Guest editorial: Special Issue on Predictable multi-core systems

Robert I. Davis¹

Published online: 24 April 2020

© Springer Science+Business Media, LLC, part of Springer Nature 2020

Multi-core hardware integrates multiple processing cores onto a single chip. To reduce costs and to improve performance in the average case, these cores typically share a number of hardware resources, including the interconnect, parts of the memory hierarchy (e.g., caches), and main memory. By contending for these shared hardware resources, tasks executing on one core can potentially interfere with tasks executing on another core, substantially increasing their execution times. Contention for shared hardware resources thus poses a significant challenge in the development of predictable hard real-time systems running on multi-core platforms.

Three concepts that are useful in a discussion of the real-time behaviour of multi-core systems are *Timing Composability*, *Timing Compositionality*, and *Timing Predictability*.

Timing Composability means that the timing properties derived for individual tasks studied separately (i.e., executing alone) still hold after their composition with other tasks, for example when they are run with the other tasks executing on the other cores. *Timing Composability* is highly valued from an industry perspective as it enables incremental development and verification. Different teams can develop and verify different sub-systems, in the knowledge that their timing behaviour will not change when integrated into the complete system.

Timing Compositionality means that the timing properties of interest, for example the Worst-Case Execution Time (WCET) of a task can be determined via a decomposition into constituent parts (for example, the worst-case processing time plus the worst-case delays waiting for the bus plus the worse-case latencies waiting for memory). *Timing Compositionality* is a key property required by much of the research on timing analysis and on integrated timing and schedulability analysis for multi-core platforms, and indeed by WCET analysis in general. Systems that are not timing compositional exhibit *timing anomalies* meaning that the (local) worst-case behaviour of some resource does not lead to the overall (global) worst-case execution time. Systems with timing anomalies are much

✉ Robert I. Davis
rob.davis@york.ac.uk

¹ University of York, York, UK

more difficult to analyse due to the fact that the problem of timing analysis cannot easily be decomposed, all possible resource behaviours and their interactions need to be considered together.

Timing Predictability has a number of different definitions in the research literature. One important measure is the ratio of the worst-case to the best-case execution time for a task (or for a sequential trace of instructions through the code of a task), or the ratio of the worst-case to the best-case latency when considering memory accesses.

This special issue contains three papers at the forefront of real-time systems research into predictable multi-core systems. Each of these papers appeared in preliminary form at the 39th IEEE Real-Time Systems Symposium (RTSS) in 2018. These preliminary papers received *Outstanding Paper Awards* identifying them as research of the highest quality. Overall, 166 papers were submitted to RTSS 2018. After a detailed review process, involving 51 Program Committee members and 4 reviews per paper, 37 papers were selected to appear at the conference; an acceptance rate of 22.3%. The papers that appear in this special issue include significant and comprehensive additional contributions to what can already be regarded as outstanding research.

The first paper is “*Design and Analysis of SIC: a Provably Timing-Predictable Pipelined Processor Core*” by Sebastian Hahn and Jan Reineke. In its preliminary form, this paper received the *Best Student Paper Award* at RTSS 2018.

This paper introduces the Strictly In-order Core (SIC), a timing-predictable pipelined processor core that is formally proven to support timing compositionality, and to be free of timing anomalies. The SIC core facilitates the use of compositional timing analysis, currently one of the most promising techniques for the analysis of the timing behaviour of tasks running on multi-cores. Compositional timing analysis first computes the “resource demand” of each task in terms of its use of each shared hardware resource. It then relies on the assumption that the response time of a task can be decomposed into contributions from each of the different shared hardware resources, which can be analysed separately much more efficiently.

The second paper is “*Work-Conserving Dynamic Time-Division Multiplexing for Multi-Criticality Systems*” by Farouk Hebbache, Florian Brandner, Mathieu Jan, and Laurent Pautet.

Time Division Multiplexing (TDM) bus arbitration policies are considered in many academic research papers, due to their advantages in terms of time composability and timing predictability. With TDM, the basic concept is that access requests to the bus are time partitioned at a fine level of granularity via a cyclic schedule of time slots. Each core may only access the bus during its dedicated time slot. Any requests that occur during time slots allocated to other cores have to wait until the start of the next slot allocated to their core.

The delays due to TDM are *timing composable*, since there is no change in behaviour or additional interference due to co-running tasks on other cores. In effect the interference is always there, whether other cores are using their time slots or not.

For this reason, TDM is rarely implemented in COTS multi-cores, as its non-work-conserving nature provides relatively poor average-case performance.

This paper introduces a set of dynamic TDM schemes, which are analysable since they are proven to preserve the guarantees afforded by TDM in the worst-case, while also providing performance that is competitive in terms of average-case behaviour. Instead of arbitrating at the level of TDM slots, these schemes operate at the granularity of clock cycles, exploiting slack time that accumulates from preceding requests. This enables the actual latencies of requests to be exploited, greatly improving memory utilization. The paper proposes an efficient hardware implementation of a dynamic TDM scheme considering implementation trade-offs and costs.

The third paper is “*Reduced Latency DRAM for Multi-core Safety-Critical Real-Time Systems*” by Mohamed Hassan. In its preliminary form, this paper received the *Best Paper Award* at RTSS 2018.

This paper makes a thorough study of the suitability of Double Data Rate Dynamic Random Access Memories (DDR-DRAMs) for use in real-time multi-core systems where timing predictability is a key requirement. The study focuses on the inherent variability in access latencies for DDRx (DDR2, DDR3, and DDR4) DRAM and hence exposes its limitations with respect to providing timing-predictable performance.

The overall variability in access times for state-of-the-art DDRx-DRAM controllers (i.e. worst-case divided by best-case latency) is in the range 2.5 to more than 10, with the lower values achieved with memory bank partitioning. Hence DDRx-DRAM may be unsuitable for real-time systems requiring highly predictable timing behaviour. Reduced Latency DRAM (RLDRAM) is proposed as an alternative solution. RLDRAM is an emerging DRAM protocol designed to provide significantly reduced worst-case latencies compared to DDRx-DRAM.

RLDRAM can make use of a non-multiplexed addressing mode that provides much more consistent access times. For example, the latency for 1600 MHz RLDRAM is between 19.5 and 28.5 ns (compared to 15 to 108 ns for DDR3-DRAM). Note the best case for RLDRAM is 30% longer than for DDR3-DRAM, but the worst-case is 3.8 times shorter. Overall variability is reduced from 7.2 to 1.46 times. The paper also considers a multiplexed addressing mode for RLDRAM, highlighting its advantages and disadvantages compared to the non-multiplexed mode. In addition, different burst lengths, and both read and write transactions are studied in terms of access latencies and their variability.

Together, these three papers form an excellent cross-section of state-of-the-art real-time systems research into predictable multi-core systems. Together, they cover the three most important elements of multi-core hardware, the processor cores, the bus, and the memory. They will undoubtedly provide a catalyst for further exciting research in this field.

Robert I. Davis
Guest Editor