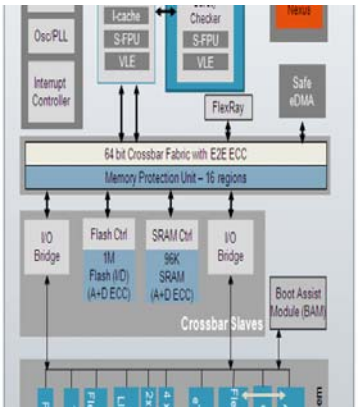
Inside Nexus

Volume 1, Issue 1, Nov 2014

Welcome to the 1st issue of the Nexus 5001 Forum's quarterly newsletter "Inside Nexus." This newsletter provides forum members and interested parties from industry and academia updates to the forum's activities related to the Nexus 5001 standard (2012), advancing the tool set, and promoting its market acceptance.

Updates/revisions to the standard are being accepted and reviewed by the forum's Technical Committee. If you have any suggestions/updates and you would like the committee to consider, please email tech@nexus5001.org. Thank you.



Block Diagram and Features

Nexus Capstone University Project

The Nexus 5001 Forum has partnered with the University of Washington Bothell as part of their Nexus Capstone Development project to further instrumentation technology solutions using the Nexus 5001 standard.

The project provides university students with industry resources and mentoring in support of doing research and development in implementing debug and trace solutions.

Among the goals of the project are to:

a. Develop the university expertise in understanding the architectures and trade-offs in adding trace and other on-chip instrumentation to open-source processors

b. Develop and integrate license-free example blocks that generate Nexus processor and SoC trace information; and

c. Model and integrate debug information across different interfaces supported by the Nexus standard.

Additionally, University of Washington Bothell students will be developing system level descriptions and building blocks of Nexus components, such as XML descriptions of the Nexus message, as future parts of this Capstone project.

This project gives Nexus members an opportunity to mentor students in the industry, and to develop a better understanding at the university level of technologies and challenges in on-chip instrumentation, as well as help further the development and support of the Nexus 5001 standard.

(continued on next page)

Member Spotlight

PLS is among the world's leading suppliers of software debugging solutions and complete development tools for the 16 Bit and 32 Bit microcontroller families.

pls Development Tools

PLS' Universal Emulation Configurator opens up the full potential of emulation devices from Freescale and STMicroelectronics

The **Universal Emulation Configurator** (UEC) from PLS Programmierbare Logik & Systeme is now also available for the emulation devices MPC57xx from Freescale and SPC57x from STMicroelectronics. With the help of this special tool for definition of trace and measurement tasks for on-chip emulation logic, the full potential of the emulation devices can be used for the first time without any limitations for troubleshooting and software quality assurance.

The MPC57xx and SPC57x emulation devices are pin-compatible to their respective production chips, but include additional emulation memory, extensive trigger and filter logic as well as connections for a serial high-speed interface based on the Aurora protocol. In order that developers can easily as possible and abstractly configure the several hundred registers of the additional emulation memory, the **Universal Emulation** Configurator is based on a three-stage programming model. The assembler-like **Trace Qualification Language** (TQL) of the first stage uses the resources of the emulation hardware. In this way, the individual register values can be set. The C-like **High-Level Trace Qualification Language** (HTQL) of the second stage already allows a more abstract description of measurement tasks by conditional actions and definitions of state machines. The third stage of the abstraction and the actual user interface is formed by a graphical editor, with which a measurement task can be put together from predefined blocks. In doing so, specific states in the target are described by signals. These, in turn, can initiate actions or shift an underlying state machine into a new state. The individual blocks, which serve to describe signals, actions and basic elements of state machines, are in turn grouped together in libraries. These can be extended as required or supplemented with own libraries. In order to achieve the optimal level of modularity, flexibility and user-friendliness, Extensible Markup Language (XML) was chosen as data format. Analysis tasks, which were created on the basis of the library elements, can also be saved in XML format for later reuse.

For a single block, its appearance in the editor, the parameter for adaptation to the respective measurement task and a template of the Hyper-Text Query Language (HTQL) code fragment to be generated are described. It is thus possible to make any HTQL construct also available as a graphical element.

The **Universal Emulation Configurator** helps the user to cope as effectively as possible with the limited resources of the on-chip emulation memory. In parallel to this, the implemented Aurora interface offers the possibility to externally record a very large amount of trace data and to carry out a statistical analysis of the program flow such as code coverage and profiling. PLS' Universal Access Device 3+ (UAD3+) with Aurora pod serves for recording, while the evaluation itself is carried out by the Universal Debug Engine (UDE).

Nexus 5001 Forum Webinar Program

(continued on next page)

Nexus 5001 Forum partners with its members to deliver industry webinars. These webinars are free to attend. To see prior webinars, please click here or visit www.nexus5001.org.

The Marketing Committee is looking for additional topics for future webinars. If you have an idea for a topic, please contact marketing@nexus5001.org.

Our next scheduled webinar is planning for Q1 of 2015. Stay tuned for further details.

Sleep better with Nexus 5001

By Neal Stollon, Ph.D, P.E.

Probably more often than most release managers will admit, they have the nightmare. Their product has shipped and all looks fine, but what if issues crop up in the field. Did they add enough instrumentation and debug features to the design to quickly get to roots of the problem … or did they acquiesce to their designers too quickly on the arguments that "it is just wasting a thousand gates" or "the simulations looked perfect, so the product must be as well". Experienced managers know in their hearts that regardless of the quality efforts and attention to design, problems show up when the final verification happens. Usually the issues are found in the lab, but sometimes, just enough to keep you awake at night, problems are discovered after that. In the field, far from all the lab equipment and test and debug infrastructure that is needed to get some visibility into what the heck is going on. Recent studies have indicated that verification and validation of a design can be half of the overall design effort in many systems, and debug of the final product, with all of its corner cases and "what if " modes of operation, can account for half of that effort. So is there a way to sleep better at night? …and to reduce the tedious and time-consuming debug activities that are part of product release. There are many schools on this topic. The simplest answer for many is make sure your design is well instrumented, make sure there are features that allow access to those deeply embedded processors and buses, interfaces that allow you to set up controllable and observable modes of operation, regardless of how your product is implemented and integrated into the bigger systems. Effective tools and capabilities for debug, calibration, and performance analysis are widely recognized as key elements in ensuring quality products. There are many different means to this end, one that has been proven in use over the last decade is Nexus 5001.

To be sure, there are different instrumentation and debug solutions available for different types of products. In the processor IP space, ARM has Coresight, and MIPS has PDTrace, in the FPGA space, Xilinx has ChipScope and Altera has Signal Tap, just to touch the surface. Intel, AMD, and others have their own proprietary debug environments. There are all solutions with many virtues, however they are all solutions focused on one aspect of the problem; the particular IP or chip that the companies sell. The system level of debug and integration of different types of instruments, well that is your problem. Especially if your product has a bigger scope, be it a SoC with different IP from multiple vendors or a board or rack with heterogeneous configurations of chips, debug solutions constrained to a single part of your product are going to be limiting. Trying to integrate debug information from these various and different sources is a daunting task. And this is where Nexus 5001 comes into the fore. Originally developed over decade ago, to support the debug and calibration needs of the automotive industry, Nexus 5001 has improved and expanded over a number of specification releases with features to address instrumentation needs of telecom, networking and related domains and to support a range of instrumentation and debug interfaces required in diverse systems.

To explain Nexus 5001 in brief- Nexus 5001 is an IEEE-ISTO industry standard processor and system instrumentation and debug interface, which was developed to be product agnostic in its support of key debug features for instruction and data trace, memory and register access and replacement, bus and port substitution and emulation, and other features required for systems debug. Over the last decade, Nexus 5001 has been ported to or integrated into over 15 processor architectures for different products. It provides a vendor independent protocol, logical infrastructure, and standard interfaces that are scalable to address a wide variety of digital measurement, calibration, and analysis features and capabilities. Nexus 5001 allows the straightforward integration of custom instruments that may be useful for your product analysis. It retains compatibility with other industry standards, including JTAG 1149.1 and 1149.7, while supporting high bandwidth interfaces needed for extensive trace and analysis of complex systems. Nexus 5001 applications are supported by leading tools venders in both the automotive and embedded space. This editorial space is too small to go into any detail on all the features that Nexus offers to improving product and system debug, calibration, and analysis, but stop by the http://www.nexus5001.org site or drop me a line at nstollon@nexus5001.org for more information. You may be sleep better as a result.

He can be reached at neals@hdldynamics.com.

(continued on next page)

Nexus 5001 Forum Members

Nexus 5001 Forum's University Partners:

University of Washington

McMaster University

University of Alabama Huntsville

Technische Uniersitat Dresden

Interested in Nexus 5001 Forum membership, please send an email to admin@nexus5001.org.

To remove your name from our mailing list, please click here.

Questions or comments? E-mail us at admin@nexus5001.org or call us at 732-465-5895

###