Nexus 5001 - Instrumentation architectures and the new Debug Specification

Neal Stollon, HDL Dynamics
Chairman, Nexus 5001™ Forum

neals@hdldynamics.com
nstownon@nexus5001.org

HDL Dynamics
SoC Solutions

www.nexus5001.org
SoC Debug – Multicore and systems

Impact of Multicore Processors on System Debug

• Traditional run control become more complex – synchronized start/stop
• Growing need for more comprehensive software instrumentation
  – Higher level of visibility for software debug
  – Tools for performance evaluation
• Single chip multicore presents additional challenges
  – High integration of cores and peripherals
  – Reduced visibility using traditional logic analyzer tools
• Applications are typically tightly coupled
  – Concurrency problems are more likely
  – Sharing on-chip peripheral is essential
  – Correlation of code running on different cores / SoC interactions

► Debugging software is the most expensive software task today
► Debugging tools must evolve into more powerful solutions

www.nexus5001.org
SoC Debug – Current Alternatives

► Build From Scratch

► Licensing

► Standards Based
  • Infrastructure – 3rd party community for probes, SW, interfaces
  • Overhead – not tied to particular architectures
  • Innovation – Updates are periodic, slower but more comprehensive
  • ROI – more work than licensing, less than build your own –

► Standards may be required for end applications
  • - Automotive, Networking, Aerospace

• Which bring us to Nexus 5001
  • IEEE-ISTO 5001 is leading on chip instrumentation/debug standard
Why Nexus 5001

Real Time Debug Instrumentation Architecture and interface standard
- IEEE-ISTO 5001 Standard – ISTO Industry organization – 20 member organizations
- CPU/SoC architecture agnostic standard (15 different architectures to date)
- Default standard use in US Automotive electronics
- Support from range of vendors in the tools community
- Aligned with other standards bodies - 1149.1, 1149.7, MIPI, Power.org, OCP-IP
- Nexus Specification is freely available – over 400 downloads in last year

Nexus provides a Instrumentation toolbox for SoC Debug
- Predefined/ User defined Debug packet messages, application registers
- Support for levels of increasing debug functionality
  - Embedded run control, Breakpoints, Instruction/data trace
  - Memory and Register configuration and system analysis access
  - Defines Multiple Trace and Debug Access Methods and interfaces
  - JTAG & Parallel AUX. Read (Trace) / Write (Configuration) Ports

- High speed SERDES (Aurora) Interface
- 2 Wire/Parallel JTAG(IEEE 1149.7) Interface

www.nexus5001.org
Nexus 5001 Benefits

**MCU Vendors**
- Simplifies tools support
- Customer understanding and access of tools strategy
- Re-use reduces time/cost
- Leverage best in class tools
- Easier, porting of tools on new architectures
- Standard methodology for development & production
- Trace without the bus (reduced pin overhead)

**Users**
- Learning cycles – tools and architecture(s)
- Quicker time to market
- Development tool reuse
- Proven capabilities - multiple architectures
- Single small footprint interface
- Real Time debugging, triggering and trace

**Tool Vendors**
- Reduced development cost
- Rapid migration to new architectures
- Standard functions
- Opportunity to differentiate tools
- Ease in addressing new tool requirements
- High performance – lower cost tools

www.nexus5001.org
Nexus 5001™ – Key Technical Points

• Packet based Debug Protocol
  – Simpler Multicore support
  – Configurable fields

• Predefined Debug Instruction set (TCODES)
  – 35 Instruction and Data Trace, register, and memory access commands
  – In place support from leading commercial tool vendors
  – Configurable for additional user/vendor defined messages

• Predefined Register Set (optional)
  – Simplified core level integration

• JTAG compatible interface
  – Support for both IEEE 1149.1 and 1149.7 standards

• Parallel interfaces options for data transfer
  – Parallel bus (AUX ports) or SerDes compatible interface (Aurora)
  – User defined channels for Output (Trace) and/or Input (Calibration)

• User defined mix and match of JTAG and parallel interfaces
  – Ie. JTAG input for control, parallel output for trace
SoC Debug – A Basic Nexus 5001 Architecture

- Nexus supports a modular architecture
- Debug Instructions (TCODES) and formatting, External Tool Interfaces are from de-coupled from Processor and SoC debug blocks

www.nexus5001.org
Nexus 5001™ Debug Environment

- Processor Independent data access
- Multiple on-chip processor/core support
- Packet-Based Messaging
  - Program Trace
  - Data Trace
  - Memory Substitution
  - Vendor-Defined
- 1149 TAP - 2 or 4 wire Protocol and/or packet based Messaging
- AUX / SERDES Ports, High Performance Access to resources

- Logic
- Performance Trace
- Debugger, SW Emulation Acquisition, Prototyping Run-Time Debugger, Parameter Tuning, Calibration

System Run Control, trigger in/out signals
Synchronized Timestamps Cross-triggers
Core A to Nexus Interface
Core B to Nexus Translation
Embedded Nexus Bus Trace
TCODE & Message Control/Formatting
Nexus Registers
JTAG Debug Registers
JTAG FSMs
AUX / SERDES Out FSMs
AUX / SERDES In FSMs
Auxiliary/SERDES Output
Auxiliary/SERDES Input
AUX/SERDES Ports
High Performance Access to resources

User Defined Domain
Nexus 5001™ Domain
New Nexus 5001™ Specification

• Nexus 5001™ Forum has released 3rd revision of the IEEE-5001 standard.
  – IEEE-ISTO 5001-1999
  – IEEE-ISTO 5001-2003

– IEEE-ISTO 5001-2012
  • Adds 1149.7 support – 2 wire and Parallel JTAG
  • Adds Hi Speed link-layer protocol for SERDES
  • Specification Ratification completed May 2012
What is IEEE 1149.7

- Next Generation JTAG – Added features for improved performance
  - Compatible with 1149.1

Advanced Debug Features

JTAG Extended Features

JTAG Compliant Features

Class 5- Background Debug Mode Data Channel (BDX), Custom Debug Mode pin support (CDX)

Class 4- 2 Pin Parallel (Star-2) Topologies

Class 3 – Parallel (Star-4) Scan Topologies

Class 2 – Chip bypass in a Series Scan Topology

Class 1 – Basic 1149.7 control and function extensions, power control

Class 0 – multiple on-chip TAPs with IEEE 1149.1 compatibility

User Selectable addition of Advanced Debug Features

www.nexus5001.org
2012 Nexus Standard –IEEE 1149.7

– 2-wire JTAG interface (TCK, TMS, TRST).
– Alternative to IEEE 1149.1 (TCK, TDI, TDO, TMS, TRST)
– Supports Parallel JTAG configurations
What is Aurora Interface

- Aurora is a high-speed serial, link level interface that supports either a single or multiple lane channel.
- The Aurora protocol defines the physical layer, the link layer, data striping for utilizing one or more lanes, and flow control.
  - Each lane is a pair of wires using LVDS.
  - Data striping is performed in lane order with 2 symbols per lane for the 8B10B signaling.

See [http://www.xilinx.com/aurora](http://www.xilinx.com/aurora)
Xilinx has Aurora Bus functional models available.
2012 Nexus standard– Aurora

– Allows upper layers Nexus 5001 protocols to use high-speed serial links.

• scalable, lightweight, low-latency link-layer protocol
• open protocol - free of charge.
• transparent interface to the physical serial links,
• Supports LVDS type interface.
• 1 to 5 Gbps trace output - increased bandwidth through bonded lanes
• Data can be packed into a vendor defined (16 or 32 bits) register size
Aurora Interface Signaling

Aurora Supports Nexus transactions in both Framed and Streaming transfer modes

www.nexus5001.org
A system level Nexus 5001™ configuration

Subsystem 1
- core
- Trace RAM

Subsystem 2

Subsystem 3

Subsystem 4

Processor Cross-triggers
1149.1 JTAG chain

JTAG
2-wire (1149.7)

Debug Control Messages

Debug/Trace Data Messages

Aurora Serdes Channels

Trace Combiner Router

Trace Buffer

Bidirectional For calibration capabilities

www.nexus5001.org
Ongoing Nexus Work

• SoC Technologies are evolving –
• Nexus Instrumentation Standards are focused on keeping pace with emerging needs

• Debug/instrumentation description language/formats
• Trusted Systems - Debug interface/data security
• Debug in diverse power domains

• Emerging Debug configurations
  - Customized JTAG implementations
  - Background Data Transport (1149.7 BDX)
    - improve throughput using idle bandwidth
  - Custom Data Transport (1149.7 CDX)
    - custom link protocols

www.nexus5001.org