Microprocessors and Microsystems 000 (2016) 1–16

Contents lists available at ScienceDirect

Microprocessors and Microsystems

journal homepage: www.elsevier.com/locate/micpro

Fitting processor architectures for measurement-based probabilistic timing analysis

Leonidas Kosmidis[a,b], Eduardo Quiñones[b], Jaume Abella[b,*], Tullio Vardanega[c], Carles Hernandez[b], Andrea Gianarro[d], Ian Broster[e], Francisco J. Cazorla[b,f]

[a] Universitat Politecnica de Catalunya, Barcelona, Spain
[b] Barcelona Supercomputing Center, Barcelona, Spain
[c] University of Padova, Padova, Italy
[d] Cobham Gaisler, Gotemburg, Sweden
[e] Rapita Systems Ltd, York, England
[f] Spanish National Research Council (IIIA-CSIC), Barcelona, Spain

ARTICLE INFO

Article history:
Received 3 January 2016
Revised 21 June 2016
Accepted 18 July 2016
Available online xxx

Keywords:
Worst-case execution time
Processor architecture
Cache memories
Probabilistic analysis
Time randomization

ABSTRACT

The pressing market demand for competitive performance/cost ratios compels Critical Real-Time Embedded Systems industry to employ feature-rich hardware. The ensuing rise in hardware complexity however makes worst-case execution time (WCET) analysis of software programs – which is often required, especially for programs at the highest levels of integrity – an even harder challenge. State-of-the-art WCET analysis techniques are hampered by the soaring cost and complexity of obtaining accurate knowledge of the internal operation of advanced processors and the difficulty of relating data obtained from measurement observations with reliable worst-case behaviour. This frustrating conundrum calls for novel solutions, with low intrusiveness on development practice. Measurement-Based Probabilistic Timing Analysis (MBPTA) techniques offer the opportunity to simultaneously reduce the cost of acquiring the knowledge needed for computing reliable WCET bounds and gain increased confidence in the representativeness of measurement observations. This paper describes the changes required in the design of several high-performance features – massively used in modern processors – to meet MBPTA requirements.

1. Introduction

The market for Critical Real-Time Embedded Systems (CRTES), which includes the automotive and avionics sectors, is experiencing an unprecedented growth [1]. While crucial to keeping competitive advantage, the inclusion of increasingly sophisticated value-added functions, such as for example Advanced Driver Assistance Systems, causes CRTES makers to continually seek higher guaranteed computational performance while striving to contain cost and power budget. This goal can only realistically be achieved by adding complex and powerful hardware accelerator features such as caches or multicore designs.[1]

However, the use of aggressive performance-enhancing hardware features may highly complicate the computation of reliable and tight timing bounds.[2] Worst-Case Execution Time (WCET) analysis is an integral step of verification for real-time systems in general, and for CRTES in particular. One common use of WCET bounds is for schedulability analysis to ascertain whether application tasks can complete within their assigned deadlines under all conditions.

Numerous techniques exist for performing WCET analysis, ranging from measurement-based to static analysis, via hybrid variants that use elements of both [2]. Measurement-based techniques rely on user's ability to design stressful tests in which the application under test is run in conditions similar to the worst ones that can arise during operation. Static timing analysis is challenged by the difficulty to model accurately the timing of complex hardware designs, and also by the increasing amount of information needed to feed the models to estimate the WCET. Finally, hybrid approaches alleviate some of the problems of those techniques to handle com-

* Corresponding author. Fax: +34 934137721.
E-mail address: jaume.abella@bsc.es (J. Abella).

[1] This trend deflects from prior practice in CRTES, where processors used to be in-order and cacheless, to simplify verification of timing behaviour.

[2] In the context of timing analysis, a reliable bound is a bound that can be supported by strong arguments and proofs.

http://dx.doi.org/10.1016/j.micpro.2016.07.014
0141-9331/

plex hardware, but hybrid approaches are subject to similar limitations.

The availability of more powerful hardware and the quest for more functional value per unit of product also prompt CRTES industry to consider adopting mixed-criticality design solutions for their systems. From the timing perspective, which is the focus of this paper, the challenge with mixed-criticality systems lays in the need for solutions to ensure strict temporal isolation between programs assigned to different criticality levels, so that their behaviour can be deemed composable in the time dimension.[3] In the absence of effective means to abate the pessimism of WCET analysis, however, mixed-criticality solutions that achieve time isolation by fencing budget allowances, risks incurring massive overprovisioning, which defeats the purpose of combining systems together.

Probabilistic techniques may greatly aid on all of those fronts. In particular, with Measurement-Based Probabilistic Timing Analysis (MBPTA) methods [3–6], the execution time of the application can be accurately modelled – at some level of execution granularity – by a probability distribution. MBPTA seeks to determine WCET estimates for arbitrarily low probabilities of exceedance, termed probabilistic WCET or pWCET. As a consequence, there is some residual risk (in the form of an exceedance probability) beyond which it cannot be proven that a pWCET bound cannot be exceeded. However, this residual risk is upper bounded with a given probability, which can be determined at a level low enough to suit the needs of system design in the application domain. For example, the residual risk can stay in the region of 10^{-9} per hour of operation, largely below the acceptable probability of failure in certified systems.

Under MBPTA, at a given granularity of execution, the response time of every individual execution component at that level (e.g., an instruction) is assigned a distinct probability of occurrence. This trait – which shall not be confused with the probability of that component *being* executed in a run of the program – is described by a probabilistic Execution Time Profile (ETP), expressed by the pair: <timing vector; probability vector>. The timing vector in the ETP enumerates all its possible response times. For each response time in the timing vector, the probability vector lists the probability of occurrence of that response time in an instance of execution. Hence, for execution component $\mathcal{C}_i$ we have $ETP(\mathcal{C}_i) = \langle \vec{t_i}, \vec{p_i} \rangle$ where $\vec{t_i} = (t_i^1, t_i^2, \ldots, t_i^{N_i})$ and $\vec{p_i} = (p_i^1, p_i^2, \ldots, p_i^{N_i})$, with $\sum_{j=1}^{N_i} p_i^{N_j} = 1$. At the program level, MBPTA requires that the ETP for the program exercised during analysis matches or upper-bounds program's ETP during operation.

The processor architecture is instrumental in ensuring that individual instructions have an associated ETP. As this guarantee in turn is a crucial enabler to a sound and effective application of MBPTA, the processor architecture is the level of execution granularity on which we focus in this work.

Contribution. Within the context of the FP7 PROXIMA project [7] we describe the architecture features that a processor should possess to be amenable by construction to the use of MBPTA. We term this quality *MBPTA compliance*. In presenting our case, we offer insight on the costs that may be incurred in actual implementation of a MBPTA-compliant processor. To that end, we categorise processor resources according to their timing behaviour and detail how they should be designed for use in a MBPTA-compliant processor. Without loss of generality, we consider the inner operation of the processor to employ a number of passive resources (e.g., caches, buffers, buses, etc.). We assume each processor instruction to use some of those resources in a given order, whether in sequence or in parallel. We design processor resources so that each of them can be assigned a given ETP. To achieve this for all resources, we use *time randomisation* in *some*, actually very few, of them. Resources that are not time randomised must be assigned a local upper bound to their response time that can be safely composed. We assume a time anomaly free baseline architecture.

The remainder of this paper is organised as follows. Section 2 introduces PROXIMA and contextualises this work. Section 3 presents the requirements that MBPTA places on processor hardware. Section 4 classifies hardware resources in a taxonomy specifically related with MBPTA. Section 5 presents software-only solutions that could be applied to make commercial-off-the-shelf processor hardware fit for MBPTA. Section 6 presents a demonstrative implementation of a processor architecture, purposely designed for compliance with MBPTA. Section 7 surveys related work. Section 8 draws some conclusions and outlines the future of this line of work.

2. Context within PROXIMA

This work has been performed within the scope of PROXIMA [7], an Integrated Project (IP) of the Seventh framework programme for research and technological development (FP7). PROXIMA objectives include providing a complete toolchain enabling low-cost timing verification for systems based on multicore and manycore processors implementing critical real-time functionalities. In particular, PROXIMA toolchain includes the following main elements:

- *Hardware and software platforms amenable for MBPTA*. One of the key elements of the toolchain is a hardware platform providing the timing properties required by MBPTA to facilitate obtaining reliable and tight pWCET estimates. This hardware platform has been implemented in a FPGA prototype used in the Space domain. Alternative software-only solutions have been developed to enable MBPTA on top of commercial off-the-shelf (COTS) processors that include a non-MBPTA-compliant version of the Space prototype, an Infineon AURIX T277 and a Freescale P4080 processors. MBPTA compliance in future manycore processors has also been investigated by means of architectural simulators.
- *MBPTA-compliant real-time operating systems (RTOS)*. The RTOS needs to be enhanced with features so that its contribution to the execution time of the analysed tasks is made constant, and hence, time-composable, and its impact on the hardware and software state is neutral w.r.t. the properties needed to attain MBPTA compliance, thus being transparent for the timing analysis process. RTOS features have been implemented as part of PikeOS, RTEMS-SMP, ERIKA and some research-oriented RTOS.
- *Timing analysis tools*. Appropriate methods for the estimation of pWCET are required to account for the timing behaviour of the underlying hardware/software platform. They must be compatible with the tracing methods in place, and capable of providing pWCET estimates that hold valid in front of the different sources of execution time variation that can be exercised during operation such as hardware/software initial state, input values, execution path traversals, etc. Some of these methods have been implemented as part of RapiTime commercial toolchain whereas others will remain as standalone tools.

These elements have been implemented by a set of industrial and academic partners including hardware, RTOS and timing analysis tool vendors and related research institutions. Evaluation is performed on a number of case studies from the avionics, space,

[3] Time composability is had when the timing behaviour of an individual software component does not change in the face of composition when the system is integrated, and so, the timing analysis performed in isolation remains valid at system integration.

Please cite this article as: L. Kosmidis et al., Fitting processor architectures for measurement-based probabilistic timing analysis, Microprocessors and Microsystems (2016), http://dx.doi.org/10.1016/j.micpro.2016.07.014