A Seamless Tool Access Architecture from ESL to End Product

Albrecht Mayer (Infineon Microcontrollers)
S4D Conference
Sophia Antipolis, Sept. 2009
Tool Access Architecture (TAA)

TAA = Abstraction of physical connection
## TAA Applications and Targets

<table>
<thead>
<tr>
<th>Software Development</th>
<th>ESL Model</th>
<th>HDL Simulation</th>
<th>FPGA Prototype</th>
<th>Silicon/Device</th>
<th>Target System</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>X</td>
<td>(X)</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Silicon Debug</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>(X)</td>
</tr>
<tr>
<td>Silicon Validation Tests Develop.</td>
<td>(X)</td>
<td>(X)</td>
<td>(X)</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Debug of Tool Chain</td>
<td>(X)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>
TAA Independence

- Operating system of host (Windows, Linux, etc.)
- Core type (TriCore™, 8051, etc.)
- Physical interface of device (JTAG, CAN, etc.)
- SoC architecture
- Device access hardware
  - Physical connection (Ethernet, USB, etc.)
  - Low cost ← range → high end
  - Virtual and abstract for ESL-model (C-model)
  - Simulated for HDL simulator
Multi-Vendor IP-Model (ESL) System

Tool Core A

Tool Core B

Tool Core C

MCD API Implementation 0

Impl. 1

MCD API

System

Device A0

Core A0

Core B0

Core A1

Device A1

Core A0

Core B0

Core A1

Device B0

Core A0

Core C0
TAA Block Diagram

- Tool
- Core specific Layer
- Client Socket Layer
- Server Socket Layer
- Device specific Layer
- Access HW
- Device

TCP/IP (remote or local)

USB or Ethernet

JTAG, CAN, etc.
Infineon’s TAA Block Diagram

- Tool
- Core specific Layer
- Client Socket Layer

- Server Socket Layer
- Device specific Layer

- Access HW
- Device

- EXE
- MCD API DLL
- DAS API DLL
- TCP/IP (remote or local)

- UDAS Server
- USB or Ethernet
- miniWiggler, etc.
- JTAG, DAP, SPD, etc.
Infineon’s DAS

- Device Access Server
- Introduced in 2000
- www.infineon.com/DAS

DAS Basic Considerations:

1. Abstraction of physical interface
2. Bandwidth ok, latency an issue
3. Encapsulate dependencies on device and physical interface type
4. Robustness is key
DAS Mission

DAS

any tool

any wire

any device
Multi-Device Operation

- Host PC
  - XC800 Tool
  - XC2000 Tool
  - TriCore Tool
  - UDAS Server
  - USB

- Evaluation Board
  - XC2000
  - On-Board Wiggler JTAG

- Target System
  - TriCore
  - minWiggler DAP or JTAG
  - minWiggler SPD, DAP or JTAG
  - XC800
Parallel device analysis setup with up to 16 individual DAS device connections (JTAG over USB miniWigglers).

Devices under analysis are outside of this picture.
Multi-Tool Operation

Host PC 1
- Test Sequencer Tool
- UDAS Server
- USB

Host PC 2
- Debugger Tool

Test Board
- miniWiggler
- Device under Test

Connections:
- TCP/IP
- LAN/WAN
- SPD, DAP or JTAG
Remote Debugging Use Case Example (1/2)
Remote Debugging Use Case Example (2/2)

- Board shown on previous slide had a tricky bug which couldn’t be found for weeks
- Found this bug within hours by remote debugging using the DAS multi-tool, multi-device feature
- Unified tooling to debug all system components: Debugger → TAA → SoC → Board
Abstraction of Device and Connection

Host PC
- Tool
  - TCP/IP
- UDAS Server
  - USB
  - Ethernet

Compute Farm
- HDL Simulator TriCore
  - LAN

Automotive ECU
- XCP Slave
  - DAP or JTAG
- TriCore

Board
- miniWiggler
  - USB
  - DAP or JTAG
- TriCore
Debugger with ESL Model

TRACE32 Config String:

```plaintext
sys.mcd McdServerName="C-Model TriCore"
```
Debugger with Real Device

TRACE32 Config String:
```
sys.mcd McdServerName="UDAS"
```
Debugger with Real Device over XCP

TRACE32 Config String:

sys.mcd McdServerName="UDAS" McdAccHw.Port=1802
McdAccHw.Address="192.168.40.16"
Debugger with HDL Simulator

TRACE32 Config String:
```
sys.mcd McdServerName="UDAS" McdAccHw.Port=1784 McdAccHw.Address="mucsfbali13.muc.infineon.com"
```
Summary

- A seamless TAA from ESL to end product is needed
  - Reuse of tests, tools and know-how
  - Execute tasks earlier
    → Cost and risk reduction

- TAA needs to reflect “independence of” requirements

- Infineon’s TAA:
  - Tool interface DAS → MCD API
  - Broad productive use for silicon
  - Emerging use or at least demonstrated for other “targets”
We commit. We innovate. We partner. We create value.