Octopus: A Gem5-Integrated Rapid Prototyping for Resource Contention Measurement and Control

Mohamed Hossam*, Shorouk Abdelhalim*, Mohamed Hassan*, Rodolfo Pellizzoni**, Danesh Germchi**
*McMaster University, Canada, {mohamed.hossam, abdels28, mohamed.hassan}@mcmaster.ca
**University of Waterloo, Canada, {rpellizz, dgermchi}@uwaterloo.ca

To address the resource contention issue in high-performance real-time Systems-on-a-chip (SoCs), the research community has proposed many different designs for components in the memory hierarchy, ranging from DRAM memory controllers [1]–[12], to cache controllers [13], to bus and coherence designs [14]–[20]. Most such proposals have been evaluated using custom standalone simulators. This makes it difficult to compare alternative designs. Furthermore, our experience is that implementing the simulator from scratch requires significant time, slowing down testing of new ideas. Following the ECRTS Arm Industrial Challenge [21], we propose Octopus, a gem5-integrated architectural simulator [22] as a prototyping platform to measure and control the impact of resource contention. Octopus is a highly modular simulation model for the whole memory hierarchy, including caches, interconnect and main memory, which is specifically designed to simplify the testing and implementation of predictable arbitration schemes for mixed-critical systems. Unlike Gem5's Ruby memory model [23], Octopus is designed with real-time systems as a first-class citizen. We further integrate our model with the ARM Adaptive Traffic Profile (ATP) gem5 engine [24] to simulate the injection of various traffic profiles for competing masters in the systems. As an example of the potentialities offered by our framework, we implemented and tested our hardware management scheme published in the main conference [25], which provides much tightened end-to-end latency bounds for memory requests through coordination of multiple resource arbiters. We plan to continue supporting our simulation framework, which we intend to release as open-source, and to expand it in multiple directions. First of all, we will add support for full-system simulation with coherent caches, thus allowing us to simulate a symmetric OS deployment and to execute the OV^{2}SLAM application [26] suggested in the Industrial Challenge. We will also seek to integrate the Gem5 AMD GPU model [27], which would allow us to run the Hopenet Head Pose Estimation application [28] on the simulated GCN3 GPU using the Radeon Open Compute platform (ROCm), rather than injecting GPU-like traffic profiles through the ATP.

References

[1] B. Akesson and K. Goossens, *Memory controllers for real-time embedded systems*. Springer, 2011.

[2] L. Ecco, S. Tobuschat, S. Saidi, and R. Ernst, "A mixed critical memory controller using bank privatization and fixed priority scheduling," in *2014 IEEE 20th International Conference on Embedded and Real-Time Computing Systems and Applications*. IEEE, 2014, pp. 1–10.

[3] L. Ecco and R. Ernst, "Improved DRAM Timing Bounds for Real-Time DRAM Controllers Read/Write Bundling," in *Real-Time Systems Symposium*, 2015.

[4] L. Ecco, A. Kostrzewa, and R. Ernst, "Minimizing DRAM Rank Switching Overhead for Improved Timing Bounds and Performance," in *Euromicro Conference on Real-Time Systems (ECRTS)*, 2016.

[5] D. Guo and R. Pellizzoni, "A requests bundling dram controller for mixed-criticality systems," in *Real-Time and Embedded Technology and Applications Symposium (RTAS), 2017 IEEE*. IEEE, 2017, pp. 247–258.

[6] J. Jalle, E. Quinones, J. Abella, L. Fossati, M. Zulianello, and F. J. Cazorla, "A dual-criticality memory controller (dcmc): Proposal and evaluation of a space case study," in *2014 IEEE Real-Time Systems Symposium*, 2014.

[7] Y. Krishnapillai, Z. P. Wu, and R. Pellizzoni, "A rank-switching, open-row dram controller for time-predictable systems," in *2014 26th Euromicro Conference on Real-Time Systems*. IEEE, 2014, pp. 27–38.

[8] Y. Li, B. Akesson, and K. Goossens, "Dynamic command scheduling for real-time memory controllers," in *2014 26th Euromicro Conference on Real-Time Systems*. IEEE, 2014, pp. 3–14.

[9] R. Mirosanlou, M. Hassan, and R. Pellizzoni, "Drambulism: Balancing performance and predictability through dynamic pipelining," in *IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*, 2020.

[10] M. Paolieri, E. Quinones, F. J. Cazorla, and M. Valero, "An analyzable memory controller for hard real-time cmps," *IEEE Embedded Systems Letters*, vol. 1, no. 4, pp. 86–90, 2009.

[11] J. Reineke, I. Liu, H. D. Patel, S. Kim, and E. A. Lee, "PRET DRAM controller: bank privatization for predictability and temporal isolation," in *Proceedings of the seventh IEEE/ACM/IFIP international conference on Hardware/software codesign and system synthesis*, ser. CODES+ISSS '11. New York, NY, USA: ACM, 2011, pp. 99–108.

[12] P. K. Valsan and H. Yun, "MEDUSA: a predictable and high-performance DRAM controller for multicore based embedded systems," in *Cyber-Physical Systems, Networks, and Applications (CPSNA)*, 2015, pp. 86–93.

[13] G. Gracioli, A. Alhammad, R. Mancuso, A. A. Fröhlich, and R. Pellizzoni, "A survey on cache management mechanisms for real-time embedded systems," *ACM Computing Surveys (CSUR)*, vol. 48, no. 2, pp. 1–36, 2015.

[14] M. Hassan, A. M. Kaushik, and H. Patel, "Predictable cache coherence for multi-core real-time systems," in *2017 IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*. IEEE, 2017, pp. 235–246.

[15] N. Sritharan, A. Kaushik, M. Hassan, and H. Patel, "Enabling predictable, simultaneous and coherent data sharing in mixed criticality systems," in *2019 IEEE Real-Time Systems Symposium (RTSS)*, 2019, pp. 433–445.

[16] A. M. Kaushik, P. Tegegn, Z. Wu, and H. Patel, "Carp: A data communication mechanism for multi-core mixed-criticality systems," in *2019 IEEE Real-Time Systems Symposium (RTSS)*, 2019, pp. 419–432.

[17] A. M. Kaushik, M. Hassan, and H. Patel, "Designing predictable cache coherence protocols for multi-core real-time systems," *IEEE Transactions on Computers*, vol. 70, no. 12, pp. 2098–2111, 2021.

[18] A. M. Kaushik and H. Patel, "A systematic approach to achieving tight worst-case latency and high-performance under predictable cache coherence," in *2021 IEEE 27th Real-Time and Embedded Technology and Applications Symposium (RTAS)*, 2021, pp. 105–117.

[19] M. Hassan, "Discriminative coherence: Balancing performance and latency bounds in data-sharing multi-core real-time systems," in *32nd Euromicro Conference on Real-Time Systems (ECRTS 2020)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2020.

[20] S. Hessien and M. Hassan, "The best of all worlds: Improving predictability at the performance of conventional coherence with no protocol modifications," in *2020 IEEE Real-Time Systems Symposium (RTSS)*, 2020, pp. 218–230.

[21] M. Andreozzi, G. Gabrielli, B. Venu, and G. Travaglini, "Industrial challenge 2022: A high-performance real-time case study on arm," in *34th Euromicro Conference on Real-Time Systems (ECRTS 2022)*.

[22] N. Binkert, B. Beckmann, G. Black, S. K. Reinhardt, A. Saidi, A. Basu, J. Hestness, D. R. Hower, T. Krishna, S. Sardashti *et al.*, "The gem5 simulator," *ACM SIGARCH Computer Architecture News*, vol. 39, no. 2, pp. 1–7, 2011.

[23] "Gem5 ruby memory system." [Online]. Available: https://www.gem5.org/documentation/general_docs/ruby/

[24] "Amba atp engine repository." [Online]. Available: https://github.com/ecrtsorg/ATP-Engine

[25] S. Abdelhalim, D. Germchi, M. Hossam, R. Pellizzoni, and M. Hassan, "A tight holistic memory latency bound through1 coordinated management of memory resources," in *35th Euromicro Conference on Real-Time Systems (ECRTS 2023)*.

[26] M. Ferrera, A. Eudes, J. Moras, M. Sanfourche, and G. L. Besnerais, "Ov^{2}slam : A fully online and versatile visual slam for real-time applications," 2021.

[27] "Gem5 gcn3 gpu model." [Online]. Available: https://www.gem5.org/documentation/general_docs/gpu_models/GCN3

[28] N. Ruiz, E. Chong, and J. M. Rehg, "Fine-grained head pose estimation without keypoints," 2018.