A PROGRAM OF THE IEEE INDUSTRY STANDARDS AND TECHNOLOGY ORGANIZATION

The NEXUS Debug Standard: Gateway to the Embedded Systems of the Future

Ashling Product Brief APB179

Ashling Microsystems, Inc.
18612 Devon Avenue
Saratoga, CA 95070
Tel: (408) 884 3020
Fax: (408) 884 3026
Email: support.usa@ashling.com

ASHLING
THE DEVELOPMENT SYSTEMS COMPANY

Ashling Microsystems Inc.
18612 Devon Avenue
Saratoga
CA 95070
Tel: (408) 884 3020
Fax: (408) 884 3026
Email: sales.usa@ashling.com

Ashling Microsystèmes
2, rue Alexis de Tocqueville
Parc de Haute Technologie
92183 Antony Cedex, France
Tel: 01.46.66.27.50
Fax: 01.46.74.99.88
sales.fr@ashling.com

Ashling Microsystems Ltd
Intec 2, Wade Road
Basingstoke
Hants. RG24 8NE, U.K.
Tel: (01256) 811998
Fax: (01256) 811761
sales.uk@ashling.com

Ashling Microsystems Ltd
National Technological Park
Limerick
Ireland
Tel: +353-61-334466
Fax: +353-61-334477
sales.ie@ashling.com

APB179-V5U-NexusBooklet

The NEXUS Debug Standard: Gateway to the Embedded Systems of the Future

By Hugh O'Keeffe, Ashling Microsystems Ltd.
Hugh.OKeeffe@ashling.com

References

1. "IEEE-ISTO 5001™-2003, the Nexus 5001 Forum™ Standard for a Global Embedded Processor Debug Interface". Version 2, 23rd. December 2003. Available at www.nexus5001.org

Contents

1 Introduction and Summary

IEEE-ISTO 5001™-2003, the Nexus 5001 Forum™ Standard, previously known as GEPDIS (Global Embedded Processor Debug Interface Standard) or NEXUS, was created to provide a standard debug interface for embedded control applications. This debug interface is used to connect tools to an embedded system to facilitate any of the following:

- Run-time control (debugging)
- Code execution trace capture and Data access trace capture
- Calibration (Data access on-the-fly)
- Logic Analysis
- Rapid-prototyping

The standard builds on the experience of a wide range of semiconductor vendors, development system manufacturers and embedded systems design engineers (see www.nexus5001.org for a full list of Nexus 5001 Forum™ members). The Nexus 5001 Forum™, which developed the standard, undertook the task of defining a common set of microprocessor on-chip debug features, protocols and interfaces for the development and debugging tools that are used by embedded systems developers.

Although the Nexus 5001 Forum™ originally targeted the requirements of the automotive industry, the group has produced a scaleable standard that addresses the needs of high-performance embedded microprocessors in all industry segments. Scaleability allows tool vendors such as Ashling Microsystems to produce a family of tools with a range of performance levels and cost that can answer the needs of development project teams right through the design, development, debug, calibration and verification cycle.

2 Why On-Chip debugging?

On-chip debugging support logic is required in today's (and tomorrow's) microprocessors because without such support logic it would be impossible or prohibitively expensive to produce debugging tools capable of supporting increasing clock speeds. Some of the main reasons for this are:

- The use of on-chip Flash memory for program storage make it difficult or impossible for external tools (Logic Analyzers or In-Circuit Emulators, for example) to determine the actual instruction being currently executed as there is no external or off-chip visibility of the program address bus on such “Single-Chip” microprocessors.
- Very high microprocessor clock frequencies mean that an emulator cannot halt target-program execution *before* it executes a particular instruction of interest, because the available timing budget is simply too short.

- Deep instruction pipelines, multiple-issue RISC architectures and on-chip caches can make it very difficult to determine which instructions were fetched and which were actually executed.
- The introduction of application-specific or customer-specific Systems-On-a Chip (SOCs), some with multiple processors, means that a uniform, reliable debugging interface is essential to reduce the task of redesigning the Microprocessor Emulator for each individual SOC design.
- Product development times are shortening and engineering teams are under more and more pressure to deliver quicker. Many of today’s products have a window of sales opportunity in the market; missing this window due to development slippage can mean that the market for that product is lost. Debug and Verification is a significant part of the development cycle; hence, an effective debug mechanism is vital.

3 The IEEE-ISTO 5001™-2003 (NEXUS) feature set

The IEEE-ISTO 5001™-2003 (NEXUS) feature set is based on today’s best on-chip debug implementations. Its goal is to create the best possible debug feature set while minimizing the required pin-count and die area. The standard is designed to be processor- and architecture-independent and to support multi-core or multi-processor designs. Physically, IEEE-ISTO 5001™-2003 defines a standard set of connectors for connecting the debug tool to the target or system under test. Logically, data is transferred using a packet-based protocol. This protocol can be IEEE JTAG 1149.1; or, for high-speed systems, an auxiliary port can be used that supports full duplex, higher bandwidth transfers.

This section summarizes the NEXUS feature set.

3.1 Run-time control

This is the most basic feature and is standard on all on-chip debug implementations. It allows the debug tool to start and stop the processor, to modify registers and to single-step (execute a single assembly instruction).

3.2 Memory Access

NEXUS supports memory access while the processor is running (“on-the-fly”). This allows the debug tool to read and write memory while the processor is running without affecting or intruding on the currently executing instructions. On-the-fly access is a very powerful feature and is a requirement for debugging truly real-time systems were it is not possible to halt the system under test. In particular, Data Memory Access is essential for the Calibration of Engine Control Units.

3.3 Breakpoints

NEXUS provides support for breakpoints that halt the program when a specified event (the breakpoint) has occurred. The event can be specified as code execution at a specified address or as a data access (read or write) to a specified address with a specified value (for example, break when 0x55 is written to address 0x123456; break when 0xAA is read from 0xFFFF00). NEXUS breakpoints are similar to the hardware breakpoints found in other processor architectures and can be set in Flash or ROM memory. AWatchpoint is a similar concept; however, when a watchpoint occurs a message is sent to the debug tool (as opposed to halting the processor).

3.4 Instruction or Program Trace

NEXUS uses the Branch-Trace technique to compress the information needed to trace program execution. Executed code address information is emitted via the NEXUS port at branch or exception instructions only; the debug tool interpolates the program trace for sequential (non-branch) instructions from a local image of code memory contents. This allows full reconstruction of the program flow by the debug tool.

3.5 Data Trace

This feature allows the debug tool to track real-time data accesses to memory locations. The trace can be qualified be specifying a specific range (start and stop address) and a specific access type (read or write).

3.6 Ownership Trace

The Ownership Trace feature allows a real-time operating system (RTOS) to identify the currently executing process or task to the debug tool. The RTOS simply writes to a predefined NEXUS register when switching tasks; this write forces an Ownership Trace message to be emitted from the NEXUS port. The message will contain an ID that identifies the Task or Process to the debug tool.

3.7 Memory Substitution and Port Replacement

This feature allows internal memory or port accesses to be implemented over the auxiliary NEXUS port. For example, this feature can be used to implement ROM patching; that is, instead of reading on-chip ROM, the instruction is fetched from the debug tool via the auxiliary port. Port replacement is useful when relatively slow I/O pins have a secondary function, for example, an NEXUS port function; it allows the debug tool to 'emulate' their primary functionality.

3.8 Data Acquisition

This feature was added to support Rapid Prototyping; it allows the rapid transfer of arbitrarily large amounts of data via the auxiliary port to the debug tools. Its uses a more efficient protocol than that used in Data Trace. It is also of major importance for Calibration in Automotive applications.

3.9 Software API

This is a low-level Application Programming Interface (API) that 'hides' the target specifics such as host connection (such as an Emulator or Calibration-instrument) mechanism and processor specific NEXUS register details. This API is produced jointly by the tool and semiconductor vendor.

4 The NEXUS compliance classes

IEEE-ISTO 5001™-2003 is a scalable standard; there are currently four classes of compliance to the standard, ranging from the basic (JTAG-only) Class 1 up to Class 4.

4.1 Class 1

This class supports Run Time Control (run, stop, memory upload/download when the processor is halted, breakpoints, read or set registers) using the JTAG interface. Communications are half duplex only and bandwidth is limited. Trace is not supported.

4.2 Class 2

This class adds ownership trace and program trace and allows the auxiliary debugging port to be shared with "slow" I/O port pins. Ownership trace allows current task or current process trace for systems based on real-time kernels or operating-systems.

4.3 Class 3

This class adds data write trace and memory read/write on-the-fly without halting execution. Data read/write tracing, sharing of the Auxiliary port with High Speed I/O ports such as the address/data bus, and support for data acquisition (visibility of related data parameters stored in internal resources, typically related calibration variables) may also be optionally part of Class 3 compliance.

4.4 Class 4

The class adds memory substitution (fetching or reading data over the NEXUS auxiliary port) and allows tracing to be triggered by a watchpoint. Triggering memory substitution on a watchpoint is an optional feature of Class 4 compliance.

The main characteristics of each Class are summarized in Table 1.

NEXUS Class	Class 1	Class 2	Class 3	Class 4
Trace features	Trace not supported	Adds ownership trace and program trace via Auxiliary ports	Adds data write trace and read/write memory on-the-fly via Auxiliary ports	Allows tracing to be triggered by a watchpoint via Auxiliary ports
Debug communication method	Half-duplex communication (Limited bandwidth)	Allows full-duplex communication may be full duplex using Auxiliary port (higher bandwidth)	Allows full-duplex communication may be full duplex using Auxiliary port (higher bandwidth)	Allows full-duplex communication may be full duplex using Auxiliary port (higher bandwidth)
Run-time control	Supports run time control features using JTAG interface	Supports run time control features using JTAG interface or Auxiliary port	Supports run time control features using JTAG interface or Auxiliary port	Supports run time control features using JTAG interface or Auxiliary port
Auxiliary Port Implementation	No Auxiliary Port	Allows Port sharing; the Auxiliary port may be shared with slow IO port pins (such as static Config pins that are latched at reset time).	Allows Port sharing with high-speed I/O ports.	Allows Port sharing with high-speed I/O ports.
Data acquisition	Not supported	Not supported	Supports data acquisition	Supports data acquisition
Memory Substitution	Not supported	Not supported	Not supported	Fetch or read data via NEXUS auxiliary port. Option: Trigger memory substitution on a watchpoint

Table 1: NEXUS (IEEE-ISTO 5001™-2003) Debug Class Characteristics

Note that 'optionally' in the context of the NEXUS specification means that a processor of a specific class does not have to support the optional features to be compliant with that class.

5 IEEE-ISTO 5001™-2003: What's currently happening?

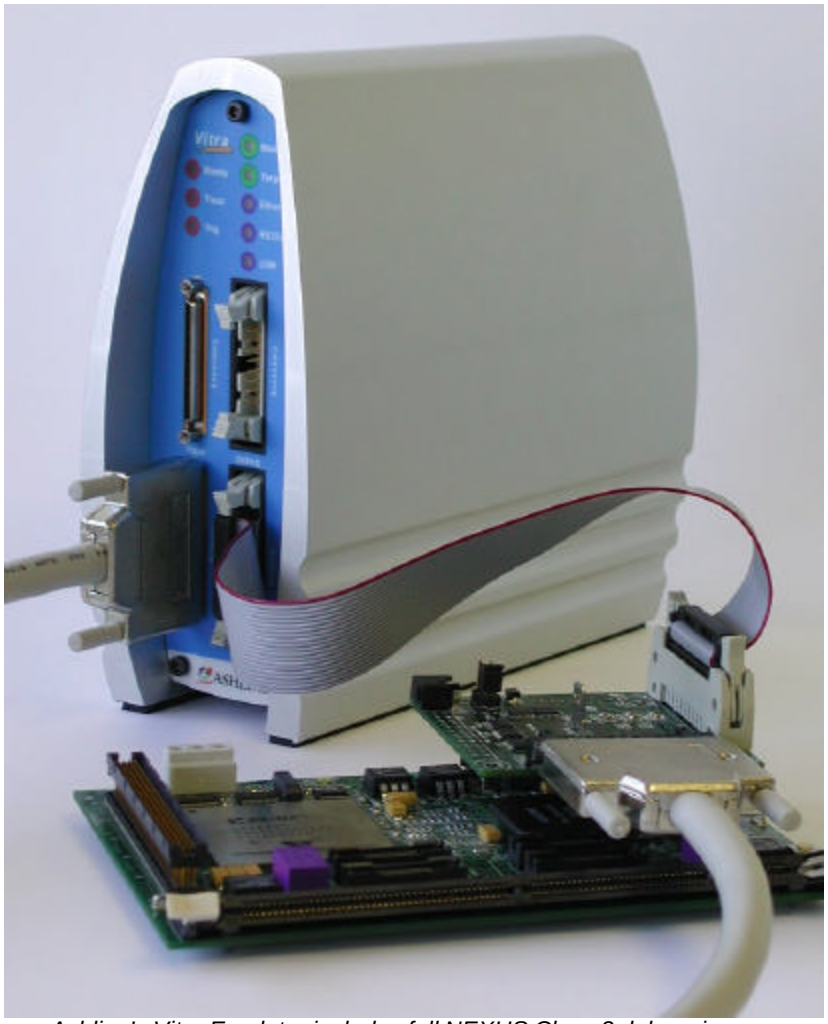
IEEE-ISTO 5001™ has been an official IEEE Industry Standards and Technology organization (IEEE-ISTO) standard since 1999; see their website at www.ieee-isto.org for more details. Formed in January 1999, the IEEE-ISTO is a not-for-profit corporation offering standards-related industry groups an innovative and flexible operational forum and support services. The IEEE-ISTO facilitates the activities that support the implementation and acceptance of standards in the marketplace. The second update to the Nexus standard, IEEE-ISTO 5001™-2003, the Nexus 5001 Forum™ Standard for a Global Embedded Processor Debug Interface is now available for download from the Forum's website at www.nexus5001.org

Ashling Microsystems, as a member of the Nexus 5001 Forum™, brings to the initiative its expertise in on-chip debug based debugging techniques and Software Quality Assurance tools such as Performance Analyzers and Code Coverage tools.

Through close co-operation with Freescale Semiconductors, Ashling has introduced a scaleable, universal range of tools that offer full support for the Freescale PowerPC MPC5500 and MPC565 families of automotive embedded processors. The MPC565 was the world's first device to feature a full Class 3 Nexus debug and calibration port. Further details on Ashling's full range of NEXUS tools are available at www.ashling.com

Silicon vendors including Freescale Semiconductors, ST and National Semiconductor have already introduced several families of embedded microprocessors with Nexus debug interfaces.

Successful implementation of both silicon and tools has allowed the NEXUS forum to 'work out the bugs' in the original specification, resulting in publication of Version 2 of the Standard in 2003. This revision addresses any ambiguities or errors in the original; in addition, it extends the connector options and adds improved support for high-speed tracing.



Ashling's Vitra Emulator includes full NEXUS Class 3 debugging, instruction trace and data read/write

Based on the experience gained by successfully applying the NEXUS debug standard to a variety of microprocessors, the member-companies of the NEXUS Forum – comprising semiconductor vendors, tools vendors and users – are ensuring that the updated Standard will further widen its range of applications.

6 Conclusions

The NEXUS standard offers many benefits to the Embedded Systems industry, including the developers of such systems, the semiconductor vendors and the development tool vendors. Silicon vendors now have a standard debug port design that can be used across different architectures; in addition, this debug port is known and easily supported by tool vendors. Tool vendors can now design universal tools that work with silicon from different vendors thus increasing the market for their tools. Finally, end users are now assured of a standard, scaleable toolset that covers all their needs and is available with first silicon.