Research Survey Report on Embedded System

Ankita Gupta

MTech, Embedded System

Department of Electronics and Tele-Communication, Symbiosis Institute of Technology

Object Oriented Programming and Data Structures

Prof. Shripad Deshpande

November 04, 2023

**Title**

Method for Testing and Debugging Flow Formal Specification in Full-Stack Embedded Systems Designs.

**Abstract -**

This paper proposes a standardized approach to testing and debugging for full-stack embedded computing systems. It suggests an ontology of ES testing and debugging processes, as well as several models that utilize a unified representation of testing, verification, validation, and debugging problems. Overall, this approach provides a more efficient and reliable method for testing and debugging full-stack embedded computing systems.

Keywords: embedded computing systems; multi-level modeling; verification; validation; design flow.

**Introduction**

Embedded Computing Systems (ES) are specialized systems that interact with complex physical objects. Designing these systems presents unique challenges, including combining various technical solutions in a custom design flow while considering computing system organization levels. Quality assurance procedures are necessary to ensure compliance with the target destination despite incomplete initial specifications. This article addresses developing a universal formal description for the ES design flow from a testing and debugging perspective.

**Literature Review**

ES design flow formalization, testing, and debugging are actively studied by leading groups and companies in ES design automation tools. Some methods like Sangiovanni-Vincentelli proposed a platform-based approach for system design using a set of components to build a system representation at a specific level of abstraction. While E.A. Lee extended this to a hierarchy of platforms, but without studying testing and debugging. In high-level design stages, requirements tracing is used to track implementation and impact on system design. Many tools are available for testing, verification, and validation, including SPIN, Uppaal, MATLAB/Simulink, and Vivado HLS. When components are combined at the same level of abstraction, verification is effective. But when technologies of different levels are combined, there are gaps between these levels that require manual transfer of design information. These gaps lead to logical and functional errors during testing and imperfect verification methods.

**Methodology**

        Ensuring compliance with computing system requirements involves identifying and fixing any errors, from minor typos to major system-level issues. Testing and debugging, including verification and validation, are key processes that ensure the final system meets all requirements. To solve the problem, a proposed solution is to create a translation hierarchy called ES. This hierarchy is a directed tree structure where nodes represent various system representations and components within their platforms, and arcs indicate translation relationships between them.  A representation is the implementation of a system or component on a computing platform at a certain level of abstraction. It includes all related artifacts, from implementation to testing and debugging. Sink nodes represent target implementations, while the source node is the initial specification in the multi-level model.  

**Results**

               The proposed method analyses design routes for gaps that limit error detection and correction. Each translation has a semantic and test data transfer gap, which can be identified through verification and validation. The authors developed automated tools for ES testing and debugging based on high-level record and replay of the computational process to bridge some of these gaps.

**Conclusion**

The article proposes a formal approach to ES testing and debugging flows. However, integrating this approach with existing methods and tools to achieve automation at high levels of abstraction requires further elaboration. The authors are developing a complete methodology using multi-level models and specialized automated tools for real design practice.

**Bibliography**

V. Pinkevich and A. Platunov, "Method for Testing and Debugging Flow Formal Specification in Full-Stack Embedded Systems Designs," 2020 9th Mediterranean Conference on Embedded Computing (MECO), Budva, Montenegro, 2020, pp. 1-4, doi: 10.1109/MECO49872.2020.9134213.

**Title**

Control of Black-Box Embedded Systems by Integrating Automaton Learning and Supervisory Control Theory of Discrete-Event Systems.

**Abstract**

The paper proposes an approach to control black-box embedded systems by integrating automaton learning and supervisory control theory. First, the system is tested against the requirements. If it fails, a supervisor is synthesized to enforce compliance. A C-learning algorithm is used to infer a Moore automaton that describes both the system's behaviour and the requirements. The controlled system is tested again to check the correctness of the supervisor. If the requirements are still not satisfied, a larger Moore automaton is learned, and a refined supervisor is synthesized. This process repeats until the requirements are met.

**Introduction**

Complicated embedded systems often use third-party components or legacy subsystems to reduce development costs. However, if these components fail to meet the requirements of the new application, modifying the system by changing its source code is impossible. An alternative approach is to use a "supervisor" component to monitor the reused component and correct it if necessary. Supervisory control theory (SCT) of discrete-event systems (DES) requires formal models of the plant and requirements, but the formal models of reused components are often unknown. This paper proposes an approach to synthesize a supervisor for a system by integrating automaton learning algorithms and SCT to make requirements satisfiable when the automaton models of both the system and requirements are unavailable. his paper proposes a novel approach to supervisory control of black-box embedded systems by integrating automaton learning algorithms and SCT. The approach constructs a Moore automaton using the C∗ algorithm adapted from the L∗ learning algorithm to describe the system's behavior and its conjunction behavior with requirements. A supervisor for the system is computed based on the learned automaton by SCT and implemented as a patch to monitor and control the original system. The procedures of automaton learning, supervisor computing, and system testing are iterated until the requirement holds. The paper provides two main contributions: a supervisory control approach for black-box embedded systems, and a C∗ learning algorithm based on the L∗ algorithm to infer a Moore automaton enabling synthesis of the supremal nonblocking supervisor of the problem from the learned automaton by SCT.

**Methodology**   
The approach involves testing a black-box system and a requirement. If the system fails, a supervisor is designed to prevent the system from reaching unacceptable states. Automaton learning algorithms and SCT are used on abstracted systems to ensure effectiveness. The C∗ algorithm infers a Moore automaton capturing the system and requirement behavior. A supervisor is computed and added to the system to ensure correctness. The system is tested again and if it fails, a new supervisor is patched to the original system until the controlled system meets the requirement.

**Conclusion**

This paper proposes an approach to improve the quality of black-box embedded systems in situations where the formal models of the system and requirements are not directly available. The approach involves integrating automaton learning algorithms and SCT (Supervisor Control Theory) of DES (Discrete Event Systems). First, the system is tested against the requirement. If the requirement is not fulfilled, the C∗ algorithm is used to infer a Moore automaton. Then, a supervisor is derived by SCT on the learned automaton and patched onto the system to correct any erroneous behavior. Finally, the controlled system is tested again to verify the correctness of the supervisor. This process is repeated until the requirement holds in the controlled system. The proposed approach is implemented automatically. Experiments are performed on the Brake-by-Wire (BBW) system, which is a hard real-time controller for automobiles, and a platooning program to demonstrate the feasibility of the approach for realistic systems.

**Bibliography**

H. Zhang, L. Feng and Z. Li, "Control of Black-Box Embedded Systems by Integrating Automaton Learning and Supervisory Control Theory of Discrete-Event Systems," in IEEE Transactions on Automation Science and Engineering, vol. 17, no. 1, pp. 361-374, Jan. 2020, doi: 10.1109/TASE.2019.2929563.

**Title**

Peak-Power-Aware Primary-Backup Technique for Efficient Fault-Tolerance in Multicore Embedded Systems.

**Abstract**

Multicore platforms offer task-level redundancy for fault-tolerance in embedded systems, but the Thermal Design Power (TDP) constraint limits the number of active cores. Violating TDP can trigger a performance throttling mechanism, which can impact the timeliness of the system. Only a few tasks can run in fully reliable mode under a given TDP constraint. To address this challenge, we propose a power-aware scheduling scheme for real-time tasks on core pairs in multicore systems. The proposed scheme removes overlaps of peak power of concurrently executing tasks to keep the power consumption below the chip-level TDP constraint. We use a task partitioning method with maximum-peak-power-first (MPPF) and maximum-peak-power-last (MPPL) policies to schedule original and redundant copies of tasks, respectively. Our experiments show that our technique provides up to 50% (on average by 29.5%) peak power reduction compared to state-of-the-art schemes, while providing the same reliability level.

**Introduction**

The development of technology has made it possible to integrate multiple cores onto a single chip to create advanced embedded systems. However, this process also increases the risk of hardware failure due to transient faults that are caused by high-energy particle strikes resulting in bit flips. To tackle this issue, redundant multithreading (RMT) and process level redundancy are used as fault-tolerant mechanisms against transient faults in multicore embedded systems. However, these mechanisms increase power consumption, which can exceed the chip Thermal Design Power (TDP) constraint and lead to system failure. TDP is the maximum power that a chip can safely dissipate, and Dynamic Thermal Management (DTM) is used to maintain the system temperature below a safe operating level. However, DTM may reduce performance and violate system timing constraints. This is especially problematic for real-time embedded systems that require strict timing control. In this article, we propose a peak-power-aware primary-backup technique scheme that effectively manages power consumption and timing constraints while using task-level redundancy on multicore systems. Our technique schedules real-time tasks on core pairs without violating their timing constraints.

**Methodology – Algorithm**

The proposed scheme for online task execution consists of two steps. Firstly, task sets are divided into appropriate core pairs based on their readiness for execution. Secondly, tasks are partitioned into execution parts and scheduled in a way that ensures the chip's TDP and the deadline constraint for the task set are not violated. To map tasks to cores, we select the core pair with the lowest utilization. This evenly distributes the workload between the cores. For task partitioning, method determine the greatest common divisor (GCD) of the execution time of the tasks, which is used as the partitioning time slot (PS). Tasks are then scheduled based on the maximum-peak-power-first (MPPF) and maximum-peak-power-last (MPPL) policies. Finally, if necessary, the tasks' execution parts are shifted throughout the schedule to ensure that peak power consumption remains below the chip's TDP.

**Conclusion**

A new technique called peak-power-aware primary-backup has been proposed to manage peak power consumption and ensure reliable task execution on multicore systems. The technique partitions original and redundant tasks, schedules them on core pairs to avoid violating timing constraints and minimizes peak power overlap using maximum-peak-power-first and maximum-peak-power-last policies. It cancels the second copy of fault-free tasks during execution to further reduce power consumption. The technique provides up to 50% peak power reduction compared to other schemes and enables task-level redundancy to improve reliability in multicore systems with tight power constraints. 

**Bibliography**

M. Ansari, M. Salehi, S. Safari, A. Ejlali and M. Shafique, "Peak-Power-Aware Primary-Backup Technique for Efficient Fault-Tolerance in Multicore Embedded Systems," in IEEE Access, vol. 8, pp. 142843-142857, 2020, doi: 10.1109/ACCESS.2020.3013721.