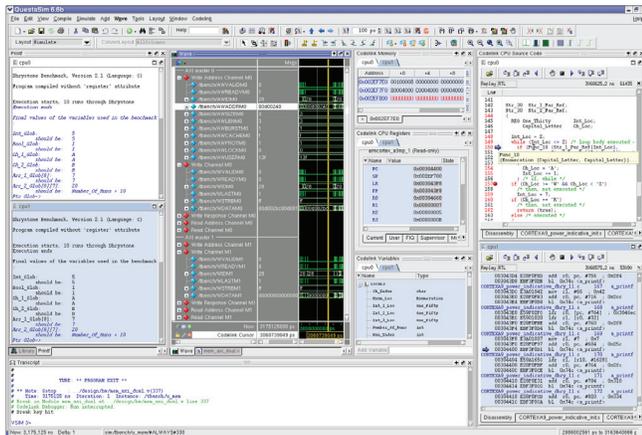


Questa® Codelink



Questa® Codelink provides full processor(s) debug visibility for more efficient SoC debugging and verification.

Overview

Today's SoC projects include embedded processors and must deliver working embedded software and hardware. Integrating embedded software with hardware in HDL simulation is challenging because of the lack of visibility into the processor(s) in the SoC. Existing HDL verification environments provide visibility into the HDL source code along with hardware waveforms, but lack similar capabilities for embedded software code and the state of the processor(s). Questa® Codelink delivers increased debug productivity for the hardware verification engineer, increased project quality, and reduced project risk using the processor or processors in the design.

Codelink provides full visibility of your ARM, MIPS, and IBM PowerPC processors while running your SoC design using an HDL logic simulator. It observes the behavior of the processor modeled at RTL or DSM (Design Simulation Model) level and provides a rich debug environment. No design, processor model or embedded software changes are required. Codelink is non-intrusive and has no impact on simulation results. The simulation runtime overhead is less than 1%. Since the processor models are provided by IP vendors, the simulation precision is 100% cycle accurate. Two modes of operations are supported: Live and Replay. The Live mode provides visibility and control of the processors and embedded software while running HDL simulation. During Replay mode Codelink reads a processor execution trace data file so you can debug the design post-simulation. In both modes software code and hardware waveforms are fully synchronized.

Hardware Defect Diagnosis

HDL logic simulators are not designed for debugging embedded CPUs. Debugging a full functional processor model when it goes into "flatline" is a challenging task. The design simulation models for processors provide instruction trace files (EIS – Executed Instruction Stream, Tarmac – Trace ARM Accesses, MIPS – trace, etc.) which include a list of fatal errors while the waveform shows no activity on the processor pins for a number of clock cycles. Relying on these two simulation outputs exclusively can be tedious.

Major product features:

- ARM, MIPS, and IBM PowerPC processor debug visibility in HDL logic simulations
- Multi-core support
- Step forward or backward through source or assembly
- Source, assembly, memory, variable, register and call-stack views
- Four-level logic (0, 1, U and X) in the SW debug windows
- Live and Replay mode for interactive post-simulation debug
- SW source code and HW waveforms are fully synchronized
- Connects to existing Sign-Off processor models without SW or HW changes
- Zero impact to simulation results providing 100% cycle accuracy

Major product benefits:

- Radically reduces debug time for tests using processors
- Replay overnight batch simulation in seconds
- Debug multi-core synchronization errors

Codelink increases the debug productivity of SoC verification by monitoring the register changes of sign-off processor models from ARM, MIPS, and IBM PowerPC and adding a software source viewer into the HDL simulation environment. Both source and assembly level embedded software are supported. Visibility into software variables, processor registers, memory and stack is provided and they update dynamically. The hardware waveform viewer cursor is synchronized with the software source window thus the wave cursor tracks your progress as you step through the code. Likewise, dragging the waveform cursor will scroll the source view to the corresponding source line containing the instruction being executed. The breakpoint mechanism is supported as well.

Existing software debuggers lack comprehensive representation of X-states that are too common in HDL logic simulation. Codelink allows X-states to appear in register and memory windows. The memory viewer provided by your HDL logic simulator shows only physical addresses. When simulating with local CPU caches turned on, some of the memory transactions can be hidden. On the other hand, turning off CPU caches slows down the entire simulation performance. The Codelink memory viewer avoids these limitations by allowing you to see virtual addresses directly supporting both cache and memory management units.

Interactive Post-Simulation Debug

There is no need to re-run failing regression tests in order to isolate the case of the failure. During simulation, Codelink observes changes in the processor's general purpose registers and memories then saves the information to a replay file. After simulation, Codelink loads the saved software replay file in conjunction with the waveform file. The mechanism is highly interactive allowing you to move instantly to any point in the simulation record. The result: full hardware/software debug visibility in a blink of an eye.

Step Through Code in Reverse

Codelink has the ability to un-execute software code so you can step backward through your code. As you step, a cursor in the hardware waveform window simultaneously tracks the instruction execution.

Multi-Core Support

For multi-core designs, Codelink simultaneously monitors multiple processors. Any mix of supported processors can be logged during a single simulation. The Codelink graphical user interface presents source, memory, register, variable, and call stack windows for each core in the design. The user can choose tabbed, tiled, vertical, or horizontal window orientation.

Standard SW Output Console Support

The software *printf* function sends a message to a standard IO which then prints it in the UART or to the logic simulator terminal window consuming multiple simulation clock cycles. The Codelink standard output window allows displaying software *printf* messages consuming almost 0 clock cycles.

<i>Supported processors</i>
ARM Cortex-A Series: A5, A8, A9, A9 MP, A15
ARM Cortex-R Series: R4, R4F, R5
ARM Cortex-M Series: M0, M3, M4
Classic Processors: ARM7, ARM9, ARM11
MIPS Cores: MIPS 4KE, 4K, 24K, 24KE, 74K
IBM Cores: PPC405, 440, 464FP, PPC750
<i>Supported platforms and logic simulators</i>
Red Hat Enterprise Linux 4 for 32-bit and 64-bit
Red Hat Enterprise Linux 5 for 32-bit and 64-bit
Mentor Graphics ModelSim®, Questa®
*Cadence Incisive Enterprise Simulator (logging processor states only)
*Synopsys VCS (logging processor states only)
*ModelSim/Quarta viewer license required for interactive post-simulation debug
<i>Supported Third Party Tools</i>
Hardware debugger: SpringSoft Verdi Automated System
Software debugger: ARM Workbench IDE in RealView Development Suite

For the latest product information, call us or visit: www.mentor.com/codelink

©2011 Mentor Graphics Corporation, all rights reserved. This document contains information that is proprietary to Mentor Graphics Corporation and may be duplicated in whole or in part by the original recipient for internal business purposes only, provided that this entire notice appears in all copies. In accepting this document, the recipient agrees to make every reasonable effort to prevent unauthorized use of this information. All trademarks mentioned in this document are the trademarks of their respective owners.

Corporate Headquarters

Mentor Graphics Corporation
8005 SW Boeckman Road
Wilsonville, OR 97070-7777
Phone: 503.685.7000
Fax: 503.685.1204

Sales and Product Information

Phone: 800.547.3000
sales_info@mentor.com

Silicon Valley

Mentor Graphics Corporation
46871 Bayside Parkway
Fremont, CA 94538 USA
Phone: 510.354.7400
Fax: 510.354.7467

North American Support Center

Phone: 800.547.4303

Europe

Mentor Graphics
Deutschland GmbH
Arnulfstrasse 201
80634 Munich
Germany
Phone: +49.89.57096.0
Fax: +49.89.57096.400

Pacific Rim

Mentor Graphics (Taiwan)
Room 1001, 10F
International Trade Building
No. 333, Section 1, Keelung Road
Taipei, Taiwan, ROC
Phone: 886.2.87252000
Fax: 886.2.27576027

Japan

Mentor Graphics Japan Co., Ltd.
Gotenyama Garden
7-35, Kita-Shinagawa 4-chome
Shinagawa-Ku, Tokyo 140-0001
Japan
Phone: +81.3.5488.3033
Fax: +81.3.5488.3004



MGC 05-11 1028250-w