PhD Thesis

AN ERROR-LOCALIZATION, VALIDATION AND OPTIMIZATION TOOL FOR EMBEDDED CODE AUGMENTATION: AN ARCHITECTURE ORIENTED APPROACH

ABSTRACT

Embedded systems are usually designed for a single or a specified set of tasks. This specificity means the system design as well as its hardware/software development can be highly optimized. Embedded software must meet the requirements such as high reliability operation on resource-constrained platforms, real time constraints and rapid development. This necessitates the adoption of static machine codes analysis tools running on a host machine for the validation and optimization of embedded system codes, which can help meet all of these goals. This could significantly augment the software quality and is still a challenging field.

This dissertation contributes to an architecture oriented code validation, error localization and optimization technique assisting the embedded system designer in software debugging, to make it more effective at early detection of software bugs that are otherwise hard to detect, using the static analysis of machine codes. The focus of this work is to develop methods that automatically localize faults as well as optimize the code and thus improve the debugging process as well as quality of the code.

Validation is done with the help of rules of inferences formulated for the target processor. The rules govern the occurrence of illegitimate/out of place instructions and code sequences for executing the computational and integrated peripheral functions. The stipulated rules are encoded in prepositional logic formulae and their compliance is tested individually in all possible execution paths of the application programs. An incorrect sequence of machine code pattern is identified using slicing techniques on the control flow graph generated from the machine code.

An algorithm to assist the compiler to eliminate the redundant bank switching codes and decide on optimum data allocation to banked memory resulting in minimum number of bank switching codes in embedded system software is proposed. A relation matrix and a state transition diagram formed for the active memory bank state transition corresponding to each bank selection instruction is used for the detection of redundant codes. Instances of code redundancy based on the stipulated rules for the target processor are identified.

This validation and optimization tool can be integrated to the system development environment. It is a novel approach independent of compiler/assembler, applicable to a wide range of processors once appropriate rules are formulated. Program states are identified mainly with machine code pattern, which drastically reduces the state space creation contributing to an improved state-of-the-art model checking. Though the technique described is general, the implementation is architecture oriented, and hence the feasibility study is conducted on PIC16F87X microcontrollers. The proposed tool will be very useful in steering novices towards correct use of difficult microcontroller features in developing embedded systems.

PhD Thesis No: 12

MARIAMMA CHACKO