

Dependability-Aware System Synthesis: SystemCoDesigner, OPT4J and JRELIABILITY

Michael Glaß, Martin Lukasiewicz, Felix Reimann, Martin Streubühr,
Joachim Keinert, Christian Haubelt, and Jürgen Teich

{glass,martin.lukasiewicz,felix.reimann,streuebuehr,keinert,haubelt,teich}@cs.fau.de

Hardware/Software Co-Design - University of Erlangen-Nuremberg – Germany

Abstract

The presented dependability-aware system synthesis approach automatically performs a redundant task binding and placement of voting structures to increase both, reliability and safety.

1. Introduction

Embedded systems typically consist of many interconnected processing units, e.g., homogeneous or heterogeneous MPSoCs, automotive and avionics ECU networks, etc. Since these systems are operating under different radiation and varying temperature conditions, a high dependability of such system becomes necessary.

In the presented framework, a novel dependability-aware design space exploration approach is introduced. During system synthesis, a redundant binding of tasks is performed automatically to increase reliability and fault detection mechanisms are inserted to improve the safety. The fault detection and, if possible, fault toleration is done by *voters*, i.e., hardware resources or software tasks that aim to find a majority of k identical values delivered by n redundant task instances. Such voters are usually implemented as so-called *duplex voters* (2-out-of-2-majority) or the well known *Triple Modular Redundancy* (TMR, 2-out-of-3-majority).

The entire system synthesis approach is realized in the *SystemCoDesigner* [1] framework and shown in Figure 1: Given a system specification written in *SystemMoC* [2] that is automatically transformed to a graph-based model, the framework performs a symbolic multi-objective design space exploration with OPT4J [3] as the underlying optimization engine. Within this optimization approach, efficient analysis techniques evaluate the systems reliability, safety, and other properties like performance based on SystemC simulation [4]. The dependability evaluation is implemented based on the JRELIABILITY [5] framework. This approach significantly extends previous work by means of (1) automatically integrating fault detection and correction mechanisms and (2) automatically evaluating not only the reliability, but also the safety of an implementation, thus, leading to high-quality solutions.

Given the high-quality solutions, an automatic prototype generator synthesizes the SystemC model for a target platform, cf. [1]. The effectiveness of the proposed approach is shown by providing results from a design space exploration of a Motion-JPEG decoder application and automatic prototyping optimized solutions for an FPGA platform.

2. Concepts

The presented synthesis approach aims to determine a set of possible high-quality implementations for a given system

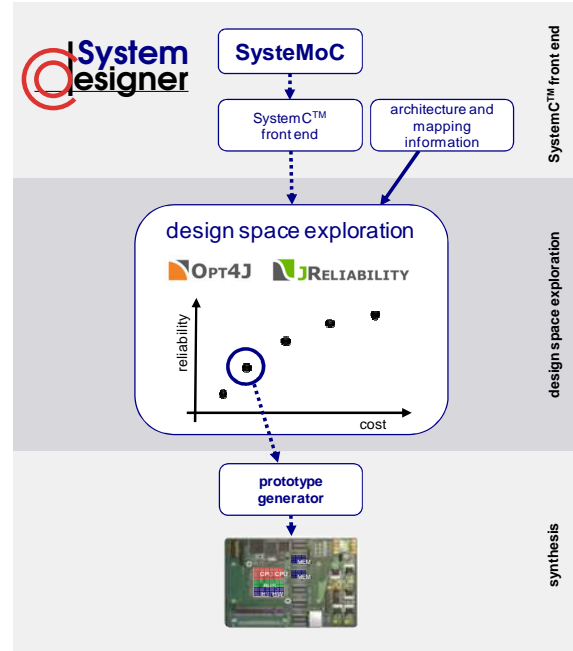


Figure 1: The SystemCoDesigner design flow.

specification by a design space exploration approach. In dependable system design, structural redundancy is an important technique to improve the systems characteristics with respect to reliability and safety. The usage of arbitrary voting structures allows to detect and to tolerate transient and permanent errors in case a fail-silent behavior cannot be assumed. This is typically the case when dealing with SoCs and MPSoCs. For an automatic flow, the used redundancy and voting structures should be transparent for the designer, but, instead, part of the optimization. Thus, the dependability-awareness is introduced at the highest design level where transparency for the designer is still given and, therefore, is embedded in the phase of design space exploration.

The design space exploration is carried out using meta-heuristic search techniques like, e.g., *Multi-Objective Evolutionary Algorithms* (MOEAs). These search techniques are provided by the OPT4J optimization framework. In embedded system design, different objectives like, e.g., monetary cost, area and power consumption, or dependability, have to be considered. Thus, the optimization process is guided by a set of so-called evaluators to estimate the properties of a found implementation. Moreover, stringent constraints arising from the computational capacity of the used processing resources and the used communication resources imply that the search space contains very few implementations that do

obey the given constraints. Finding these so-called *feasible* implementations is a challenging task itself. For this purpose, a symbolic approach called *sat-decoding* [5] has been proposed. A 0-1 *Integer Linear Program* (ILP) solver is used to gather feasible implementations while the meta-heuristic search is used to optimize the objectives.

For the dependability-aware system synthesis approach, the necessary 0-1 ILP formulation for the decoding phase is extended to allow an automatic voter placement. This is achieved by permitting the placement of several instances of a single task in the system, cf. [6]. The data produced by the different task instances can be voted by the preceding tasks such that permanent or transient errors, respectively, can be detected and, if possible, tolerated.

For the dependability analysis, the different characteristics of arbitrary voting structures require to evaluate both, the reliability and safety of the system as two conflicting objectives. As measures, the *Mean-Time-To-Failure* (MTTF) for reliability and *Mean-Time-To-Unsafe-Failure* (MTTUF) have been used. These are effectively determined by a modeling of the system as Boolean functions encoded in *Binary Decision Diagrams* (BDDs), cf. [6, 7]. This encoding allows a fast determination of the MTTF and MTTUF using a specific Shannon-decomposition provided by the JRELIABILITY framework.

3. Case Study

As a case-study, a system specification of a Motion-JPEG decoder, cf. Figure 2, is used. The system specification is given as a *SystemC* behavioral model written in *SysteMoC*. Each *SystemC* module corresponds to a task in application and is transformed into software modules by code transformation or into hardware accelerators (resources) by using a behavioral synthesis tool. In the latter case, the *Forte Synthesizer* is used which allows a swift extraction of important characteristics of the hardware accelerator like, e.g., throughput or required area.

For the desired *Xilinx* FPGA target platform, resource needs in form of flip-flops, look-up tables, and block RAMs are estimated. Based on these values, the failure rates of the system elements can be estimated. Using the specified behavioral model and the estimation of the characteristics, a design space exploration model can be derived automatically.

A reference implementation x_0 without a multiple task binding and, thus, without voters was selected to quantify the amount of additional costs as well as reliability and safety increase using the proposed approach. In the following table, the best found implementations are compared to the reference implementation:

ID	LUTs	flip-flops	BRAM	MTTF	MTTUF
x_1	+80%	+88%	+3%	+81%	+81%
x_2	+26%	+51%	+24%	+69%	+69%
x_3	+28%	+61%	-3%	+18%	+83%

Timing properties are neglected since even by inserting five voters in the data path, the delay only increases by 58.9 μ s (0.47%).

4. Conclusion

In this project, dependability-awareness has been introduced into an embedded system synthesis approach. A symbolic placement of arbitrary voting structures, fault detection and fault toleration mechanisms are automatically integrated into the system implementations. A saving of additional costs is forced by a multiple binding of tasks taking resource reuse in account.

To evaluate the introduced dependability and to allow an optimization together with other objectives like, e.g., monetary costs, power consumption, or latency, an efficient symbolic analysis approach was proposed that quantifies lifetime reliability and safety.

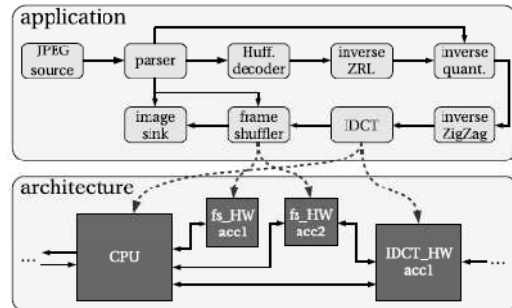


Figure 2: Specification of a Motion-JPEG decoder consisting of the application, an excerpt of the available architecture, and mapping edges from tasks to resources.

5. References

- [1] J. Keinert, M. Streubühr, T. Schlichter, J. Falk, J. Gladigau, C. Haubelt, J. Teich and M. Meredith, “SystemCoDesigner—An Automatic ESL Synthesis Approach by Design Space Exploration and Behavioral Synthesis for Streaming Applications.”, *ACM Transactions on Design Automation of Electronic Systems* 14(1), pp. 1-23, 2009.
- [2] J. Falk, C. Haubelt and J. Teich, “Efficient Representation and Simulation of Model-Based Designs in SystemC.”, *Proceedings of FDL ‘06*, pp. 129-134, 2006.
- [3] OPT4J – The Optimization Framework. Version 1.5. URL: <http://www.opt4j.org>
- [4] M. Streubühr, J. Falk, C. Haubelt, J. Teich, R. Dorsch, and T. Schlipf, “Task-Accurate Performance Modeling in SystemC for Real-Time Multi-Processor Architectures.”, *Proceedings of DATE ‘06*, pp. 480-481, 2006.
- [5] JRELIABILITY – The Java-based Reliability Library. Version 1.2. URL: <http://www.jreliability.org>
- [6] M. Lukasiewicz, M. Glaß, C. Haubelt and J. Teich, “Efficient Symbolic Multi-Objective Design Space Exploration.”, *Proceedings of ASP-DAC ‘08*, pp. 691-696, 2008.
- [7] F. Reimann, M. Glaß, M. Lukasiewicz, C. Haubelt and J. Teich, “Symbolic Voter Placement for Dependability-Aware System Synthesis.”, *Proceedings of CODES+ISSS ‘08*, pp. 237-242, 2008.
- [8] M. Glaß, M. Lukasiewicz, F. Reimann, C. Haubelt and J. Teich, “Symbolic Reliability Analysis and Optimization of ECU Networks.”, *Proceedings of DATE ‘08*, pp. 158-161, 2008.

Acknowledgement

The authors would like to thank the German Science Foundation SFB 694 and SPP1148 for partially supporting this work.