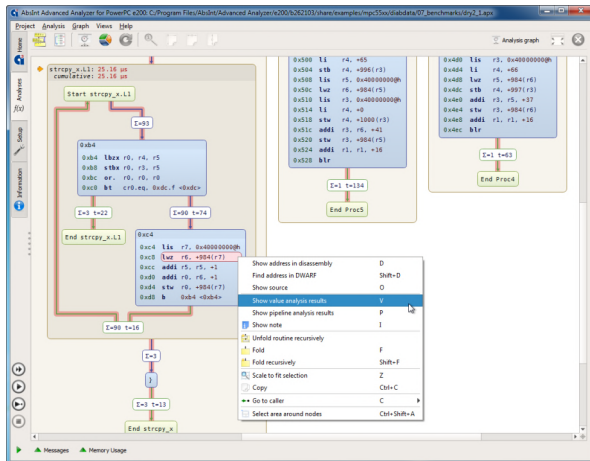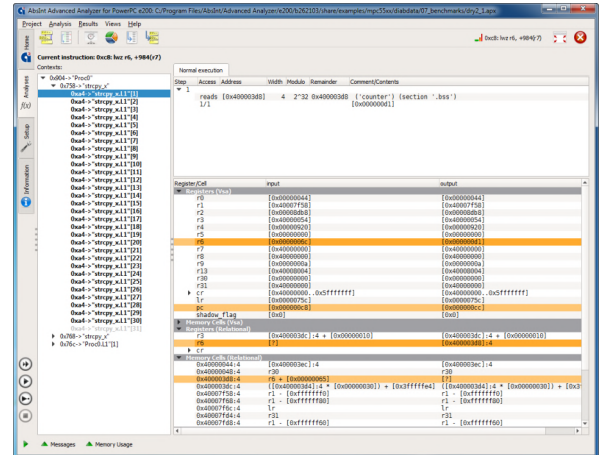# ValueAnalyzer

Static Value Analysis for Embedded Systems

ValueAnalyzer is a tool for static analysis of register and memory contents. It is particularly useful for:
- verifying the absence of illegal accesses to an embedded software system from within third-party modules provided as object code,
- initialization analysis for execution-time or stack-usage analyses, to improve precision and minimize manual annotations.

Visualization of Control Flow Graph...

... with value information for selected instruction.

Why do you need ValueAnalyzer?

- ValueAnalyzer is an **add-on** to aiT, StackAnalyzer, TimingProfiler, and TimeWeaver.

- In embedded control software, global data structures like (function) pointer variables are often initialized by separate initialization tasks. When a periodic or asynchronous task or runnable is analyzed for its worst-case timing or stack usage, such initializations are unknown.
  With ValueAnalyzer you can **automatically collect initialization data** for global variables and feed them as auto-generated annotations to a WCET, stack-usage, or another value analysis. Hence, it can **improve the precision** of aiT/StackAnalyzer/TimingProfiler/TimeWeaver analyses and **avoid manual annotations**.

- ValueAnalyzer helps to **detect illegal memory accesses** caused by programming errors, non-compliance with system specifications, or compiler/linker errors. It can verify **static user-defined assertions**, such as
  - assertions that a user-selected instruction or function does (not) access a user-selected memory area, or
  - global assertions on user-selected memory areas (is read-only, is write-only, is inaccessible, ...).

- ValueAnalyzer visualizes the call and control flow graph and allows to easily explore the analysis results. The contents of registers and memory cells are shown before and after each instruction execution context.

- ValueAnalyzer creates comprehensive reports for documentation and certification purposes, including:
  - all memory accesses per function
  - all calls to user-selected functions
  - all accesses to user-selected memory areas, register and memory contents at entry/exit of selected functions.

- ValueAnalyzer supports batch mode execution and integration in **continuous test and integration** frameworks.

Supported processors and compilers

ValueAnalyzer is available for all processor–compiler combinations supported by aiT, StackAnalyzer, TimingProfiler, and TimeWeaver.

AbsInt