Statement on the revision of paper wksp129

Simon Conrady Manu Manuel Arne Kreddig Walter Stechele

April 26, 2019

The authors would like to thank the reviewers and editors for their comments and suggestions, which significantly helped to improve the manuscript. We believe that we have addressed the different notes in the revision of the manuscript. Below, we list all of the notes given in the review, provide our responses and indicate how the manuscript was adjusted. We use the following convention to distinguish the review comments and our answers:

> Comment or note provided by committee or reviewer

Our response or comment

Part of the manuscript which has been modified or added to answer the comment

Notes & Comments from the Committee

Committee Note 1

> Clearly state why you base your approach on LCS.

We believe that employing an LCS for tuning the configuration of approximation methods has many advantages. The motivation to use the LCS is based on the applicability of LCS-based frameworks to problems from neighboring fields as demonstrated by other works. As such a rule-based approach can be decoupled from the inner workings of the approximation methods it configures, it is not restricted to specific approximations. Because the reward is generated from a global view on the application, it automatically also considers interactions between various approximation methods. Finally, we believe that the best rules emerging from the learning phase can be used to change parameters on the fly with relatively low computational overhead.

We have rephrased and complemented part of the introduction section to highlight our inspiration and reasoning for using an LCS-based approach as follows:

Previous works have shown the applicability of learning classifier systems (LCSs) to the configuration [8] as well as workload distribution [34] for embedded systems. Inspired by the results of their research, we propose an LCS-based approach which optimizes the parameters of multiple approximations on FPGA devices for image processing applications. The core of the LCS is a set of rules with conditions and actions collectively modeling an intelligent decision maker along with the fitness function [29]. The actions are used to adjust the configuration of approximation parameters and are chosen depending on conditions which reflect the current system state and also based on fitness values generated during the learning phase. A major benefit of this rule-based approach is that it is not restricted to specific kinds of approximation methods and that it can be used to optimize arbitrary combinations of approximations on system level. Furthermore, the best rules trained in the learning phase can be used to dynamically adjust parameters during runtime without incurring high overhead.

Additionally, we rephrased part of the section on related work to give more details on the works that motivated us:

> Danek et al. employed the idea of LCS to FPGA technology mapping problems [8]. Their work introduced a rule-based eXtended classifier system (XCS) adaptive mapper in order to achieve good area and performance results on heterogeneous FPGA. The mapper is trained on a benchmark circuit and evolves a set of generic mapping rules during the learning phase with minimal number of CLB and critical signal path delays. Moreover, a reward is generated for each action which reflects the decrease in the number of CLB and critical path delay along with the utilization of generated CLB. The adaptive nature of the XCS mapper ensures that the final rules take the global characteristics of the FPGA into account. This work shows that an LCS can be effectively applied to optimize the configuration of FPGA systems. However, it does not consider any approximation methods. In the context of performance and power optimization of System-on-Chips (SoCs), Zeppenfeld et al. introduced the learning classifier table (LCT) [35], a simplified XCS-based reinforcement learning technique, and used the concept for dynamic parameterization to optimize task distribution and workload management in multi-core systems [34]. The LCT monitors the current workload of the cores during runtime, dynamically scales core frequencies, and migrates tasks between cores to achieve optimal system utilization. Their results demonstrate the applicability of an LCS-based approach to the dynamic configuration of embedded systems with low overhead.

Finally, we complemented part of the related work section that reviews other approaches for tuning approximated systems to show the drawbacks in current approaches that we aim at overoming with our approach:

> Vasicek et al. introduced a two-stage approximation and optimization concept which uses cartesian genetic programming (CGP) on gate level [30]. The objective is to reduce the number of gates in the circuit implemented on the FPGA with a tolerable error rate. To achieve this, the CGP is applied once to the exact circuit to obtain an optimized error-free design first and then again after introducing a certain level of error, resulting in an area-reduced approximate design. This approach directly approximates circuits on gate level and is therefore not applicable to higher level approximation methods. Further, their approach completely eliminates the possibility of dynamic adaption. Akhlaghi et al. proposed an approach that uses gradient descent to introduce multiple approximations in a data flow graph [1]. However, their system can only handle approximate adders and main memory. It is also not suited for dynamic adaption. Regarding the challenge of dynamically adapting approximation parameters, Laurenzano et al. have presented a framework that analyses the current input and searches for the best approximation methods and parameters to adequately process this input [17]. While this approach enables the system to fine tune the approximations to every input, it incurs high runtime overhead and is limited to approximation methods in software.

Committee Note 2

> Add a paragraph which clearly states how the proposed variant relates to standard LCS approaches (e.g., XCS) and which modifications have been made for which purposes.

Our system inherits many of the main concepts used in the Michigan-style LCS but does not directly relate to any specific existing LCS. We added a paragraph to the end of the section on our proposed approach to state the major similarities and differences:

The proposed approach is not directly equivalent to any of the existing LCSs. However, it inherits basic concepts from the Michigan-style LCS by using a finite rule set population that represents the problem solution [12]. Like Zeppenfeld et al. [35], we derive probabilities from the fitness values for the prediction mechanism. The condition function plays the role of the detectors which populates the match set rules with the relevant actions. In contrast to general LCSs, all conditions are mutually exclusive in our approach, but multiple rules share the same condition. Therefore, the resulting match set can not contain duplicate actions. Our initial concept uses a deterministic set of rules, but we are evaluating the possibility of integrating the genetic mechanism as well. However, we are using reinforcement-based credit assignment to iteratively update the fitness values of selected rules with benefits and drawbacks to end up with an optimal rule set which can be used for parameter adjustments at runtime.

Committee Note 3

> Since space is not an issue (limit is 8 pages excluding references), at least a short section introducing prerequisites for the FPGA domain as well as for LCS-based learning is requested.

As requested, we added a background section that contains basic information for LCS-based learning and the major LCS styles as well as an introduction into FPGA architectures and design challenges.

Committee Note 4

> Additionally, please make sure to also address each of the other reviewer suggestions as indicated in their comments.

The comments and suggestions from the reviews have been addressed as described in the following sections.

Notes & Comments from Review 1

Review Note 1.1

> It is not clear what extensions are added on top of a standard LCS to solve this problem, and even before that, it is not clear to my why you need an LCS at all. This task seems to me an optimisation problem.

*We believe that the points raised in this comment have been picked up by the committee in **Committee Note 1** (reason for using an LCS-based approach) and **Committee Note 2** (relation of the proposed system to a standard LCS). We complemented and rephrased parts of the introduction and related work sections to highlight the motivation and benefits of using an LCS. Furthermore we added a paragraph to the description of our proposed approach to state similarities and differences to a standard LCS approach. For more information on the changes, we refer to the response given to the committee notes.*

Notes & Comments from Review 2

Review Note 2.1

> Additionally, in view of the remaining space, a more "gentle" introduction to FPGA-specific details would be very helpful for unfamiliar readers.

*This comment has been picked up by the committee in **Committee Note 3**. We added a section that introduces FPGA details to the manuscript (see the response given to the committee note).*

Review Note 2.2

> Expressions such as "select condition" and "fitness table" appear very uncommon to me. Thus, I strongly encourage the authors to highlight the conceptual and architectural differences to conventional Michigan-style LCS variants more prominently.

The request to highlight the relation of the proposed system to conventional LCS variants was picked up by the editors in ***Committee Note 2****. We rephrased the description of the proposed system for more clarity and removed the term fitness table to avoid confusion. We also added a paragraph to clarify the relation of our approach to a standard LCS. For more details, we refer to the response given to the committee note.*

Review Note 2.3

> The reference [25] cited for referring to the introduction of XCS is not the correct one. XCS was introduced in 1995 by Wilson.

This was an oversight on our part. We added the correct reference (see below) to the bibliography and replaced the citation in the manuscript.

[31] Stewart W. Wilson. 1995. Classifier Fitness Based on Accuracy. Evolutionary Computation 3, 2 (Jun 1995), 149–175. `https://doi.org/10.1162/evco.1995.3.2.149`

Review Note 2.4

> Since the contribution is already rather shallow, since no results are presented, the authors should spend a section on LCS basics.

This comment has been picked up by the committee in ***Committee Note 3****. As for the FPGA introduction, we also added a section on LCS basics to the manuscript (see the response given to the committee note).*

Review Note 2.5

> The references need revision -> Please be consistent with the data you provide in the bibliography. Either provide information such as pages and location for any references, or omit them consistently. Further, I would prefer to see the doi's removed.

We added some missing information on page numbers to the references as well as some missing author information. Further, we decided to keep the DOIs as we think they help accessing the original source of the references faster and because they are also used in the sample-sigconf *template. For consistency, we omitted the location information for all references and added a link (DOI or URL) to all references.*

Notes & Comments from Review 3

Review Note 3.1

> Why use LCS? There are multi-objective optimization methods that take tradeoff into account. Please show the benefits of using LCS, which is a rule-based approach.

This has been summarized by the committee in ***Committee Note 1****. We have added our motivation and argumentation for using LCS to the introduction and related work sections. For further information, we refer to the response given to the committee note.*

Review Note 3.2

> A detailed explanation of the figures should be in the text. (In particular, Figure 1)

To improve clarity in the description of our approach, we added multiple links between the text and the overview figure. Also, we added a paragraph giving a high-level overview of the system as displayed in the figure, leading into the more detailed descriptions of single system components.

> The system is composed of different components: the approximated application and its parameters which are implemented in FPGA, the LCT which holds all relevant LCS data, and both the Condition and Fitness Function which work on the LCT data to train the LCS. These components are explained in detail in the following.