Embedded Software Debug App

Shorten the time to find the root cause of both software bugs and hardware bugs found by software

Embedded software is used not only to deliver functionality on most of today's SoCs, it is also an important tool for software-driven verification. This growing software dependence creates demand for an efficient way to develop and debug these different types of software as well as any hardware bugs identified by the software. The Cadence® Embedded Software Debug App, part of the next-generation unified Indago™ Debug Platform, provides a software-centric environment to shorten the time to find the root cause of both software bugs and hardware bugs found by software.

Requirements for Embedded Software Debug

Since embedded software is developed for a range of needs, from software-driven verification to production operating systems and applications, embedded software debug must be available across a variety of platforms. It is important to consider both the performance and the model accuracy for both the development and application of the embedded software. For example, when developing software for verification of the hardware, it can be valuable to develop the software on a fast platform to improve developer productivity, but because the software is testing the hardware, it will ultimately need to be run on a fully accurate platform. These requirements are why the unified Embedded Software Debug App (Figure 1) is part of our Indago Debug Platform, offering a consistent debug environment across platforms and processor model abstractions.

In addition to the performance and accuracy tradeoff, another key characteristic is hardware visibility and hardware/software synchronization. Typically the level of functionality and visibility into software is good, but when performing a detailed debug

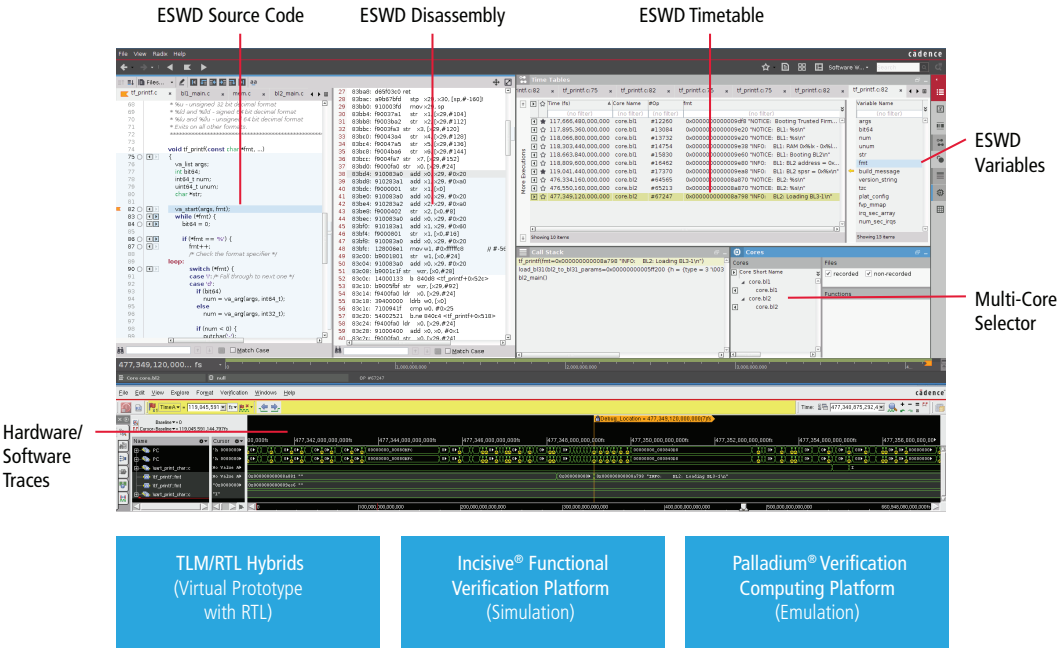

Figure 1: Embedded Software Debug App for the Indago Debug Platform

of hardware and software interaction, the critical factors are the visibility of the hardware activities and how the hardware activity is synchronized with the software. The Embedded Software Debug App uses post-process analysis of hardware and software trace information to provide the most accurate combined views of the hardware and software.

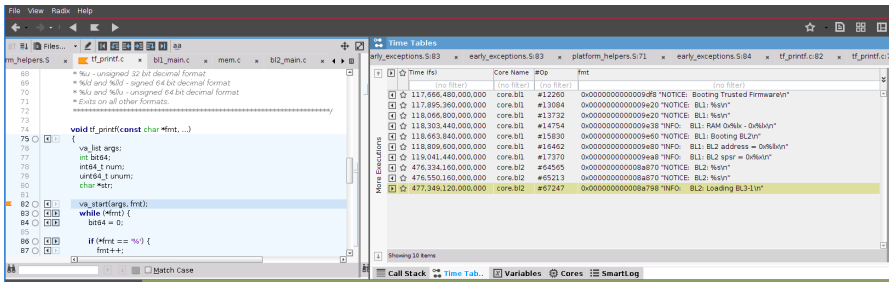

Figure 2: Timetable view of source code execution including variable of interest

Product Overview

Embedded software is an integral part of today's SoCs. Consequently, debugging embedded software as it will execute on the SoC and the ability to look at the interactions of the software with the hardware can save valuable time in finding and fixing problems. The Embedded Software Debug App leverages the core Indago technology to manage and visualize the "big data" of a complete multi-core/multi-cluster software trace per core to arrive at the root cause more quickly. The key is providing the complete context of the software execution, making it easier to locate the origin of a problem.

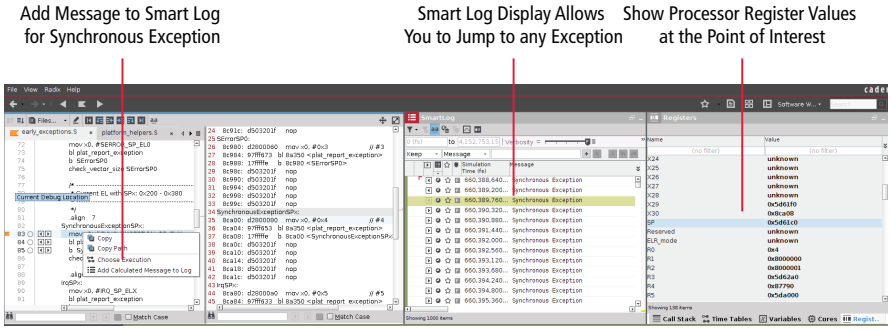

Figure 3: Calculated Message and Smart Log

For example, the Embedded Software Debug App offers a powerful timetable view where you can get a complete execution view for any line of source code. Figure 2 provides a timetable view of a print function, which shows all execution for all cores and all software images running on the cores, making it easy to jump to the point of interest and continue debugging.

Another powerful feature that saves time in debugging is the Embedded Software Debug App's Smart Log and calculated messages. This feature allows the user to describe a new printed message that will occur each time a line of code is executed. The screenshot in Figure 3 illustrates how this feature enables setup of a full context view of the software.

Of course, without the ability to mix the hardware view with the software, you only have half of the data for debug. In Figure 4, a screenshot from the Embedded Software Debug App shows the linkage between the hardware signal waveform display and the software. The screenshot also shows the Embedded Software Debug App's memory view.

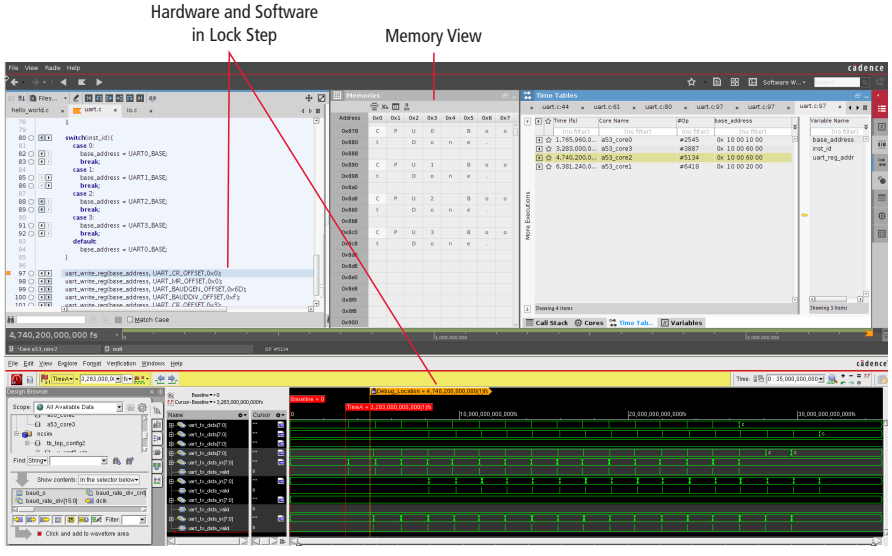

Figure 4: Embedded Software Debug App synchronized hardware/software view

The Embedded Software Debug App is simple to set up and use. An example flow for debugging a multi-core design is shown in Figure 5. There are three steps:

1. Compile and run
2. Generate database
3. Debug offline

Features

Software debug features

- C source
  - Go to previous or next execution
  - Breakpointing
- Assembly code synchronized with C source
- Multi-core register view
- Call stack
- Variable window
- Memory view
- Smart Log with calculated messages
- Timetable window
- Waveform view of software functions and variables

Synchronized hardware and software

- Step forward or backward in software
- Jump to next signal transition
- Debug collaboration
- Make notes and set stars on key debug views

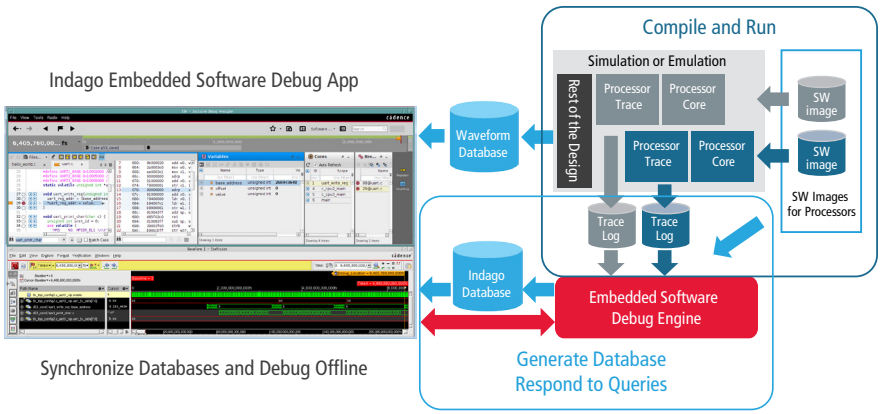

Figure 5: Example debug flow

- Share debug session including notes and stars

Processor core support

- Multiple architectures and families supported
  - Contact Cadence for latest list of supported cores
- Program counter trace support for source-level trace on any core
- Integration kit for full software debug feature support on proprietary cores

Related Products

- All Indago Debug Platform apps

Cadence Services and Support

- Cadence application engineers can answer your technical questions by telephone, email, or Internet—they can also provide technical assistance and custom training
- Cadence certified instructors teach more than 70 courses and bring their real-world experience into the classroom
- More than 25 Internet Learning Series (iLS) online courses allow you the flexibility of training at your own computer via the Internet
- Cadence Online Support gives you 24x7 online access to a knowledgebase of the latest solutions, technical documentation, software downloads, and more

cadence®

Cadence Design Systems enables global electronic design innovation and plays an essential role in the creation of today's electronics. Customers use Cadence software, hardware, IP, and expertise to design and verify today's mobile, cloud, and connectivity applications. www.cadence.com