Universidade Federal de Santa Catarina
Atividades de Extensão - Res. Nº 03/CUn/09
Formulário de Tramitação e Registro

Situação:**Relatório Final em Aprovação**
Protocolo nº: **2014.3239**

Título da Atividade:	Revisão de artigo para o IEEE Real-Time Systems Symposium (RTSS 2014)
Objetivos e metodologia:	Revisão do artigo Automatic Cache Partitioning and Time-Triggered Scheduling for Real-time MPSoCs para o IEEE Real-Time Systems Symposium (RTSS 2014).
Palavras chave:	Revisão de artigo
Entidade parceira:	IEEE
Município / Estado:	Joinville / SC
Forma de Extensão:	PRESTAÇÃO DE SERVIÇOS
Complemento da Forma de Extensão:	Assessoria
Período de realização:	06/19/2014 a 06/19/2014
Carga horária total da atividade:	4 horas
Número de pessoas atingidas por esta atividade:	200
A atividade receberá algum aporte financeiro?:	Não

Envolvidos nesta atividade de extensão

Coordenador	
Nro do SIAPE:	2056505
Nome do Coordenador:	Giovani Gracioli
CPF do Coordenador:	1083806025
Departamento:	CAMPUS DE JOINVILLE
Centro:	CAMPUS DE JOINVILLE
Regime de trabalho:	DE
Fone de contato:	(48) 9118-3244
E-mail:	giovanig@gmail.com
Carga horária na atividade:	Não entra no PAD
Número de Horas TOTAIS:	4 horas
Receberá remuneração nesta atividade de extensão?	Não

Outros prof. ou servidores da UFSC envolvidos?	Não
Alunos da UFSC envolvidos?	Não

Pessoas externas à UFSC envolvidas?	Não

Outras Considerações

Revisão realizada:

Summary Ranking

Please evaluate the submission according to the criteria below.

Evaluation Category Enter Your Score
Originality (1-6)
3: Little new material

Relevance of the Contributions (1-6)
2: Not interesting

Technical Correctness (1-6)
2: There are problems whose impact needs to be assessed

Presentation (1-6)
3: Borderline

Relevance to RTSS (1-6)
5: Relevant

Overall Score (1-6)
3: Comparable to weaker papers I have seen at conferences and not liked

Reviewer Confidence
-Expert: I am an expert in the field

Nomination for best paper
Please indicate if you want to nominate this paper for the best paper award

No

Detailed Comments

Please supply detailed comments to back up your rankings. These comments will be forwarded to the authors of the paper. The comments will help the committee decide the outcome of the paper, and will help justify this decision for the authors. Moreover, if the paper is accepted, the comments should guide the authors in making revisions for a final manuscript. Hence, the more detailed you make your comments, the more useful your review will be - both for the committee and for the authors.

Enter comments here:

* Summary
The paper proposes a framework for providing predictability for real-time applications running on top of MPSoCs with shared cache. Predictability is achieved by way-based cache partitioning and partitioned time-triggered scheduling. The framework takes as input the platform, mapping (tasks assigned to cores), and task specifications and generates as output a time-triggered scheduling and the assignment of ways to tasks to decrease the number of miss in the shared cache.

Way-based cache partitioning is performed by a hardware-specific implementation in a FPGA, which was named "Reconfigurable Cache Architecture". The paper describes the main parts that form this cache architecture and evaluate it in terms of speed and FPGA area (cache size and processor frequency), cache correctness, timing predictability, and runtime performance. Although interesting, the paper lacks a better motivation and position with respect to related works. Furthermore, the timing predictability and runtime performance experiments are not completely fair, which result in a few relevant observations regarding timing guarantees. These two main drawbacks are the reason for the given scores.

* Pros
It is always nice to see implementation-oriented paper

* Cons
The paper lacks a better motivation, mainly in the introduction. Related work is not complete and is not well organized. Experiments are not fair. The paper needs a careful english revision.

Detailed comments are given below.

- Introduction

In the introduction, the main motivation for the work is given in the third paragraph. The authors state that the main problem for SW cache partitioning techniques based on page coloring in the context of real-time systems is the overhead of page recoloring. For supporting this affirmation, the authors cite the work [19], which is not a real-time work. The main point here, in my point of view, is that hard real-time (HRT) systems usually do not require page recoloring. All tasks, including their cache partitioning, must be defined off-line, i.e., tasks and their partitions are static. Otherwise, obviously the time to perform a page recoloring would increase their WCETs due to the copy of data from the old page to the new page. Then, the authors state that "software cache partitioning approach can only work well when recoloring is performed infrequently [37]". Again, this motivation seems not be correct in the context of hard real-time systems and must to be clarify. Moreover, [37] is not proposed for real-time systems either. In my point of view, page recoloring could be performed in software-based cache partitioning as well. For example, the utilization slack could be used for that purpose. Of course, the overhead is high and would have to be carefully analyzed. In summary, this motivation must be clarify and better references must be used.

In the fourth paragraph of the introduction, the authors discuss the relation between scheduling and cache allocation. However, again, I am not sure how relevant this issue is for HRT. If cache partitions are assigned to tasks off-line, the maximum cache region given to a task does not depend on the scheduling. It will only impact intra-task interference (i.e. self-evictions). If page recoloring is performed, then this is an issue at the cost of performance, but the text should make it clearer. In the fifth paragraph, the authors say that "during runtime, the cache is reconfigured by the scheduler according to offline computed configurations", but how this is done is not clear in the other paper's sections. Also, if the assignment of partitions to tasks is performed offline, why software-based cache partitioning could not be used? The overhead is higher, but could be integrated in the analysis. To better motivate and discuss this, a comparison in terms of overhead between page recoloring and the proposed cache reconfiguration would certainly help.

Finally, it would be nice to give some hints of the achieved results in the last contribution item.

- Related Work

The authors state that few works in the literature have been done in the context of real-time multicore system. This is true if only way-based cache partitioning is considered. If software-based cache partitioning is considered, then this is not true. The related work section could be better organized. First, the advantage and disadvantage of software and way-based techniques are not discussed. Way-based partitioning has two main advantages. First, it requires limited changes to the set-associative cache organization which do not have a dramatic impact on the overall structure. Second, this partitioning scheme keeps requests for the different compartments isolated from each other. Thus, there is no contention for the cache ways at the cores. However, the key drawback with this approach is that the number of partitions, as well as the granularity of the allocations, is limited by the associativity of the cache. Increasing the associativity of a cache is not always feasible or efficient since a higher-order associativity determines an increase in the cache access time and tag storage space. Also, it is usually not supported by COTS multicore processors.

In summary, cache partitioning works could be divided in index- and way-based cache partitioning techniques and further classified in software and hardware. Advantages and disadvantages of each classification must be discussed.

- Background Section B

The authors state that a WCET estimation technique is used to determine the WCET and will be discussed in Section VII. Section VII defines a measurement-based approach, but does not explain how the measure was performed. How many executions? Which platform? Cache size? How many ways for each task? Which RTOS? Are tasks executing alone in the system? How many tasks? Although there is a reference there, many important details are not given.

- Section V

The equation must be better explained. To ease readability, it is desirable to add text explaining each equation.

- Section VI-A

"Note that inter-core cache interference still exists although software on each core runs ON different regions of the same off-chip memory" => explain how this is possible. If tasks are individual ways, the only possible interferences are in the memory bus and in the shared cache bus. A discussion about these issues is needed.

- Section VIII-A
What is the size of a way? Is there an RTOS running? If so, which one? Are the benchmark applications multi-threaded? How is the assignment of ways considering the multiple threads? How self-evictions are accounted (refer to three

comments above)?

- Section VIII-D
It is compared two different caches: the proposed one with another one with a single port without cache partitioning. In my point of view, the comparison is not fair. It should compare the same cache architecture with and without cache partitioning. Moreover, if the WCET is estimated using cache partitioning and running the tasks experimentally, would we expected a different value in Figure 10? I mean, if the WCET was estimated by running tasks with individual partitions, running them again would result in different WCET? Maybe this confusion is created by the lack of information regarding how the WCET is estimated.

- Section VIII-E
The authors compared the proposed approach with the one proposed in [17]. First of all, the work in [17] is not discussed in details, which makes it difficult to follow the evaluation. What EQUAL means? The explanation is short. What exactly was implemented from [17]? A discussion of the cache reservation is essential. The work in [17] is designed for a COTS multicore processor with a cache coherence protocol. Cache interference is also derived from the coherence protocol. In the proposed work, there is not cache coherence protocol. Without these discussions, it is not possible to follow the results in Figure 10. Again, the comparison seems not to be fair.

Also, a shared cache with only 256KB seems not to be relevant if compared with current multicore processors, which have much larger shared caches.

* Minor points
- Introduction
Par. 1: move citations [14][17][22][23] to the end of the sentence. Also, modify the citation style to [14,17,22,23]
Par. 1: "... series [3] and openSPARC series [31] all.." => remove all
Par. 2: move citations [11][14] to the end of the sentence
Par. 2: ".. L2 cache space" => "L2 cache lines"
Par. 2: ".. will cause the increase" => ".. will cause an increase"
Par. 2: ".. to a corresponding decrease in the performance" => ".. to a corresponding decrease in performance"
Par. 2: "Being aware of this problem, this paper studies the problem" => problem is repeated
Use always page coloring instead of page-coloring
FPGA is used in the first time without defining the acronym

- Related work
Par. 1: LLC is used without definition
Par. 2: ".. cache by sets in OS-level" => at OS-level

- Background
"as an periodic task set" => "as a task set"

- Figures must be place after their citations. Examples: Figures 2, 4, 8, and 9, and Listing 1.
- Table IV is placed before Table III.
- Figure 10 is placed after Figure 11.
- Spelling mistake in references [3] and [33].
- Task switch => context switch

Confidential Comments for Committee

You may wish to withhold some comments from the authors, and include them
solely for the committee's internal use. For example, you may want to express
a very strong (negative) opinion on the paper, which might offend the authors
in some way. Or, perhaps you wish to write something which would expose your
identity to the authors. If you wish to share comments of this nature with the
committee, this is the place to put them.

Parecer do Departamento:	Aprovado
Data de aprovação:	07/02/2014 - Ad-referendum