NEXUS 5001 Forum  Debug Interface Standard

www.ieee-isto.org/Nexus5001/

Gerhard Martin
Founding Member of NEXUS Technical Technical Committee

Infineon Technologies AG
Presentation Topics

- Introduction
- Development Features
- Hardware Interface
- Software Interface
- Board/Connector Interface
- Summary
- Question/Answers
Introduction, Why NEXUS is needed

- Introduction
- NEXUS Consortium
- Development Features
- Hardware Interface
- Software Interface
- Board/Connector Interface
- Summary
- Question/Answers

ECU = Electronic Control Unit

IEEE-ISTO NEXUS 5001 debug standard
Introduction, Conventional ECU Design

- Introduction
- NEXUS Consortium
- Development Features
- Hardware Interface
- Software Interface
- Board/Connector Interface
- Summery
- Question/Answers
Introduction, Tomorrows ECU’s

- Introduction
- NEXUS Consortium
- Development Features
- Hardware Interface
- Software Interface
- Board/Connector Interface
- Summery
- Question/Answers

Introduction

NEXUS Consortium

Development Features

Hardware Interface

Software Interface

Board/Connector Interface

Summery

Question/Answers
The NEXUS Consortium
Current IEEE-ISTO NEXUS 5001 Membership List

Silicon Vendors:
- Hitachi Semiconductor Inc.
- Infineon Technologies AG
- Mitsubishi Electric Corp.
- Motorola
- ST Microelectronics

Tool Vendors (cont’d):
- Emulation Technology
- ETAS
- Green Hills Software Inc. (GHS)
- Hitex Development Tools
- HIWARE
- Lauterbach Inc.
- Macraigor Systems, L.L.C.
- Metroworks Corporation
- Nohau
- Noral Micrologics
- SDS
- PLX Technology, Inc
- Tektronix
- Yokogawa Digital Computer Corp.
Nexus Committees and Their Roles

- **Business**: Construct and co-ordinate activities between competing companies
- **Technical**: Develop workable spec. Co-ordinate technical activities with other groups
- **Validation**: Develop verification methodology for architectures and tools
- **API**: Develop abstraction layers and software interface for tools and silicon

- Introduction
- NEXUS Consortium
- Development Features
- Hardware Interface
- Software Interface
- Board/Connector Interface
- Summary
- Question/Answers
What Benefits Does Nexus Provide?

**Benefits for the developer:**
- Investment protection since the tools can be reused for another architecture
- Tools can be available with the introduction of new embedded MCUs
- Reduced customization, training and development time

**Benefits for the industry:**
- The verification of the tools & embedded MCU capabilities, prior to the use in customer applications, will reduce the risk & will improve the quality

**For the tools companies:**
- allows tool partner to focus investments at solving more customer problems and not at the various MCU traceport implementations

**For the semiconductor manufacturers:**
- It is much easier to get tool vendor support for new embedded MCUs.
Scope of IEEE-ISTO NEXUS 5001 Standard

- Calibration
- Logic Analysis
- Run Control
- Rapid Prototyping

Nexus Standard Focus:
- Debug/Cal Protocol
- Mechanical Interconnect
- Silicon

Controller Board

Instrumentation Interface
Development Features

- Introduction
- NEXUS Consortium
- Development Features
- Hardware Interface
- Software Interface
- Board/Connector Interface
- Summary
- Question/Answers

The Infineon NEXUS debug port
Nexus Development Features

- Introduction
- NEXUS Consortium
- Development Features
- Hardware Interface
- Software Interface
- Board/Connector Interface
- Summery
- Question/Answers

Processor independent
Supports multiple on-chip processors

Packet-Based Messaging
- Program Trace
- Data Write
- Virtual Memory
- Vendor-Defined

Embedded MCU/DSP

Auxiliary Output

Auxiliary In

JTAG

Debugger, Logic Analyzer, Data Acquisition, Prototyping

Debugger, Logic Analyzer,

JTAG Protocol or Packet-Based Messaging
- Development Control and Status
- Read/Write Access During Runtime

IEEE-ISTO NEXUS 5001 debug standard
Public Message - Direct Branch Message

- Introduction
- NEXUS Consortium
- Development Features
- Hardware Interface
- Software Interface
- Board/Connector Interface
- Summary
- Question/Answers

**MCKO**

**MSEO**

**MDO**

- TCODE
- Processor ID
- # of Completed Instructions (variable length)

(Output)
# NEXUS Compliance Classes

<table>
<thead>
<tr>
<th>GEPDIS Class</th>
<th>Class 1</th>
<th>Class 2</th>
<th>Class 3</th>
<th>Class 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trace features</td>
<td>Trace not supported</td>
<td>Adds ownership trace and program trace via Auxiliary ports</td>
<td>Adds data write trace and read/write memory on the fly via Auxiliary ports</td>
<td>Allows tracing to be triggered by a watchpoint via Auxiliary ports</td>
</tr>
<tr>
<td>Debug communication method</td>
<td>Half-duplex communication (Limited bandwidth)</td>
<td>Communication may be full duplex using Auxiliary port (higher bandwidth)</td>
<td>Communication may be full duplex using Auxiliary port (higher bandwidth)</td>
<td>Communication may be full duplex using Auxiliary port (higher bandwidth)</td>
</tr>
<tr>
<td>run-time Control</td>
<td>Supports run time control features using JTAG interface</td>
<td>Supports run time control features using JTAG interface or Auxiliary port</td>
<td>Supports run time control features using JTAG interface or Auxiliary port</td>
<td>Supports run time control features using JTAG interface or Auxiliary port</td>
</tr>
<tr>
<td>Auxiliary Port Implementation</td>
<td>No Auxiliary Port</td>
<td>Allows Port sharing; the Auxiliary port may be shared with slow IO port pins (e.g. static Config pins that are latched at reset time).</td>
<td>Allows Port sharing with high-speed I/O ports.</td>
<td>Allows Port sharing with high-speed I/O ports.</td>
</tr>
<tr>
<td>Data acquisition</td>
<td>Not supported</td>
<td>Not supported</td>
<td>Supports data acquisition</td>
<td>Supports data acquisition</td>
</tr>
</tbody>
</table>
| Memory Substitution | Not supported | Not supported | Not supported | Supports memory substitution (fetching or reading data over the GEPDIS auxiliary port)
Triggersing memory substitution on a watchpoint is an optional feature of Class 4 compliance |
## Static Development Features

<table>
<thead>
<tr>
<th>Development Feature</th>
<th>Class 1</th>
<th>Class 2</th>
<th>Class 3</th>
<th>Class 4</th>
<th>Nexus Feature</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read/write user registers in debug mode</td>
<td>V</td>
<td>V</td>
<td>V</td>
<td>V</td>
<td>Refer to SECTION 5</td>
</tr>
<tr>
<td>Read/write user memory in debug mode</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>Read/Write Access</td>
</tr>
<tr>
<td>Enter a debug mode from reset</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>Development Control and Status</td>
</tr>
<tr>
<td>Enter a debug mode from user mode</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td></td>
</tr>
<tr>
<td>Exit a debug mode to user mode</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td></td>
</tr>
<tr>
<td>Single step instruction in user mode and re-enter debug mode</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>Breakpoints/Watchpoints</td>
</tr>
<tr>
<td>Stop program execution on instruction/data breakpoint and enter debug mode (min. 2 breakpoints)</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td></td>
</tr>
</tbody>
</table>

V = Vendor specific implementation  
A = NEXUS API support required
## Dynamic Development Features

<table>
<thead>
<tr>
<th>Development Feature</th>
<th>Class 1</th>
<th>Class 2</th>
<th>Class 3</th>
<th>Class 4</th>
<th>Nexus Feature</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ability to set breakpoint or watchpoint</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>Breakpoints/Watchpoints</td>
</tr>
<tr>
<td>Device Identification</td>
<td>A</td>
<td>A &amp; P</td>
<td>A &amp; P</td>
<td>A &amp; P</td>
<td>Device ID Message (see SECTION 6)</td>
</tr>
<tr>
<td>Ability to send out an event occurrence when watchpoint matches</td>
<td>P&lt;sup&gt;1&lt;/sup&gt;</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>Watchpoint Message (see SECTION 6)</td>
</tr>
<tr>
<td>Monitor process ownership while processor runs in real-time</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td></td>
<td>Ownership Trace</td>
</tr>
<tr>
<td>Monitor program flow while processor runs in real-time (logical address)</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td></td>
<td>Program Trace</td>
</tr>
<tr>
<td>Monitor data writes while processor runs in real-time</td>
<td>P</td>
<td>P</td>
<td></td>
<td></td>
<td>Data Trace (Writes only)</td>
</tr>
<tr>
<td>Read/write memory locations while program runs in real-time</td>
<td>A &amp; P</td>
<td>A &amp; P</td>
<td></td>
<td></td>
<td>Read/Write Access</td>
</tr>
<tr>
<td>Program execution (instruction/data) from Nexus port for reset or exceptions</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Memory Substitution</td>
</tr>
<tr>
<td>Ability to start ownership, program, or data trace upon watchpoint occurrence</td>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td>Development Control and Status</td>
</tr>
<tr>
<td>Ability to start Memory Substitution upon watchpoint occurrence or upon program access of device-specific address</td>
<td></td>
<td></td>
<td></td>
<td>O</td>
<td>Development Control and Status</td>
</tr>
<tr>
<td>Monitor data reads while processor runs in real-time</td>
<td>O</td>
<td>O</td>
<td></td>
<td></td>
<td>Data Trace (Reads and Writes)</td>
</tr>
<tr>
<td>Low speed I/O port replacement and High speed I/O port sharing</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td></td>
<td>Port Replacement/Sharing</td>
</tr>
</tbody>
</table>

A = NEXUS API support required  
P = NEXUS T-CODES protocol support required  
O = optional but not required for Nexus Compliance
NEXUS On-Chip Hardware
The hardware implementations will be dependent on the respective silicon vendor.

The design should be scaleable, a Class 2 Implementation should be reusable for a Class 3 Implementation.
Efficient Data Transfers using Nexus Hardware

Example of how the target processor generates the address to send in a trace message:

Previous absolute address (A1) = 0x003FC01,
Absolute address associated with new trace occurrence (A2) = 0x0003F365

\[
\begin{align*}
A1 & = 0000 \ 0000 \ 0000 \ 0011 \ 1111 \ 1100 \ 0000 \ 0001 \\
A2 & = 0000 \ 0000 \ 0000 \ 0011 \ 1111 \ 0011 \ 0110 \ 0101 \\
A1 \oplus A2 & = 0000 \ 0000 \ 0000 \ 0000 \ 1111 \ 0110 \ 0110 \ 0100 \\
\end{align*}
\]

The unique portion of the address (M1), sent in the message (high-order zeros are suppressed):

\[
M1 = 1111 \ 0110 \ 0100
\]

Example of how the tool recreates the address based on its previously calculated address and the address contained in the trace message:

Previously calculated address (A1) = 0x003FC01,
Address in message (M1) = 0xF64

\[
\begin{align*}
A1 & = 0000 \ 0000 \ 0000 \ 0011 \ 1111 \ 1100 \ 0000 \ 0001 \\
M1 & = 0000 \ 0000 \ 0000 \ 0000 \ 0000 \ 1111 \ 0110 \ 0100 \\
A1 \oplus M1 & = 0000 \ 0000 \ 0000 \ 0011 \ 1111 \ 0111 \ 0110 \ 0101 \\
\end{align*}
\]

Address recreated by the tool = 0x0003F365
NEXUS API Software Interface
• Nexus Standard supplies API header files
• Semiconductor vendor provides Emulator vendor with Target Abstraction Layer
• Emulator vendor provides Tool vendor with integrated Nexus API layer
NEXUS API Calls

Target Abstraction Layer

– Open - Establish connection to Target
– Close - Close connection to Target
– ioctl - Controlling connection to Target
– ReadMem / WriteMem - Memory access to Target
– SetEvent / GetEvent / ClearEvent - Event handling of Target
– GetLastError - Error Handling/Recovery

Emulator Hardware Abstraction Layer (HAL)

– Open - Establish connection to Emulator
– Close - Close connection to Emulator
– ReadNJR / WriteNJR - Access of JTAG resources via Emulator.
Board / Connector Interface
# Nexus Connector Interface

## Table: Nexus Connector Interface

<table>
<thead>
<tr>
<th>Pin Name</th>
<th>Conn. A JTAG Port Only</th>
<th>Conn. B Option 1 - JTAG</th>
<th>Conn. B Option 2 - Aux Port</th>
<th>Conn. B Option 3 - Mixed</th>
<th>Conn. C Aux Port Only</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>MCKI</td>
<td></td>
<td>1</td>
<td>1</td>
<td></td>
<td>1</td>
<td>Auxiliary Port</td>
</tr>
<tr>
<td>MDI</td>
<td></td>
<td>2</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSEI</td>
<td></td>
<td>1</td>
<td>1</td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>MCKO</td>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MDO</td>
<td></td>
<td>4</td>
<td>2</td>
<td>8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSEO</td>
<td></td>
<td>1</td>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EVTO</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>Port Replacement</td>
</tr>
<tr>
<td>EVTI</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>RSTI</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>PORT</td>
<td></td>
<td>1</td>
<td>16</td>
<td></td>
<td></td>
<td>Port Replacement</td>
</tr>
<tr>
<td>JTAG</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td></td>
<td></td>
<td>IEEE 1149.1</td>
</tr>
<tr>
<td>RDY</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Vendor Defined</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>VREF</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>System Signals</td>
</tr>
<tr>
<td>RESET</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>CLOCKOUT</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Other</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>

**IEEE-ISTO NEXUS 5001 debug standard**
NEXUS Connector A

- 20 Pin Connector which supports JTAG 5 pin Protocol
- Amp 2 row x 10 columns,
- 0.050 inch pitch pin spacing and 0.10 in row spacing.
- Includes System_Clock_out, System_Reset_in,
- Data_Rdy_out, Evti_in, Evto_out
- Supports 3.3v and lower with System_Vref_out
NEXUS Connector B

- **30 Pin Amp Connector** supports JTAG 5 pin Protocol and/or Aux Port
- **Amp 3 row x 10 columns, 0.050 in. column X 0.10 in row spacing**

### Signal Table

<table>
<thead>
<tr>
<th>Signal Name</th>
<th>Signal Name</th>
<th>Signal Name</th>
<th>I/O</th>
<th>Pin</th>
<th>Pin</th>
<th>I/O</th>
<th>Signal Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1) JTAG Mode</td>
<td>2) Aux Mode</td>
<td>3) Mixed Mode</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RESET</td>
<td>RESET</td>
<td>RESET</td>
<td>IN</td>
<td>1</td>
<td>2</td>
<td>OUT</td>
<td>VREF</td>
</tr>
<tr>
<td>EVTI</td>
<td>EVTI</td>
<td>EVTI</td>
<td>IN</td>
<td>3</td>
<td>4</td>
<td></td>
<td>GND</td>
</tr>
<tr>
<td>TRST</td>
<td>TRST</td>
<td>TRST</td>
<td>IN</td>
<td>5</td>
<td>6</td>
<td></td>
<td>GND</td>
</tr>
<tr>
<td>TMS</td>
<td>TMS</td>
<td>TMS</td>
<td>IN</td>
<td>7</td>
<td>8</td>
<td></td>
<td>GND</td>
</tr>
<tr>
<td>RESERVED</td>
<td>MDIT</td>
<td>RESERVED</td>
<td>IN</td>
<td>9</td>
<td>10</td>
<td></td>
<td>GND</td>
</tr>
<tr>
<td>TDI</td>
<td>MD10</td>
<td>TDI</td>
<td>IN</td>
<td>11</td>
<td>12</td>
<td></td>
<td>GND</td>
</tr>
<tr>
<td>TCK</td>
<td>MCKI</td>
<td>TCK</td>
<td>IN</td>
<td>13</td>
<td>14</td>
<td></td>
<td>GND</td>
</tr>
<tr>
<td>RESERVED</td>
<td>NSEC</td>
<td>RESERVED</td>
<td>IN</td>
<td>15</td>
<td>16</td>
<td></td>
<td>GND</td>
</tr>
<tr>
<td>TDO</td>
<td>MD03</td>
<td>TDO</td>
<td>OUT</td>
<td>17</td>
<td>18</td>
<td></td>
<td>GND</td>
</tr>
<tr>
<td>RDO</td>
<td>MD02</td>
<td>RDO</td>
<td>OUT</td>
<td>19</td>
<td>20</td>
<td></td>
<td>GND</td>
</tr>
<tr>
<td>RESERVED</td>
<td>MD01</td>
<td>MD01</td>
<td>OUT</td>
<td>21</td>
<td>22</td>
<td></td>
<td>GND</td>
</tr>
<tr>
<td>RESERVED</td>
<td>MD00</td>
<td>MD00</td>
<td>OUT</td>
<td>23</td>
<td>24</td>
<td></td>
<td>GND</td>
</tr>
<tr>
<td>CLOKOUT</td>
<td>MCKO</td>
<td>MCKO</td>
<td>OUT</td>
<td>25</td>
<td>26</td>
<td></td>
<td>GND</td>
</tr>
<tr>
<td>RESERVED</td>
<td>MSEE</td>
<td>MSEE</td>
<td>OUT</td>
<td>27</td>
<td>28</td>
<td></td>
<td>GND</td>
</tr>
<tr>
<td>EVTO*</td>
<td>EVTO*</td>
<td>EVTO*</td>
<td>OUT</td>
<td>29</td>
<td>30</td>
<td>I or O</td>
<td>Vendor Defined*</td>
</tr>
</tbody>
</table>
NEXUS Connector C

- Intended for target processors which support wide aux. port and port replacement requirements.
- 80 Pin Samtec MOLC/FOLC Connector
- Amp 2 row x 10 columns, 0.050 in. column X 0.050 in row spacing

<table>
<thead>
<tr>
<th>Signal Name</th>
<th>I/O</th>
<th>Pin</th>
<th>Pin</th>
<th>Pin</th>
<th>Pin</th>
<th>I/O</th>
<th>Signal Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESET</td>
<td>IN</td>
<td>1</td>
<td>2</td>
<td>UBATT*</td>
<td>3</td>
<td>4</td>
<td>OUT</td>
</tr>
<tr>
<td>EVTI*</td>
<td>IN</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>OUT</td>
<td>EVTO*</td>
</tr>
<tr>
<td>RSTI</td>
<td>IN</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>I/O</td>
<td>Vendor Defined*</td>
</tr>
<tr>
<td>MDI3*</td>
<td>IN</td>
<td>13</td>
<td>14</td>
<td>15</td>
<td>16</td>
<td>I/O</td>
<td>Vendor Defined*</td>
</tr>
<tr>
<td>MDI2*</td>
<td>IN</td>
<td>17</td>
<td>18</td>
<td>19</td>
<td>20</td>
<td>I/O</td>
<td>PORT15*</td>
</tr>
<tr>
<td>MDI1*</td>
<td>IN</td>
<td>21</td>
<td>22</td>
<td>23</td>
<td>24</td>
<td>I/O</td>
<td>PORT14*</td>
</tr>
<tr>
<td>MDI0</td>
<td>IN</td>
<td>25</td>
<td>26</td>
<td>27</td>
<td>28</td>
<td>I/O</td>
<td>PORT13*</td>
</tr>
<tr>
<td>MCKI</td>
<td>IN</td>
<td>29</td>
<td>30</td>
<td>31</td>
<td>32</td>
<td>I/O</td>
<td>PORT12*</td>
</tr>
<tr>
<td>MSEI</td>
<td>IN</td>
<td>33</td>
<td>34</td>
<td>35</td>
<td>36</td>
<td>I/O</td>
<td>PORT11*</td>
</tr>
<tr>
<td>MDC07*</td>
<td>OUT</td>
<td>37</td>
<td>38</td>
<td>39</td>
<td>40</td>
<td>I/O</td>
<td>PORT10*</td>
</tr>
<tr>
<td>MDC06*</td>
<td>OUT</td>
<td>41</td>
<td>42</td>
<td>43</td>
<td>44</td>
<td>I/O</td>
<td>PORT9*</td>
</tr>
<tr>
<td>MDC05*</td>
<td>OUT</td>
<td>45</td>
<td>46</td>
<td>47</td>
<td>48</td>
<td>I/O</td>
<td>PORT8*</td>
</tr>
<tr>
<td>MDC04*</td>
<td>OUT</td>
<td>49</td>
<td>50</td>
<td>51</td>
<td>52</td>
<td>I/O</td>
<td>PORT7*</td>
</tr>
<tr>
<td>MDC03*</td>
<td>OUT</td>
<td>53</td>
<td>54</td>
<td>55</td>
<td>56</td>
<td>I/O</td>
<td>PORT6*</td>
</tr>
<tr>
<td>MDC02*</td>
<td>OUT</td>
<td>57</td>
<td>58</td>
<td>59</td>
<td>60</td>
<td>I/O</td>
<td>PORT5*</td>
</tr>
<tr>
<td>MDC01*</td>
<td>OUT</td>
<td>61</td>
<td>62</td>
<td>63</td>
<td>64</td>
<td>I/O</td>
<td>PORT4*</td>
</tr>
<tr>
<td>MDC00</td>
<td>OUT</td>
<td>65</td>
<td>66</td>
<td>67</td>
<td>68</td>
<td>I/O</td>
<td>PORT3*</td>
</tr>
<tr>
<td>MCKO</td>
<td>OUT</td>
<td>69</td>
<td>70</td>
<td>71</td>
<td>72</td>
<td>I/O</td>
<td>PORT2*</td>
</tr>
<tr>
<td>MSEO1*</td>
<td>OUT</td>
<td>73</td>
<td>74</td>
<td>75</td>
<td>76</td>
<td>I/O</td>
<td>PORT1*</td>
</tr>
<tr>
<td>MSEO0</td>
<td>OUT</td>
<td>77</td>
<td>78</td>
<td>79-UBATT*</td>
<td>80</td>
<td>I/O</td>
<td>PORT0*</td>
</tr>
</tbody>
</table>
Global Embedded Processor Debug Interface Standard

Eliminating Development Problems so that Real Application Problems can be Solved
For a copy of the current specification and to provide technical feedback visit our web site at

www. www.ieee-isto.org/Nexus5001/