

# Static Program Analysis and its application to TI's DSP Software

Ramakrishnan Venkitaraman and Gopal Gupta

The “Texas Instruments TMS320 DSP Algorithm Standard” defines a set of requirements for DSP algorithms that, if followed, will allow system integrators to quickly assemble production quality systems from one or more algorithms. The standard aims to provide a framework that will enable the development of a rich set of Commercial Off-The-Shelf (COTS) components marketplace for the DSP algorithm technology that will significantly reduce the time to market for new DSP based products.

Static Analysis provides significant benefits and is increasingly recognized as a fundamental tool for analyzing programs. We develop a “*Static Analyzer*” for checking an algorithm (given only its binary code) for its compatibility with the “General Programming Rules” defined by the standard. The analyzer is used to detect hard coded addresses in programs. The analyzer takes object code as input, disassembles it, builds the flow-graph, and statically analyzes the flow-graph for the presence of dereferenced pointers that are hard coded. The analyzer can be used to validate DSP software for conformance with the TI TMS320 DSP Algorithm Standard's “General Programming Rules”.

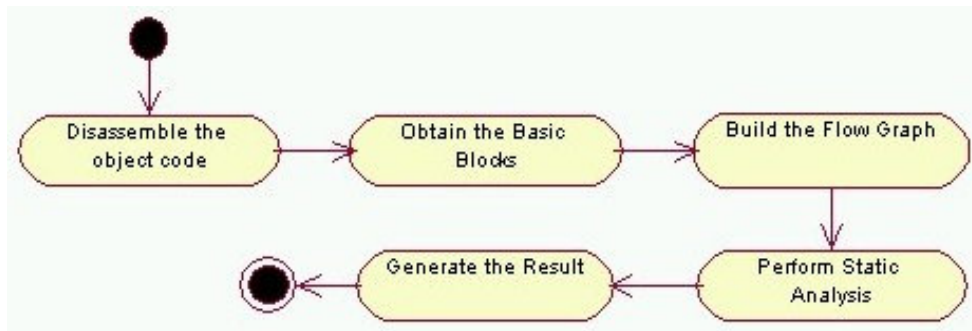


Figure: Activity Diagram for the Analyzer

To perform analysis, and check for standard compliance, we don't have access to the source code. So we disassemble the object code to get the corresponding Assembly Language Code. The analysis is done at the assembly language level and involves the formation of “*unsafe sets*” by statically analyzing the disassembled code. The code and the “unsafe sets” are checked for violations of the standard. In the case of recursive function calls and looping structures that are detected in the code, the analyzer automatically builds and populates the unsafe sets until a “*fixed point*” is reached.

The development and testing of the tool is currently in progress. Current work includes fine tuning the handling of loops and extending our system for the cover all the programming rules.

**Reference:** Ramakrishnan Venkitaraman and Gopal Gupta, “Static Program Analysis to Detect Hard Coded Addresses and its Application to TI's DSP Processor”, UTD Computer Science Department technical report (UTD CS-23-03).