IEEE Industrial Electronics Society

IECON 2011 Main Page

IEEE IECON11 Manuscript Submission System

Comments from Reviewers Regarding Your Manuscript

MD-011983

Manuscript Information

View the submitted manuscript

Contact Person

Name (first, last):	Mr. Tiago Rogério Mück
Affiliation:	Federal University of Santa Catarina, Brazil
E-mail:	tiago@lisha.ufsc.br
Address:	Mr. Tiago Rogério Mück UFSC/CTC/LISHA PO Box 476 88040-900 Florianópolis - SC Brazil
Telephone:	+55 48 3721-9516
Facsimile:	+55 48 3721-9516

Manuscript

Title:	Run-time Scratch-pad Memory Management for Embedded Systems
Authors:	Tiago Rogério Mück, Antônio Augusto Fröhlich
Technical Track:	SS Industrial Applications of FPGAs and Embedded Systems
Main Keywords:	II-C0. Intelligent Components and System Architectures SA-B7. Low Power Electronic Circuits for

Author Database

1st author:	Mr. Tiago Rogério Mück, Federal University of Santa Catarina, Brazil <tiago@lisha.ufsc.br>
2nd author:	Prof. Antônio Augusto Fröhlich, Federal University of Santa Catarina, Brazil <guto@lisha.ufsc.bı

Technical Details

Transaction Number: MD-011983

Data Type:	**pdf**
Date:	**1305386574**
Origin:	**spectrum.lisha.ufsc.br/150.162.62.60**
Status:	**OK - submission completed**

Feedback from the Reviewers

- C. Clarity of presentation:
```
    English grammar and spelling are proper ------------------------------------------ [3 - I agree]
    Mathematical symbols and equations are easy to understand ------------------------ [4 - I strongly agree]
    Figures and tables are well constructed and informative -------------------------- [4 - I strongly agree]
    The paper is well organized ------------------------------------------------------ [4 - I strongly agree]
    Considering the issues above, the paper is readable ------------------------------ [3 - I agree]
```
  T. Technical innovation and relevance
```
    The authors cite other relevant publications ------------------------------------- [4 - I strongly agree]
    Authors describe relevance of work to the research field ------------------------- [3 - I agree]
    The authors apply sound technical approaches ------------------------------------- [4 - I strongly agree]
    New ideas are convincingly and logically described ------------------------------- [3 - I agree]
    Results are convincing ----------------------------------------------------------- [3 - I agree]
    Considering the issues above, this work should be presented  --------------------- [3 - I agree]
```
  Comments:
  This paper presents a runtime memory management approach for Scratch-pad memories (SPMs). The approach uses annotations in the code to guide the allocation of data to the SPM or to the main memory in the system. The authors use an embedded system implemented in an FPGA to evaluate the approach.

  I liked this paper. However I would like to see more details about how the information regarding annotations is represented in the executable code. Do you embed them with the assembly instructions of the program? Do you include them as part of the new and delete operators? Do you have a compiler aware of the annotations or do you need to modify assembly code to insert the annotations?

  I would like to see in the paper the latencies achieved for the different approaches.

  This paper is worth to be presented at the conference.

- C. Clarity of presentation:
```
    English grammar and spelling are proper ------------------------------------------ [4 - I strongly agree]
    Mathematical symbols and equations are easy to understand ------------------------ [2 - I am neutral]
    Figures and tables are well constructed and informative -------------------------- [3 - I agree]
    The paper is well organized ------------------------------------------------------ [2 - I am neutral]
    Considering the issues above, the paper is readable ------------------------------ [2 - I am neutral]
```
  T. Technical innovation and relevance
```
    The authors cite other relevant publications ------------------------------------- [3 - I agree]
    Authors describe relevance of work to the research field ------------------------- [3 - I agree]
    The authors apply sound technical approaches ------------------------------------- [4 - I strongly agree]
    New ideas are convincingly and logically described ------------------------------- [1 - I disagree]
    Results are convincing ----------------------------------------------------------- [2 - I am neutral]
    Considering the issues above, this work should be presented  --------------------- [1 - I disagree]
```
  Comments:
  This paper suggests a way to utilize SPM dynamically by some help from the operating system. For this a new memory region for keeping heap memory of applications is deployed over SPM, and some extra annotations for “new” keyword in C++ is used. So we can control dynamic memory allocation. Although the presented idea seems simple but interesting, the paper would still need to be revised further for the

more contribution to this field.

There are several major issues with this manuscript.

The many paragraphs of this manuscript seem to come from the author's prior work- Tiago Rogério Mück and Antônio Augusto Fröhlich, "A Run-time Memory Management Approach for Scratch-pad-based Embedded Systems", In: Proceedings of the 15th IEEE International Conference on Emerging Techonologies and Factory Automation, pages 1-4, Bilbao, Spain, 2010. Although experiments and result section of this paper contains more new information from the prior work, it still explains the pretty much same contents with some rephrasing.

This paper is sometime ambiguous since required information is not presented, and the authors locate their work to wrong category of this field. The authors claim that the suggested scheme works at OS-level without compiler support. But the explained scheme is seen as a compiler based work since “new” keyword of C++ is annotated with the allocation hint, and it is handled by system library. Furthermore there is no explanation how the compiler processes the hints annotated in the C++ source code.

Abstract

"their higher efficiency"
This would be controversial if in what senses SPM shows higher efficiency than cache memory is not provided.

"The operating system memory manager takes annotations inserted into the code"
This would incur some confusion due to the two points. It is not clear how the operating system actually takes the annotation inserted into code, and which kind of code is modified to annotate the hints for allocation. Of course these are answered in later sections, but this would set readers with confusion when they start reading this manuscript.

I. Introduction

This section starts with good briefings about the motivation but is pretty confusing with the paper's contribution and novelty.

"In this work we propose a run-time ~ profiling or hardware support"
This is hardly agreeable since authors seem to upgrade C++ compiler and library to support the proposed scheme. There is some confusion on this through the whole paper.

The paper covers only dynamic data objects but this is not discovered in the introduction too. It would be helpful to state more clearly and accurately where the proposed work is positioned in multi dimensional space of approaching methods and declaring which kind of data objects are covered by this work.

"Several software allocation approaches that ~ have a major drawback"
Reference would be required for this part. Moreover, there are many works which statically choose instructions and data being allocated into SPM rather than by profiling.

Much research works have performed to allocate data variables on the worst case to SPM as well. Hence just arguing the drawback neither tells the drawback is general in this field and justifies the benefit of the proposed work.

"applications were the memory"
This seems a typo of "applications where the memory".

II. Related Works

This section mentioned about some of major works in this field but the terms used are not quite clear.

Compile-time techniques are generally called as static techniques, while run-time techniques as dynamic techniques. But the authors already use the terms (static and dynamic) for telling how to load the data into SPM so that similar terms having same meanings are used confusingly such as “compile-time technique, post-compile technique and run-time technique”. Combination of two different domains would make more sense such as “statically decided dynamic methods” as [15] did.

Moreover, it is not quite sure [6], [7], and [11] are static approach. For example, in case of [7], they convert automatic variables into static variables but that is just implementation method for the paper presentation. In that paper the authors mentioned about this. They explained about needs for multiple stack frames and corresponding cost to cover with automatic variables, which are statically decided to move to SPM.

"All of the previous techniques handled only code and/or static data"
Some of them consider automatic (stack) data as well as static data and code.

III. RUN-TIME SPM MANAGEMENT

It might be mentioned how to implement the framework for the C++ language. The authors already mentioned that no compiler support is required for the proposed scheme. If then, the next question by readers would be how the keyword following "new" is processed during compiling.

A. Memory management on EPOS
The more detailed description about the "kernel", "built-in" and "library" would be helpful to figure out the author's idea. It is not easy to connect the first paragraph to the topic of this manuscript.

B. The allocation framework
"ALLOC_HIGH, ALLOC_LOW, and ALLOC_NORMAL" seems typos. In the following paragraphs, "ALLOC_P_(something)" is used.

"ALLOC_P_NORMAL" seems balanced allocation. Is there particular intention on this?
Why does this case become the default allocation?

IV. EVALUATION
A. Benchmarks

```
Showing the total footprint in Table I would help to figure
out the benchmarks. The total sizes of "Dijkstra", "SHA"
and "FFT" of "Original benchmarks" are different from those
of "Modified benchmarks". Why does nothing change in "Susan-
Smoothing" and "Susan-Corners" after the modification?

"In our approach, only data which is dynamic allocated at
run-time can be handed out to the SPM. However, this
limitation can be softened by declaring global and stack
data as heap data."

This might turn to be a serious weak point of this idea.

The first question is how to cover overhead occurring when
we convert global and stack data to dynamic data. To
allocate data variables dynamically, we have to pay some
cost in terms of execution time.

As the second question, these sentences may not make sense.
In case of global data, they will stay in memory (wherever)
until the end of the program. Hence their memory space can
not be shared by other data variables at all. Then their
location can be controlled by a linker script rather than
converting them to dynamic type does not make sense
because. In case of stack (automatic) variables, their life
span is same as period of a procedure. Is it worth to
convert static variables to dynamic ones to support only
the limited life span of them?

C. Discussion
Without explanation about implementation and experiments
setup of "Dominguez et al [5]" case Figure 8 compares
average energy reduction. This does not seem fair
comparison.
```

Accepted or Rejected?

- The manuscript is on the list of accepted manuscripts.
- The manuscript is not on the list of rejected manuscripts.
- The manuscript is not on the list of withdrawn manuscripts.
- The manuscript is not on the list of registered manuscripts.

Done Viewing