Reply letter to IJES-130703

“Multi-Core Model Checking and Maximum Satisfiability Applied to Hardware-Software Partitioning”

August 28, 2016

Dear Dr. Chen Liu,

The authors would like to thank the anonymous reviewers for their valuable and constructive comments, suggestions and discussions, which helped us improve the quality of our manuscript.

The following pages list our responses to each of the comments. In particular, the changes in the revised version of our manuscript are highlighted in blue color along the original text. Our responses to all comments, which were raised by the reviewers, are also given in the sequel using the blue color.

We hope that the revisions in the manuscript and our responses will be sufficient to make our manuscript suitable for publication in the International Journal of Embedded Systems (IJES).

Responses to the comments of Reviewer #A

[A-1] I think that the abstract should be reviewed because it does not provide a clear view of the problem, its solution, methodologies, contributions, etc.

> **Response:** We appreciated the advice. The abstract was rewritten.

[A-2] According to the bibliography style, the list of references should follow an alphabetical order (e.g., by authors) and then year, etc., in order to be able to find the reference easily.

> **Response:** We thank the reviewer for point out this inconsistency. We have corrected the references according to the (recommended) bibliography style of IJES.

[A-3] What is the meaning of "Aeq" and "beq" in Eq. 1?

> **Response:** We implemented as suggested. Actually, *Aeq* and *beq* are matrices that describe the linear equality constraints.

[A-4] In page 4, "Paci et al." (dot) and "informs the solver" (to).

> **Response:** We implemented as suggested.

[A-5] In page 5, the text of first sentence in Section 3.2 should be re-written: "The inputs ... are: a .... , is necessary.".

> **Response:** We implemented as suggested.

[A-6] In page 5, "Based on ..., (?) is called ..." (please, re-write the sentence).

> **Response:** We have rewritten the respective sentence as suggested by the reviewer.

[A-7] In page 5, "Three different optimization and decision problems", what are these problems? Is the problem addressed among these? why does you focus on that specific problem?

> **Response:** We have written the following sentence to answer those questions: "In particular, different optimization (and decision) problems can be defined for partitioning HW-SW, as described by Arato *et al*. In this paper, however, the focus is on systems with hard real-time constraints: *S0* is given (initial cost of software), *i.e.*, the goal is to find a P HW-SW partitioning so that *Sp* <= *S0* and *Hp* is minimal, which is thus related. Consequently, the constraints can be reformulated based on Equations (1) and (3) as…"

[A-8] In eq. 4, what is "E" ? does "x" represent a vector or a single number? what is "s"? what is "c"? (please, explain the variables, parameters, operators, etc. of the eq.).

Response: We thank the reviewer for pointing out this inconsistency in the paper. We wrote the following sentence in Section 3.2: “In particular, *s* and *c* are the vectors representing the cost functions, *E* is the transposed incidence matrix of *G* (indicating which edges cross the boundary between the contexts of hardware and software), and *x* represents the decision variable (a binary vector indicating the partition: 1 if the node is realized in hardware and 0 if the node is realized in software).”

[A-9] The author/s should explain how the constraint are addressed in GA. What is the meaning of "slack variables"?. Also, they should explain how the solutions are represented, what kind of operators (crossover, mutation, parent selection and offspring selection) used, etc. What is the meaning of "Elite" counter that you have modified from 2 to 50? Are 75 generations enough to ensure the convergence of the GA? The author/s should show the objective function. A similar analysis and/or explanation should be performed for ILP. I think that the author/s should give a more detailed explanation of both techniques because the authors executed GA and ILP (not just used the results given in another paper), and besides their results are strongly compared and mentioned in Section 5.3 and Section 7.

Response: We agree with the reviewer that something is missing when we address the ILP and GA techniques. However, we must take into account, that in previous work issued by the authors, all parameters were presented and discussed. Therefore, the focus of this work was to move forward, not only with just a few improvements related to GA, but also with new ESBMC approaches to prove that it is an interesting tool to solve the HW/SW partitioning problem. Based on the reviewer’s comments, however, we have included explanations about some key issues, in order to make the understanding easier for the reader (see Section 4.1., Equations 4 and 5).

[A-10] In page 5 (right column), I think that the text "The ESBMC ... (2015)" is unclear and it should be rewritten. Beside, why do you use a "transposed incidence matrix" and "identity matrix"? (maybe is it related to eq. 4??).

Response: We have rewritten this sentence as follows: “The ESBMC algorithm starts with the declarations of hardware, software, and communication costs. *S0* must also be defined, as the transposed incidence matrix used in Eq (4) and the identity matrix (necessary to work with matrices), as typically done in MATLAB.

[A-11] In fig. 5, what is the meaning of "/"? ("Populate Xi with nondet divided by test values is ?."). what is meaning of "Hmax"? what are the values for "i"? Is "Xi" different from "x"? "isued"?

Response: We rewrote the sentences before and after Fig. 5 in Section 4.2, in order to improve the explanation of the algorithm details.

[A-12] In page 5 (right column), "inform ESBMC" (without "to"). I think the author/s should rewrite avoiding the word "inform".

> **Response:** We have replaced “inform” by “instruct” as follows: “It is possible to instruct ESBMC with which type of values the variables must be tested.”

[A-13] In page 6 (left column), please, rewrite the last sentence in Section 4.2: "In the ... ? is not ...".

> **Response:** We rewritten this sentence as follows: “In the ESBMC algorithm, there is no need for adding slack variables in Eq. (4), which reduces the number of variables to be solved if compared to ILP and GA.”

[A-14] In ESBMC-SS, how are the decision variables selected in each instance? Could a solution be evaluated several times in a same iteration or/and in different iterations?

> **Response:** The ESBMC-SS is not responsible for selecting the value of the decision variables. Actually, these values are selected by the SMT Solver to solve the equation produced by ESBMC. Additionally, the set of values are symbolically evaluated once for each interaction (*i.e.*, non-deterministic values + tipH). We have included the following text in Section 4.2 to clarify those questions. “As a result, the Boolean value that is assigned to each decision variable x_i is actually selected by the SMT Solver, during its solving phase, which checks once all possible combinations to yield a feasible solution, e.g., by handling the terms in the given background theory using a decision procedure Moura and Bjorner (2008); Brummayer and Biere (2009). If this is achieved, the ASSUME directive ensures the compliance of the constraint A.x <= b.”

[A-15] In ESBMC-PS, how are the decision variables selected by each worker? Could two workers evaluate the same possible solution? Are the "fork" and "join" of the process employed a single time? Do the workers have communication or information interchange? Besides, the author/s said "the processor does not remain idle for a longer period and thus there is almost no optimization", then what is the contribution of this proposal?

> **Response:** Similar to ESBMC-SS, the decision variables are selected by the SMT solver to check the verification conditions produced by ESBMC. Additionally, each worker is instantiated using a different value for *TipH*, which checks once for a possible solution. There is no communication among workers; however, each worker is responsible for sending a message to the controller if a violation is found for the TipH value during its execution. The ESBMC-SS instantiates a block of threads based on the number of available cores in the CPU. A new block will be executed only if all threads in the previous execution were concluded. In contrast to ESBMC-SS, ESBMC-PS does not wait for all threads in a block, *i.e.*, after a

thread is finished, a new one is instantiated with a different value of TipH. Based on the reviewer's comments, we have improved the text to properly describe the contribution of this proposal.

[A-16] In ESBMC-PB, could two workers evaluate the same possible solution? Do the workers interchange information?

> **Response:** In ESBMC-PB, it is not possible for two workers to evaluate the same (possible) solution because when a worker invokes the method *controller.GetNextStep()*, which is synchronized, and returns a possible *TipH* value, this value is removed from the list that contains the available candidates. Additionally, the workers do not interchange information among them, but they are responsible for sending a message to the controller if a given violation is found. We have described all those considerations in Section 4.3.3.

[A-17] In ESBMC-vZ, where is "CC" in fig. 12?

> **Response:** We have fixed the text. The correct word is "CMC" instead of "CC", where "CC" represents the communication cost (see Section 4.4).

[A-18] In Table 2, why is it showed the error rate for GA? (why is not it showed value Hp?).

> **Response:** We have implemented as suggested. In particular, we have included a new line in Table 2 to show the error rate for the GA results.

[A-19] Are there a special reason to select 3600 sec. as verification time limits?

> **Response:** We included an explanation in the paper (at the end of Section 5.1) as follows: "The TO was defined based on previous empirical tests, where a larger TO (e.g., 5000 seconds) did not produce substantial differences in the experimental results.

[A-20] In Ref. [1], the authors solved the benchmark "Mars" with GA and ILP, however, in this paper, the proposals cannot reach a solution (TO or MO). Maybe, the author/s should analyze this issue and/or extend the verification time limit.

> **Response:** In Ref. [1] the authors do not describe the platform or even the programming language that they used to implement the proposed algorithms. However, based on previous tests with the same benchmarks, done in Ref. [3], the results were improved. This is the reason by which we created a new standard of testing environment and repeated the execution of all benchmarks. As described in answer [A-19], a larger verification time limit does not necessarily lead the present approaches to the HW/SW partitioning solution.

[A-21] In my opinion, Section 6 should be before Section 2.

> **Response:** Section 6 was moved to Section 1 (Introduction) and rewritten to avoid redundancy, following the IJES guidelines.

[A-22] In page 9 (right column), "It [is] worth mentioning".

> **Response:** We implemented as suggested.

[A-23] I think that the author/s should review the following two sentences: "If considering off-the-shelf tools, as Matlab to ILP and GA, the coding is simpler. However ESBMC and vZ has BSD-Style and MIT licenses... can be downloaded ... free". Matlab is a (non-free) general-purpose mathematical software, and ILP and GA are techniques to solve optimization problems. In the internet, the author/s can find several GA and ILP libraries (simples and BSD-Style and MIT licenses and publicly available -free-) for different programing languages. Consequently, I think that, in a certain way, those two sentences are not totally true and hence conclusive.

> **Response:** We completely agree with the reviewer. We have decided to rewrite that entire sentence in the Conclusion Section (see Section 6) to address the reviewer's comment.

[A-24] Is T(s) affected by the programing language in which is coded the proposal? ESBMC-based approaches are coded in C++ but ILP and GA are coded in Matlab programing language.

> **Response:** Yes, we agree with the reviewer that the selected programing language affects the performance tests. Furthermore, MATLAB is an interpreted language, while ESBMC is compiled (and runs binary code). Therefore, we decided to include a new sentence to explain the impact of those tools in the Experimental Setup Section (see Section 5.1).

[A-25] What is the reason to obtain MO in ESBMC but not in ESMBC-(SS|PS|PB|vZ)? I think that it should be analyzed.

> **Response:** Regarding the amount of MOs, the sequential ESBMC approach has to explore all (possible) states until it finds the HW-SW partitioning solution; it starts from an extreme, where all variables are selected as software, and then incrementally tests one by one to check whether a given node will be implemented in software or in hardware. In contrast, all multi-core ESBMC approaches and ESBMC-vZ, are optimized to reach faster the HW/SW partitioning solution than the sequential one, without the need for exploring all states as the sequential ESBMC approach does. As a result, if the HW/SW partitioning problem grows in complexity, then the sequential ESBMC approach easily tends to reach MO due to the state-space explosion problem. This information was included in Section 5.3.

[A-26] Is ESBMC-vZ a multi-core proposal? I think that the experimental results are contradictory with regards to title and address of your investigation. In consequence, the contributions and research directions should be clearly express and discussing in the abstract, introduction and conclusions.

> **Response:** We really appreciate the reviewer's comment. Actually, vZ is not multi-core and that information appeared just once throughout the text. However, we have adapted the manuscript in different parts (i.e., introduction, results, and conclusion), in order to clarify that term for the reader.

[A-27] Instead of GA, the author/s could study other metaheuristic techniques or local search techniques to solve the HW/SW partitioning problem. Besides, it should be studied and analyzed the similarity between certain search local techniques and ESBMC.

> **Response:** The reason to use ILP and GA was included into this new revised paper version (see Section 4.1). In addition, referencing the question [A-9], there is now an explanation of this paper focus.

Responses to the comments of Reviewer #B

[B-1] Improve the abstract, adding information about which method is better and what conditions for that

> **Response:** We appreciate the advice. This section was rewritten.

[B-2] In page 8, column 2: " Each time was measured 3 times (average taken). Based on standard deviation and tolerance interval to each set of time sample, it was obtained a statistical confidence of 91.7% to ESBMC (sequential, SS, PB and vZ), 95.9% to ESBMC-PS, and 92.0% to ILP and GA. A timeout condition (TO) is reached when the verification time is longer than 3600 seconds". These results are o.k BUT, the problem is 3 times is very short number of times for making an experiment. Generally, people use more than 30 (and use Normal Gaussian on the analysis) or minor 30 (uses T-student). Why the number (3). It is no adequate. Repeat the experiments more times, or show a better justificative for it.

> **Response:** The reason of the time that is measured 3 times is based on previous empirical tests, where a larger measurement (e.g., 5 or 7 times) did not produce significant differences in the experimental results (which were always below of 10% and mostly around 3%). Therefore, we decided to fix it in three times. We also included that information into the paper, see Section 5.1. We also refer the reviewer to [A-19] that describes about the timeout information.

[B-3] It was used three programs from Mibench benchmark (CRC3, Dijkstra and Patricia). Why them ? Mibench has 8 programs into 6 categories. Explain the motivation for only these 3 programs. It is possible to add information about more program from Mibench ?

> **Response**: We have followed the same evaluation scheme used in previous (related) work, in order to choose those three benchmarks. Therefore, in order to compare our proposal to other existing

approaches, it was necessary to keep the same benchmarks. Now this information is highlighted in the manuscript (see Section 5.2).

[B-4] Improve section 6 (related works), adding information about some results from other related works, and compare them to your results. Maybe a table ....and discussing, when possible, when your results are better or not. Specifically results from four references: Ramalho et al. (2013) and Cordeiro et al. (2012), and Trindade and Cordeiro (2015), and Trindade et al. (2015).

Response: See item A-21. The section was rewritten and moved to the introduction of the paper, following the IJES guidelines. With that, we tried eliminating redundancy, which was also a problem of Section 6 with Introduction and Contribution sections.

[B-5] Finally, organize the references in alphabetical ordering.

Response: We thank the reviewer for point out this mistake. We have fixed the references according to the bibliographical style defined by the IJES guideline.

Responses to the comments of Reviewer #C

[C-1] The introduction in Section 1 is hard to follow. This section must be rewritten describing the importance of the hardware/software partitioning problem with appropriate references of hardware/software partitioning problem. The author uses the term "according to" for references which is grammatically incorrect.

Response: We appreciate this comment. The introduction was rewritten, the related work redundancy (from introduction, contributions, and related work sections) was removed in the new revised version, where all the information are included into the Introduction Section only. There is no more the term "according to" throughout the paper.

[C-2] The introduction to the Optimization in subsection 2.1, OpenMP architecture in 2.3 and the vZ tool in 2.4 is completely redundant. These are well known techniques and the author must consider to add more works on the hardware/software partitioning problems and their variants in this section.

Response: We agree that Optimization, OpenMP architecture, and vZ are well known techniques. However, when these issues are put together, they are not known in depth by most embedded systems engineers. Although the formal verification community is familiar with all that topics, we think that it would be necessary to keep that part of the paper to introduce the subject to the embedded systems community (e.g., vZ is not well-known by most embedded systems researchers). More references about HW-SW partitioning were also included, as suggested by the reviewer.

[C-3] The Formal model described in Section 3.2 is hard to follow. The problem statement for HW-SW partitioning is ambiguous. The author must describe the problem statement using formal semantics. In particular, the variable S_0 is never described. Further, the assumptions proposed in the informal model in section 3.1 must be formalized. Without this, there is no formal modelling of the partitioning problem.

> **Response:** We did some text modifications in Sections 3.1, 3.2, and 3.3 to clarify the formal model as suggested by the reviewer. Variables were defined as well. Maybe it is not explicit in the paper, but the assumptions proposed in the informal model (Section 3.1) appear in the formal model (Section 3.2), as example, the value assumed by decision variable "x" is *zero* or *one* depending on the context which it is allocated (software or hardware); additionally, the communication, software, and hardware costs are present in the formal model, according to the assumptions taken during the informal model.

[C-4] The assumptions made for the partitioning constraint are too coarse. What parameters are actually considered for the hardware cost (area, power, heat ?). In real embedded system, each of these parameters can play significant role in the cost factor. The running time of hardware are considered as zero. Is this assumption required to simplify the model ? It is assumed that there is only one hardware and software context. The author must describe the complexity of their technique for multiple contexts of hardware and software.

> **Response:** We just realize that the text is not clear about the assumptions and the HW-SW model. Therefore, we did some text modification in the Informal Model (Section 3.1) and in Conclusions (Section 6). The reviewer is correct about the choice of a simplified model: the focus is on first-generation co-design as stated in the Informal Model Section. Therefore, in this new revised version of the paper, we have tried to make it clearer.

[C-5] The term TipH and Hmax is used in Figure5 but is not declared. Figure 5 uses "Populate", but actually it is assignment of decision variables. The pseudocode in Figure 5 is extremely inconsistent. It is very hard to follow what each step of the algorithm is achieving. For the three multi-core version of ESBMC, the author must comment on the complexity of ESBMC-SS, ESBMC-PS, ESBMC-PB. Also, the Open-MP overhead must be described for each version of these algorithms.

> **Response:** Figure 5 was overhauled. Now, all variables are properly declared and described in the manuscript. With respect to the time complexity of ESBMC-SS, ESBMC-PS, and ESBMC-PB algorithms, they can all be described in two parts, which include the parallelism and the optimization solving. In the first part, the ESBMC-SS, ESBMC-PS, and ESBMC-PB time complexity is considered to be linear ({\it i.e.}, they are denoted by $O(n)$), taking into account a sequential time since each algorithm runs all possible solutions at once. However, each execution instance of ESBMC-SS, ESBMC-PS, and ESBMC-PB solves a specific

> optimization problem that is considered to be NP-Hard, as described by Arato *et al*. Thus, even a parallel execution being implemented, including (possible) overheads due to the use of the OpenMP library, the time complexity of ESBMC-SS, ESBMC-PS, and ESBMC-PB is still NP-Hard. We included this explanation in Section 4.3.4.

[C-6] In the experimental section, the author uses “as shown” at several places to refer to different tools. This is grammatically incorrect. The sentence “it was obtained a statistical confidence” in the second paragraph of section 5.1 is unclear.

> **Response:** We removed all the “as shown” from the revised text. The statistical information is not clear, and we did some text adjusts in order to clarify the information in the “Experimental Setup” section. However, in a nutshell, the correct is “confidence level”, and is taken from the mean, the standard deviation, and tolerance interval for each set of time sample.

[C-6] In section 5.2, what does “time dimensional” mean as software cost, does it mean running time of software and the communication time. Also, how the area is computed to evaluate the hardware cost must be explained in the paper. The sentence “are designed from Mann” is grammatically incorrect. Section 5.2 last sentence need to be rephrased.

> **Response:** We agree with the reviewer. In particular, “time dimensional” means running time; that information was added to the new revised version of the paper. We have also conducted a complete and thorough revision of Section 5.2 in order to make it correct and clearer.

[C-7] The rows and columns of Table 2 are never described in the paper. So, the results in the table does not make any sense. What is T(s), Sp and Hp in the table. These symbols are not introduced in the text.

> **Response:** We appreciate the reviewer’s advice. Now all the missing information is properly described in the “Experimental Results” section.

[C-8] The results in the paper is bit confusing. The main idea of the paper is based on the idea of implementing different configurations of ESBMC supporting multi-core features. But the results show that ESBMC with the vZ backend that uses a single-core implementation always wins over other ESBMC configurations as well as ILP (except RC6). This defeats the claims made by the authors about the possible speed-up obtained due to the multi-core support of model checking. The parallelisation of a sequential algorithm with Open-MP often leads to speed-up due to obvious reasons. So, the better performance of multi-core ESBMC implementations over sequential version is not surprising.

> **Response:** Based on rewritten parts of the text (introduction, experimental results, and conclusion), we have tried our best to clarify the comparative information about all techniques. We agree the reviewer is correct about the aspect that vZ-ESBMC is faster than any other parallelized ESBMC, but vZ is specialized to solve linear

> optimization problems. The variety of parallel ESBMC approaches was an effort to test, compare, and measure the difference among them, which was never done before in the literature.

[C-9] The experimental result need more rigorous analysis to understand what kind of constraints are generated from the partitioning problem which are solved by the vZ tool. The benchmarks for which vZ tool time-out need further explanation. How does the problem structure influence the solving time by the backend solvers is worth investigating. A possible direction is to compare solving times for Boolector versus vZ, and compare the statistics of the number of decisions, backtracking, restarts, deduction, clause learning made by each of these solvers. What happens when the timeout is increased ? Are these backends able to solve the problem for higher timeout.

> **Response:** We have included an explanation in the paper (at the end of Section 5.1) as follows: "The TO was defined based on previous empirical tests, where a larger TO (e.g., 5000 seconds) did not produce substantial differences in the experimental results". Additionally, we have tried to obtain the statistics of the number of decisions, backtracking, restarts, deduction, and clause learning made by each of the solvers (e.g., Boolector and Z3), as suggested by the reviewer. In particular, we have added the command *(get-info :all-statistics)* at the end of each SMT file generated by ESBMC, but only Z3 was able to provide those statistics. Boolector reports "unsupported command 'get-info'". If we read carefully the SMT-LIB specification document, it states that *":all-statistics replies with the values of a number of solver-specified statistics about the current state of the solver. The content is not defined by SMT-LIB (yet)"*. As a result, we believe that it is difficult to compare the solvers in terms of those statistics suggested by the reviewer. Consequently, we state in Section 5 that the main objective of our experimental evaluation is to compare among MATLAB, ESBMC-SS, ESBMC-PS, ESBMC-PB, and ESBMC-vZ using a set of standard HW-SW partitioning benchmarks. Thus, the intention of this paper is not to compare the algorithms implemented in Boolector and vZ (or other SMT solvers) in terms of the number of decisions, backtracking, restarts, deduction, and clause learning; the SMT-COMP already does a very good job in comparing SMT solvers. Here, our intention is to compare the performance in terms of time (and correctness) of all proposed approaches against existing ones related to HW/SW partitioning. Furthermore, throughout the paper, we describe the constraints that are actually generated to Boolector and vZ; in particular, the constraints generated from the partitioning problems for vZ are described in Section 4.4 (see Figure 12).

[C-10] It is important to know the details of these benchmarks which would give information about the problem structure and the type of constraints that are generated. Also, the number of benchmarks are too few to draw any conclusion. The author must run the experiments with more benchmarks which has > 200 nodes.

> **Response:** Each benchmark file has specific information, which was defined during the informal/formal sections of the paper: hardware cost to each node if implemented in hardware, software cost to each node if implemented in software, communication costs if the node is partitioned to different contexts, initial cost of software, transposed incident matrix of the directed single graph responsible for establishing the connections among the nodes. We believe that the most important thing is to run the same benchmark with each technique included in the paper. We agree that, considering the limits of each technique must be necessary to include more benchmarks (ranging from 100 to 400 nodes) to conclude precisely. Therefore, we modified the "Experimental Results" and "Conclusions" sections to make this information clear.

[C-11] The first two paragraphs of related work contains several grammatical mistakes. The conclusion section is full of errors. What does "faster and notorious" and "expressive" mean as execution times.

> **Response:** The "Related work" section was completely rewritten and moved to introduction, as suggested by the reviewers. The "Conclusion" section also suffered a few modifications and now there is no "expressive" word anymore.

[C-12] Overall, the paper just compares a bunch of multi-core variants with different backend solvers which are implemented on top of ESBMC tool. Thus, it is more of an experimental paper and does not provide any new insight to solve the hardware/software partitioning problem. There is no new theory in the paper and the existing formalisms are extremely straightforward with coarse assumptions which limits the scope of the paper.

> **Response:** We have emphasized the main contributions of our manuscript in Section 1.1. We agree with the reviewer that we do not describe a new (breakthrough) theory since SMT-based BMC techniques have been around for nearly one decade and only MaxSMT is somewhat recent, but we have not seen yet the application of those techniques (in combination) to solve HW/SW partitioning problems in the literature. To the best of our knowledge, this is the first work to use a multi-core SMT-based verification and a MaxSMT solver to check for HW-SW partitioning problems in embedded systems.

[C-13] Another major concern of this paper is that the text contains several grammatical mistakes and completely meaningless sentences at several places. The authors must focus on rewriting the paper with correct sentences and proper grammatical construction. The paper also need significant restructuring. A lot of sections are redundant and takes up space without providing too much new information. The author must consider removing these sections.

Response: We appreciate the reviewer’s advice. Some sections of the paper were rewritten; a completely restructuring was also carried out about the related work section to avoid redundancy. Moreover, additional explanation was made in different parts of the paper to clarify it further.