi-Agent Optimization: Optimal Iterative Distributed Algorithms*. ACM New York, NY, USA: ACM Symposium on Principles of Distributed Computing, 2016 Note=trash, 100 sec.

[2] H. Homulle, S. Visser, B. Patra, G. Ferrari, E. Prati, C. G. Almudéver, K. Bertels, F. Sebastiano, and E. Charbon, *CryoCMOS hardware technology a classical infrastructure for a scalable quantum computer*. ACM New York, NY, USA: ACM Symposium on Principles of Distributed Computing, 2016 Note=trash, 120 sec.

[3] S. Hukerikar, R. A. Ashraf, and C. Engelmann, *Towards New Metrics for High-Performance Computing Resilience*. ACM New York, NY, USA: ACM Workshop on Fault-Tolerance for HPC at Extreme Scale, 2017 Note=trash, 110 sec.

[4] Z. W. Parchman, G. R. Vallee, T. Naughton, C. Engelmann, D. Bernholdt, and S. L. Scott, *Adding Fault Tolerance to NPB Benchmarks Using ULFM*. ACM New York, NY, USA: ACM Workshop on Fault-Tolerance for HPC at Extreme Scale, 2016 Note=trash, 90 sec.

[5] R. Gensh, A. Romanovsky, and A. Yakovlev, *On structuring holistic fault tolerance*. ACM New York, NY, USA: ACM 15th International Conference on Modularity, 2016 Note=trash, 100 sec.

[6] D. Kuvaiskii, R. Faqeh, P. Bhatotia, P. Felber, and C. Fetzer, *HAFT: hardware-assisted fault tolerance*. ACM New York, NY, USA: EuroSys '16 Proceedings of the Eleventh European Conference on Computer Systems, 2016 Note=scan, 130 sec.

[7] K. Echtle and Z. Moztarzadeh, *Fault-tolerant clock synchronization in ring-networks*. ACM New York, NY, USA: SAC '17 Proceedings of the Symposium on Applied Computing, 2017 Note=trash, 80 sec.

[8] M. Parter and D. Peleg, *Sparse Fault-Tolerant BFS Structures*. ACM New York, NY, USA: ACM Transactions on Algorithms (TALG) TALG Homepage archive, 2016 Note=trash, 130 sec.

[9] N. A. Rink and J. Castrillon, *Trading Fault Tolerance for Performance in AN Encoding*. ACM New York, NY, USA: CF'17 Proceedings of the Computing Frontiers Conference, 2017 Note=trash, 190 sec.

[10] M. Parter and D. Peleg, *Fault-Tolerant Approximate BFS Structures*. ACM New York, NY, USA: ACM Transactions on Algorithms (TALG) TALG Homepage archive, Journal, 2018 Note=trash, 108 sec.

[11] M. S. Ardekani and R. P. Singh, *Rivulet: a fault-tolerant platform for smart-home applications*. ACM New York, NY, USA: Middleware '17 Proceedings of the 18th ACM/IFIP/USENIX Middleware Conference, 2017 Note=trash, 100 sec.

[12] Y. Harchol, D. Hay, and T. Orenstein, *FTvNF:fault tolerant virtual network functions*. ACM New York, NY, USA: ANCS '18 Proceedings of the 2018 Symposium on Architectures for Networking and Communications Systems, 2017 Note=trash, 90 sec.

[13] I. Perez, *Fault tolerant functional reactive programming (functional pearl)*. ACM New York, NY, USA: ACM on Programming Languages archive, 2018 Note=trash, 100 sec.

[14] S. Karol, N. A. Rink, B. Gyapjas, and J. Castrillon, *Fault tolerance with aspects- a feasibility study*. ACM New York, NY, USA: 15th International Conference on Modularity, 2016 Note=trash, 130 sec.

[15] N. Chatterjee, S. Paul, and S. Chattopadhyay, *Fault-Tolerant Dynamic Task Mapping and Scheduling for Network-on-Chip-Based Multicore Platform*. ACM New York, NY, USA: ACM Transactions on Embedded Computing Systems (TECS) - Special Issue on Secure and Fault-Tolerant Embedded Computing and Regular Papers TECS Homepage archive, 2017 Note=trash, 110 sec.

[16] S. Bhagat and K. Mukhopadyaya, *Fault-tolerant Gathering of Semi-synchronous Robots*. ACM New York, NY, USA: ACM Transactions on Embedded Computing Systems (TECS) - Special Issue on Secure and Fault-Tolerant Embedded Computing and Regular Papers TECS Homepage archive, 2017 Note=trash, 90 sec.

[17] G. Chockler and A. Spiegelman, *Space Complexity of Fault-Tolerant Register Emulations*. ACM New York, NY, USA: PODC '17 Proceedings of the ACM Symposium on Principles of Distributed Computing, 2017 Note=trash, 110 sec.

[18] A. Benoit, A. Cavelan, F. Cappello, P. Raghavan, Y. Robert, and H. Sun, *Identifying the Right Replication Level to Detect and Correct Silent Errors at Scale*. ACM New York, NY, USA: FTXS '17 Proceedings of the 2017 Workshop on Fault-Tolerance for HPC at Extreme Scale, 2017 Note=trash, 160 sec.

[19] K. S.-M. Li and S.-J. Wang, *Design Methodology of Fault-Tolerant Custom 3D Network-on-Chip*. ACM New York, NY, USA: ACM Transactions on Design Automation of Electronic Systems (TODAES) TODAES Homepage archive, 2017 Note=trash, 100 sec.

[20] L. Zeng, P. Huang, and L. Thiele, *Towards the design of fault-tolerant mixed-criticality systems on multicores*. ACM New York, NY, USA: CASES '16 Proceedings of the International Conference on Compilers, Architectures and Synthesis for Embedded Systems, 2016 Note=trash, 90 sec.

[21] O. Subasi, O. Unsal, and S. Krishnamoorthy, *Automatic Risk-based Selective Redundancy for Fault-tolerant Task-parallel HPC Applications*. ACM New York, NY, USA: ESPM2'17 Proceedings of the Third International Workshop on Extreme Scale Programming Models and Middleware, 2017 Note=trash, 120 sec.

[22] J. Xu, L. Zhang, A. Memaripour, A. Gangadharaiah, A. Borase, T. B. D. Silva, S. Swanson, and A. Rudoff, *NOVA-Fortis: A Fault-Tolerant Non-Volatile Main Memory File System*. ACM New York, NY, USA: SOSP '17 Proceedings of the 26th Symposium on Operating Systems Principles, 2017 Note=scan, 110 sec.

[23] A. L. Sartor, A. F. Lorenzon, S. Kundu, A. Gangadharaiah, I. Koren, and A. C. S. Beck, *Adaptive and polymorphic VLIW processor to optimize fault tolerance, energy consumption, and performance*. ACM New York, NY, USA: CF '18 Proceedings of the 15th ACM International Conference on Computing Frontiers, 2018 Note=trash, 130 sec.

[24] K. Ma, X. Li, H. Liu, X. Sheng, Y. Wang, K. Swaminathan, Y. Liu, Y. Xie, J. Sampson, and V. Narayanan, *Dynamic Power and Energy Management for Energy Harvesting Nonvolatile Processor Systems*. ACM New York, NY, USA: ACM Transactions on Embedded Computing Systems (TECS) - Special Issue on Secure and Fault-Tolerant Embedded Computing and Regular Papers TECS Homepage archive, 2017 Note=trash, 140 sec.

[25] A. L. Sartor, A. F. Lorenzon, L. Carro, F. Kastensmidt, S. Wong, and A. C. S. Beck, *Exploiting Idle Hardware to Provide Low Overhead Fault Tolerance for VLIW Processors*. ACM New York, NY, USA: ACM Transactions on Embedded Computing Systems (TECS) - Special Issue on Secure and Fault-Tolerant Embedded Computing and Regular Papers TECS Homepage archive, 2017 Note=trash, 160 sec.

[26] R. Cammarota and R. Sheikh, *VPsec: countering fault attacks in general purpose microprocessors with value prediction*. ACM New York, NY, USA: CF '18 Proceedings of the 15th ACM International Conference on Computing Frontiers, 2018 Note=trash, 180 sec.

[27] L. Xia, M. Liu, X. Ning, K. Chakrabarty, and Y. Wang, *Fault-Tolerant Training with On-Line Fault Detection for RRAM-Based Neural Computing Systems*. ACM New York, NY, USA: DAC '17 Proceedings of the 54th Annual Design Automation Conference 2017, 2017 Note=trash, 140 sec.

[28] S. K. Bukasa, R. Lashermes, J.-L. Lanet, and A. Leqay, *Let's shock our IoT's heart: ARMv7-M under (fault) attacks*. ACM New York, NY, USA: ARES 2018 Proceedings of the 13th International Conference on Availability, Reliability and Security, 2018 Note=trash, 100 sec.

[29] Y. Luo, S. Ghose, E. F. Haratsch, and O. Mutlu, *Improving 3D NAND Flash Memory Lifetime by Tolerating Early Retention Loss and Process Variation*. ACM New York, NY, USA: SIGMETRICS '18 Abstracts of the 2018 ACM International Conference on Measurement and Modeling of Computer Systems, 2018 Note=scan, 180 sec.

[30] B. Hasircioglu, Y.-A. Pignolet, and T. Sivanthi, *Transparent Fault Tolerance for Real-Time Automation Systems*. ACM New York, NY, USA: IoPARTS'18 Proceedings of the 1st International Workshop on Internet of People, Assistive Robots and Things, 2018 Note=trash, 100 sec.

[31] W. Liu, Z. Wei, and W. Du, *A Novel Fault-Tolerant Last-Level Cache to Improve Reliability at Near-Threshold Voltage*. ACM New York, NY, USA: GLSVLSI '18 Proceedings of the 2018 on Great Lakes Symposium on VLSI, 2018 Note=trash, 130 sec.

[32] J. Proy, K. Heydemann, A. Berzati, and A. Cohen, *Compiler-Assisted Loop Hardening Against Fault Attacks*. ACM New York, NY, USA: ACM Transactions on Architecture and Code Optimization (TACO) TACO Homepage archive, 2017 Note=trash, 120 sec.

[33] G. Lee, E. Cetin, and O. Diessel, *Fault Recovery Time Analysis for Coarse-Grained Reconfigurable Architectures*. ACM New York, NY, USA: ACM Transactions on Embedded Computing Systems (TECS) - Special Issue on MEMCODE 2015 and Regular Papers (Diamonds) TECS Homepage archive, 2018 Note=trash, 150 sec.

[34] F. F. dos Santos and P. Rech, *ScalA '17 Proceedings of the 8th Workshop on Latest Advances in Scalable Algorithms for Large-Scale Systems*. ACM New York, NY, USA: ACM Transactions on Embedded Computing Systems (TECS) - Special Issue on MEMCODE 2015 and Regular Papers (Diamonds) TECS Homepage archive, 2017 Note=trash, 100 sec.

[35] J. Han, Y. Kwon, Y. Cheol, P. Cho, and H.-J. Yoo, *A 1GHz Fault Tolerant Processor with Dynamic Lockstep and Self-recovering Cache for ADAS SoC Complying with ISO26262 in Automotive Electronics*. Seoul, South Korea: IEEE Asian Solid-State Circuits Conference (A-SSCC), 2017 Note=scan, 170 sec.

[36] J. Xu, *Handling Process Overruns and Underruns on Multiprocessors in a Fault-Tolerant Real-Time Embedded System*. Oulu, Finland, Finland: 2018 14th IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications (MESA), 2018 Note=trash, 100 sec.

[37] K. Arora, S. Bansal, and R. K. Bansal, *Energy Aware Fault Tolerant Fixed Priority Task Scheduling in Multiprocessor System*. Noida, India, India: IEEE 2018 8th International Conference on Cloud Computing, Data Science & Engineering (Confluence), 2018 Note=trash, 90 sec.

[38] Y. Jin, Y. Huan, H. Chu, Z. Zou, and L. Zheng, *TMR Group Coding Method for Optimized SEU and MBU Tolerant Memory Design*.

Florence, Italy, Italy: IEEE International Symposium on Circuits and Systems (ISCAS), 2018 Note=scan, 150 sec.

[39] P. R. C. Villa, R. Travessini, F. L. Vargas, and E. A. Bezerra, *Processor checkpoint recovery for transient faults in critical applications*. Sao Paulo, Brazil: IEEE 19th Latin-American Test Symposium (LATS), 2018 Note=scan, 170 sec.

[40] C. M. Fuchs, T. P. Stefanov, N. M. Murillo, and A. Plaat, *Bringing Fault-Tolerant GigaHertz-Computing to Space: A Multi-stage Software-Side Fault-Tolerance Approach for Miniaturized Spacecraft*. Taipei, Taiwan: IEEE 26th Asian Test Symposium (ATS), 2017 Note=trash, 110 sec.

[41] C. Mohamedin, M. J. Kishi, and R. Palmieri, *Shield- A middleware to tolerate CPU transient faults in multicore architectures*. Cambridge, MA, USA: IEEE 16th International Symposium on Network Computing and Applications (NCA), 2017 Note=trash, 100 sec.

[42] S. H. M. Hedesh and M. R. Zolghadri, *Open-circuit fault detection and localization in five-level active NPC converter*. Beijing, China: IECON 2017 - 43rd Annual Conference of the IEEE Industrial Electronics Society, 2017 Note=trash, 90 sec.

[43] Y.-J. Ke, Y.-C. Ghen, and J.-J. Huang, *An integrated design environment of fault tolerant processors with flexible HW/SW solutions for versatile performance/cost/coverage tradeoffs*. Taipei, Taiwan: International Test Conference in Asia (ITC-Asia), 2017 Note=scan, 190 sec.

[44] G. Losa, A. Barbalace, Y. Wen, H.-R. Chuang, and B. Ravindran, *Transparent Fault-Tolerance Using Intra-Machine Full-Software-Stack Replication on Commodity Multicore Hardware*. Atlanta, GA, USA: IEEE 37th International Conference on Distributed Computing Systems (ICDCS), 2017 Note=trash, 110 sec.

[45] E. Strollo and A. Trifiletti, *A fault-tolerant real-time microcontroller with multiprocessor architecture*. Lodz, Poland: MIXDES - 23rd International Conference Mixed Design of Integrated Circuits and Systems, 2016 Note=scan, 150 sec.