A Seamless Tool Access Architecture from ESL to End Product

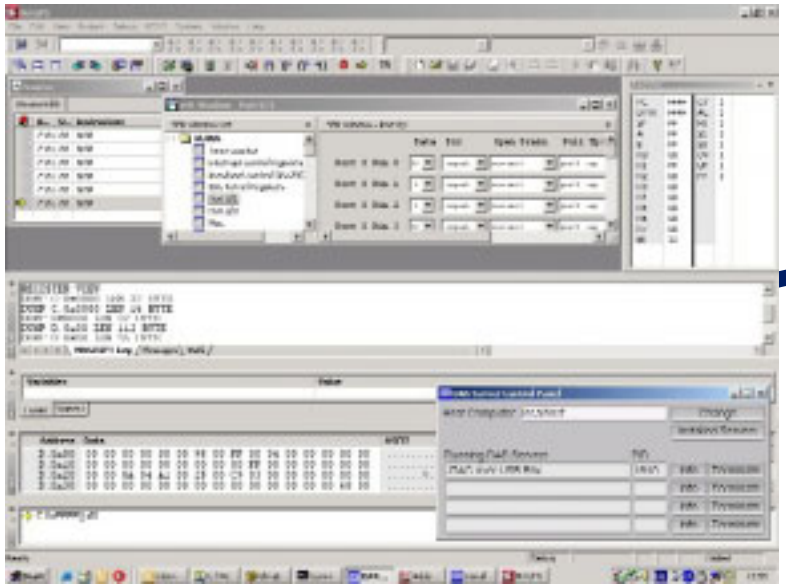
Albrecht Mayer (Infineon Microcontrollers)

S4D Conference

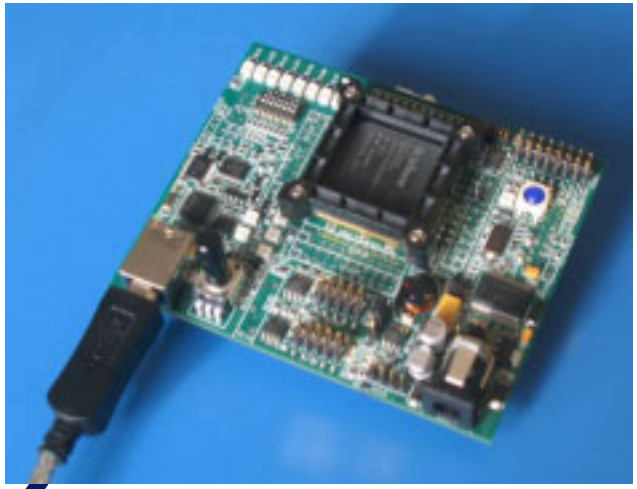
Sophia Antipolis, Sept. 2009

Never stop thinking

Tool Access Architecture (TAA)



Tool
to
Device



TAA
=
Abstraction
of physical
connection

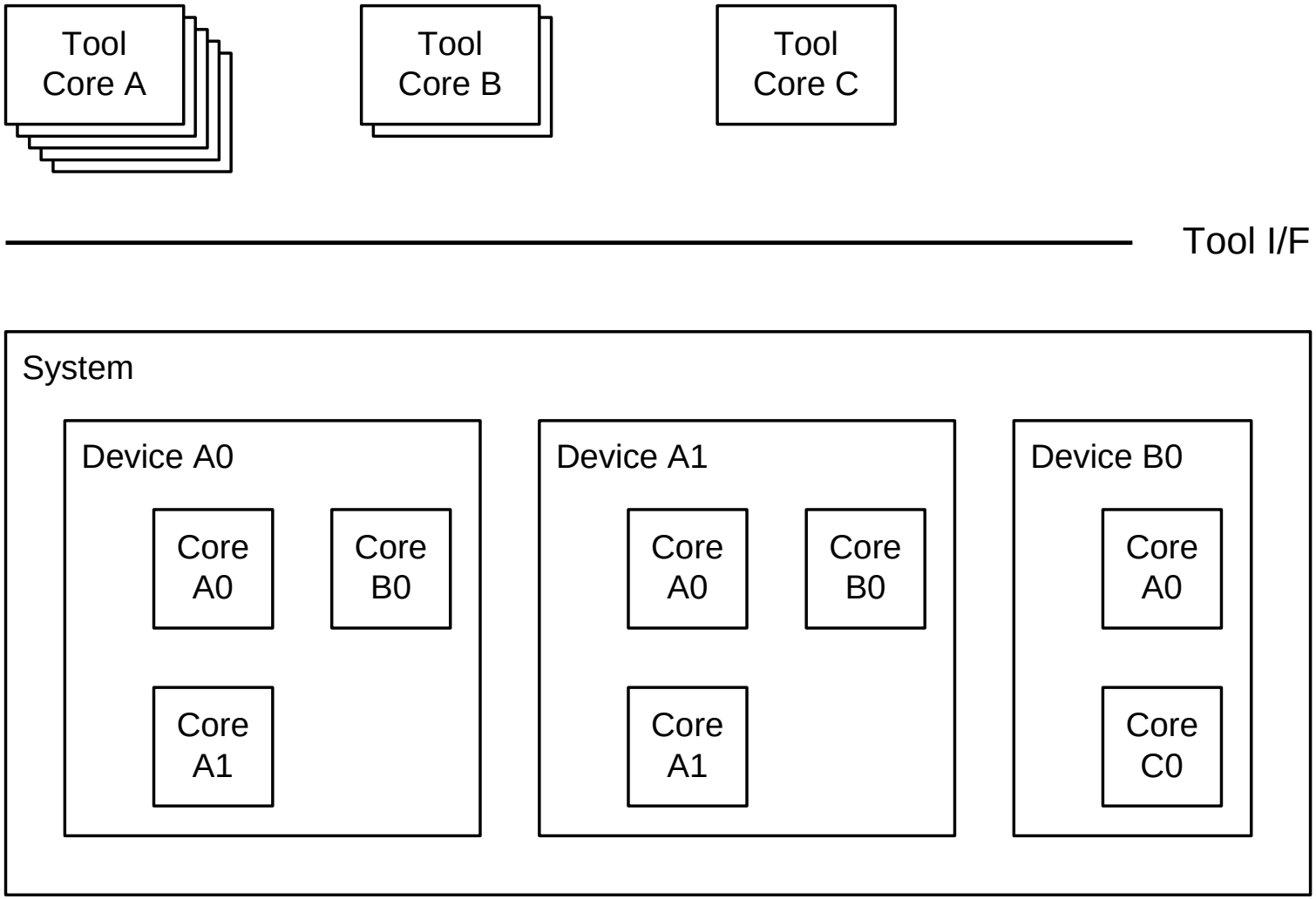
TAA Applications and Targets

	ESL Model	HDL Simulation	FPGA Prototype	Silicon /Device	Target System
Software Development	X	(X)	X	X	X
Silicon Debug				X	(X)
Silicon Validation Tests Develop.	(X)	(X)	(X)	X	
Debug of Tool Chain	(X)	X	X	X	

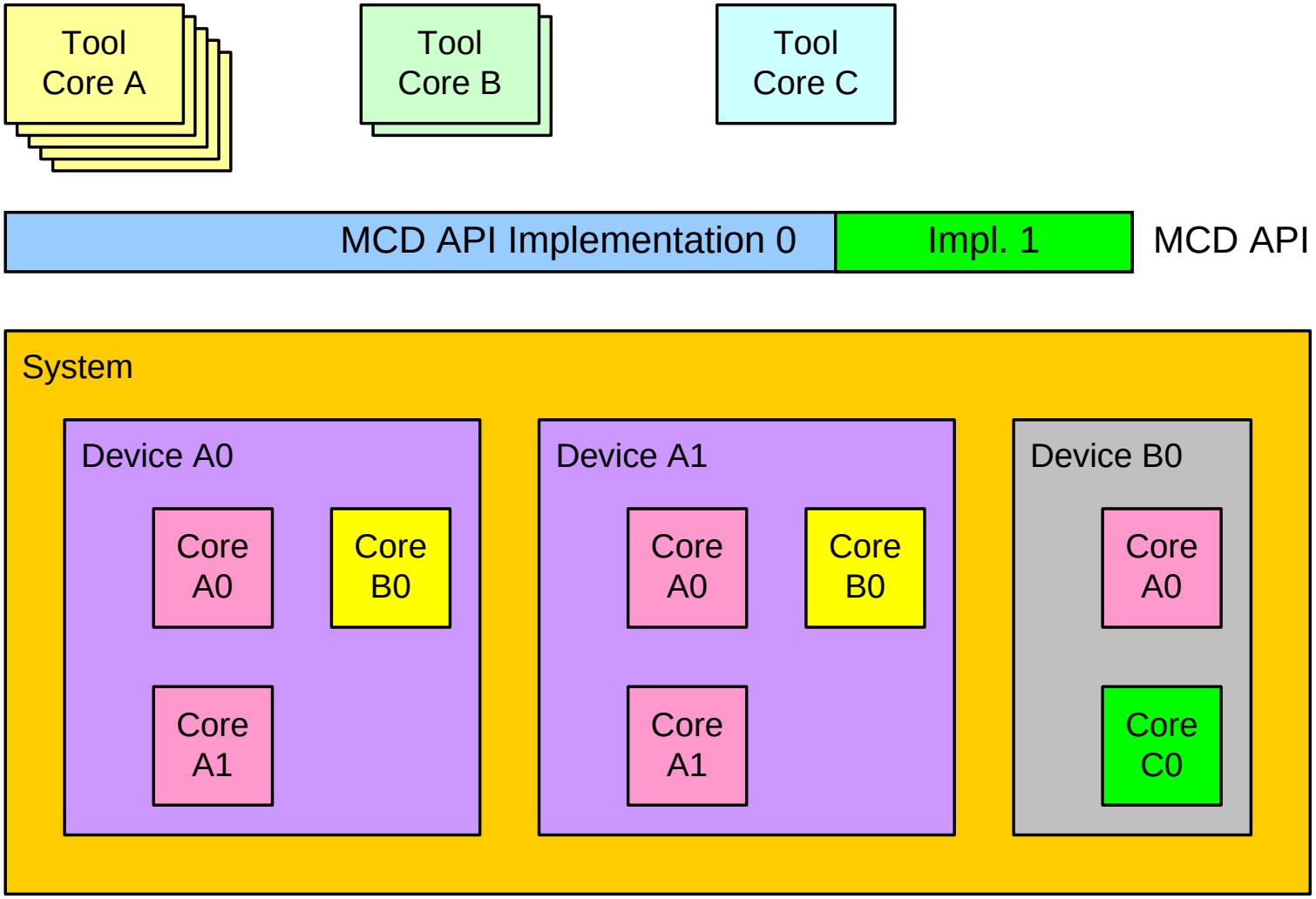
TAA Independence

- 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

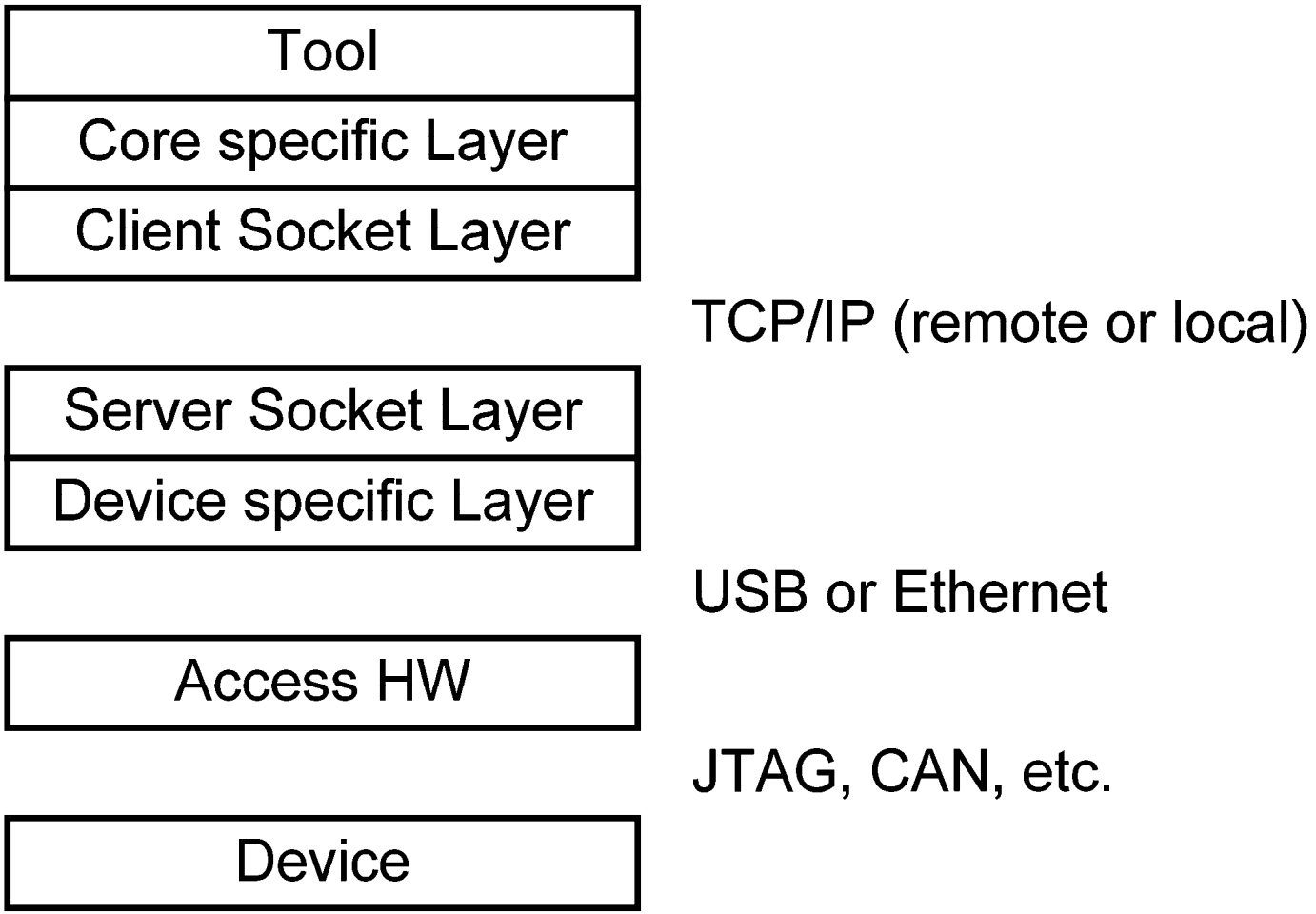
Multi-Device, Multi-Core System



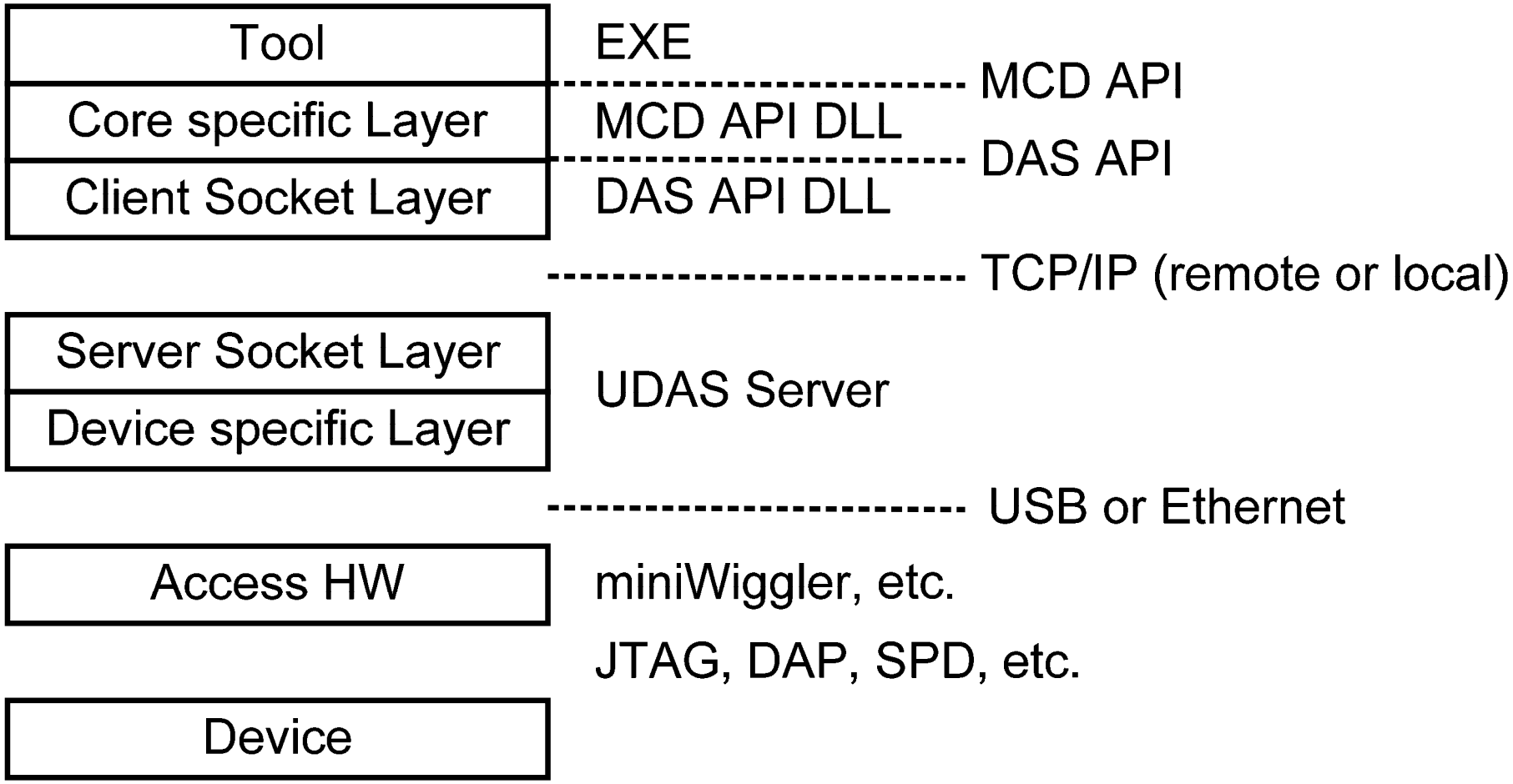
Multi-Vendor IP-Model (ESL) System



TAA Block Diagram



Infineon's TAA Block Diagram



Infineon's DAS

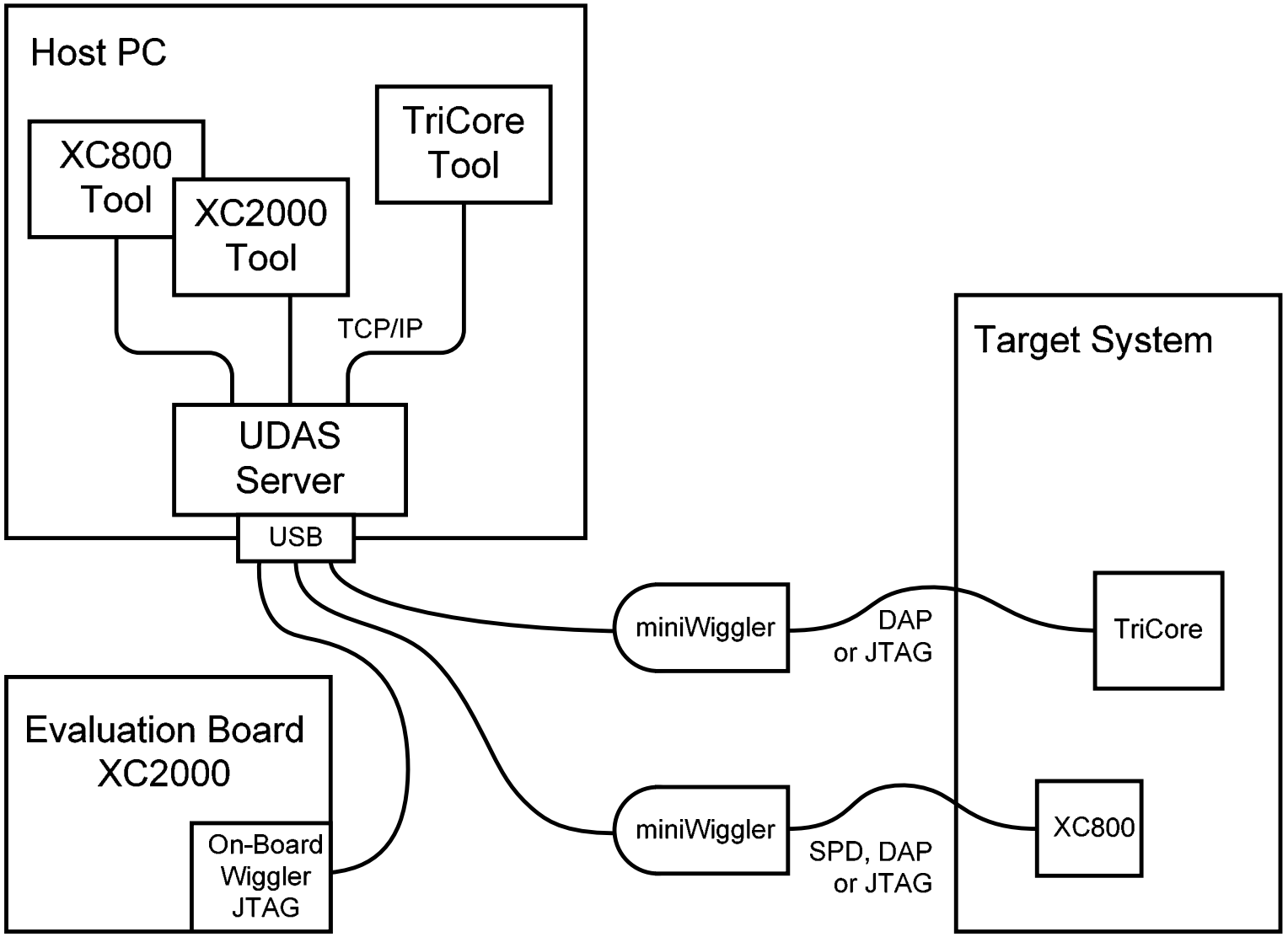
- Device Access Server
- Introduced in 2000
- www.infineon.com/DAS

DAS Basic Considerations:

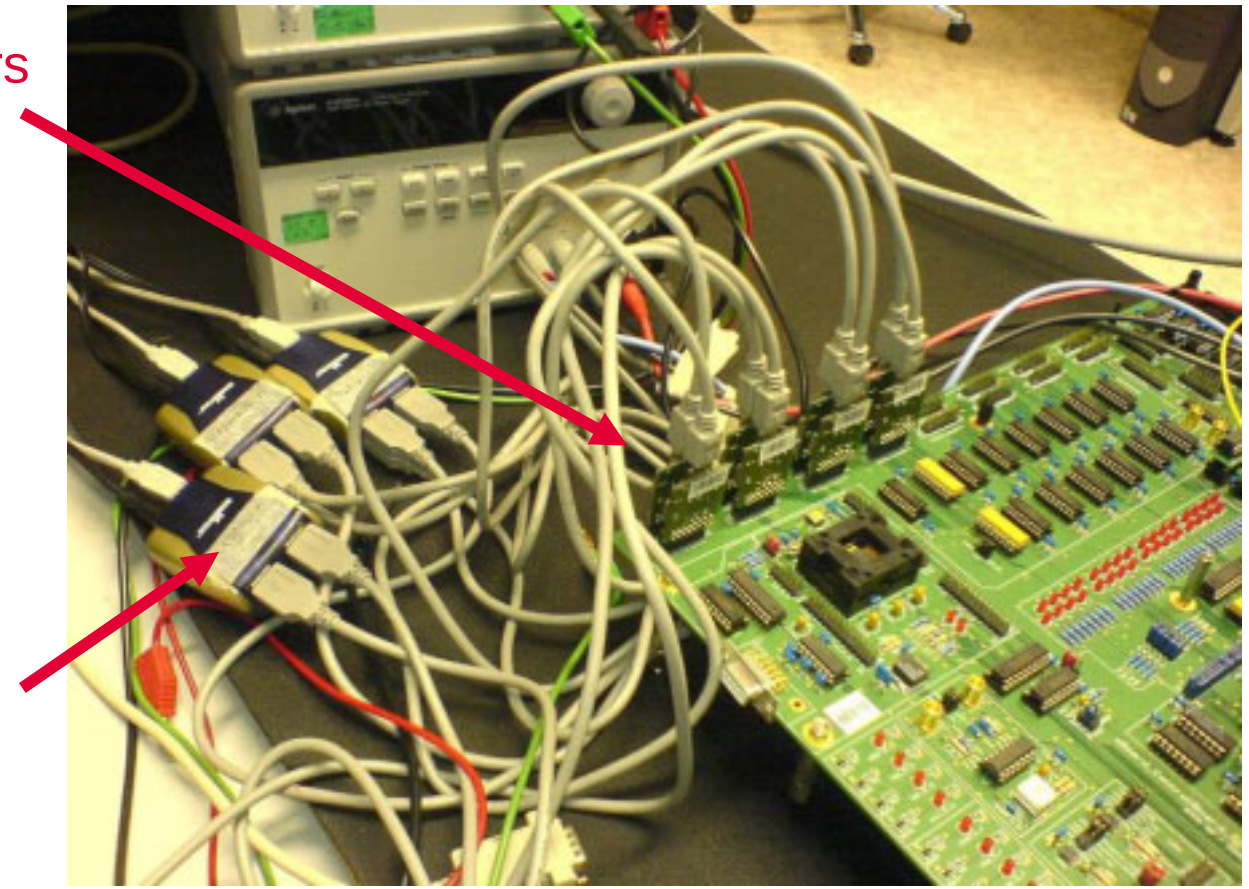
1. Abstraction of physical interface
2. Bandwidth ok, latency an issue
3. Encapsulate dependencies on device and physical interface type
4. Robustness is key

DAS Mission

Multi-Device Operation



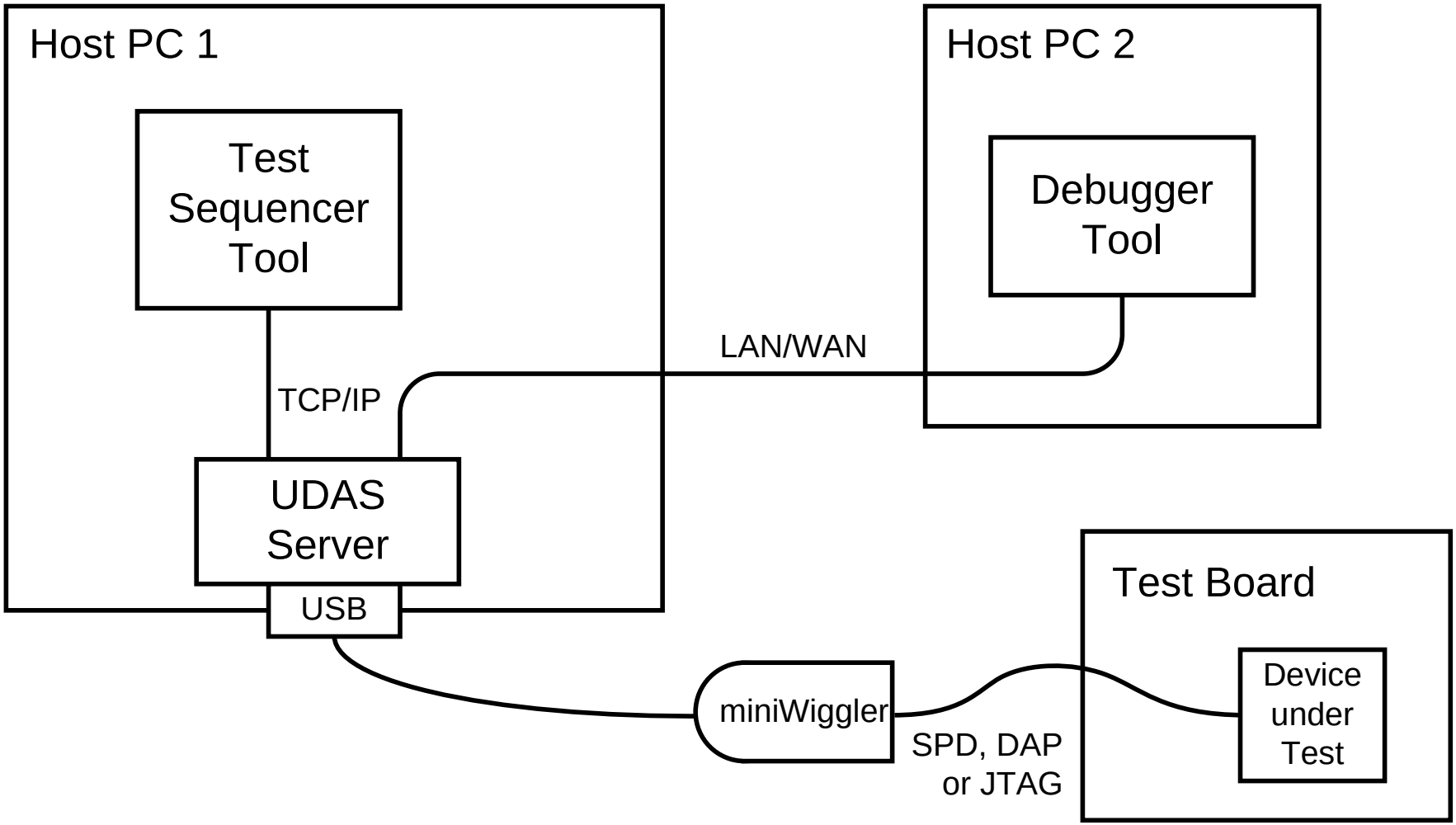
Multi-Device Use Case Example



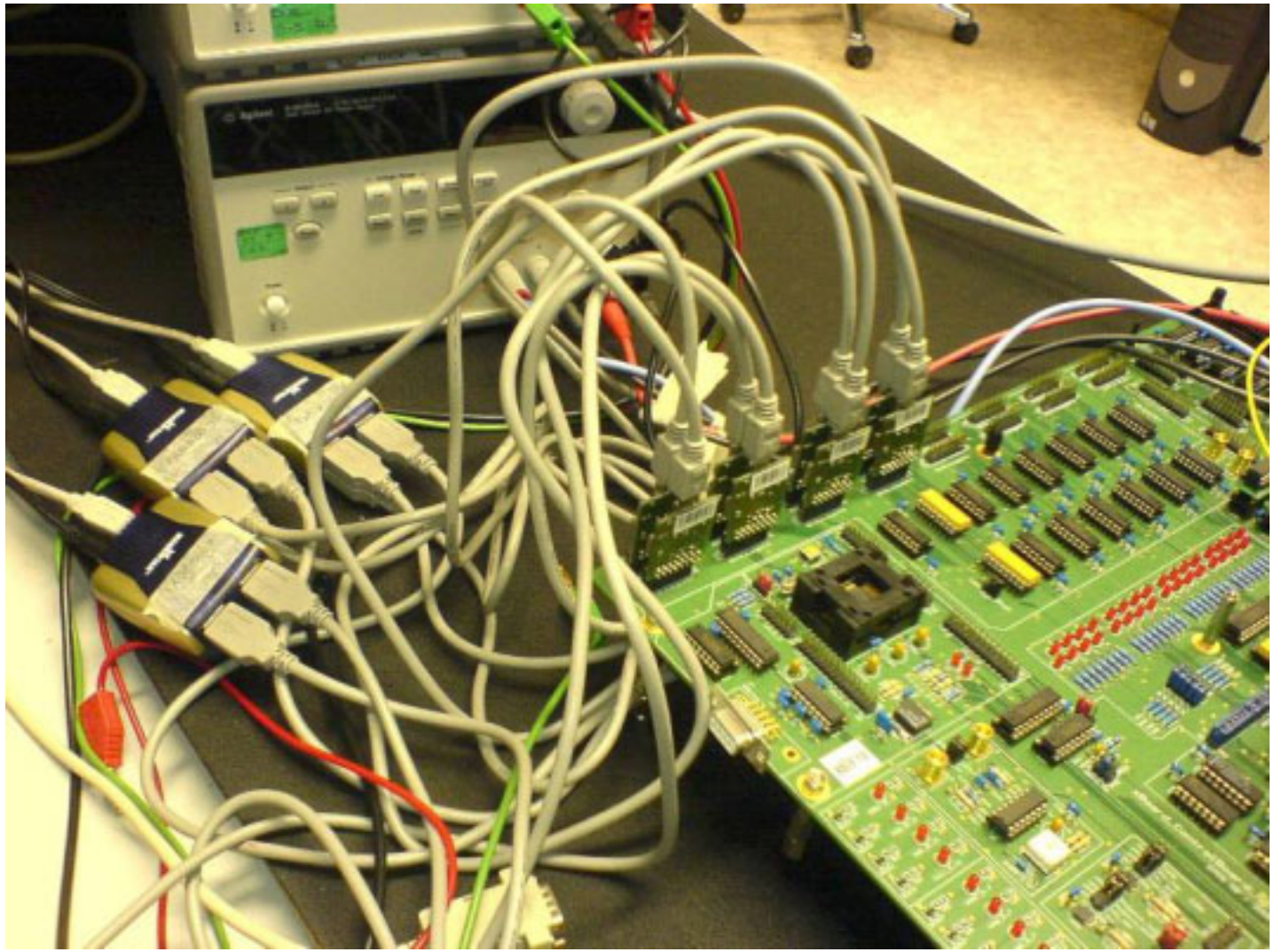
Parallel device analysis setup with up to 16 individual DAS device connections (JTAG over USB miniWigglers).

Devices under analysis are outside of this picture.

Multi-Tool Operation



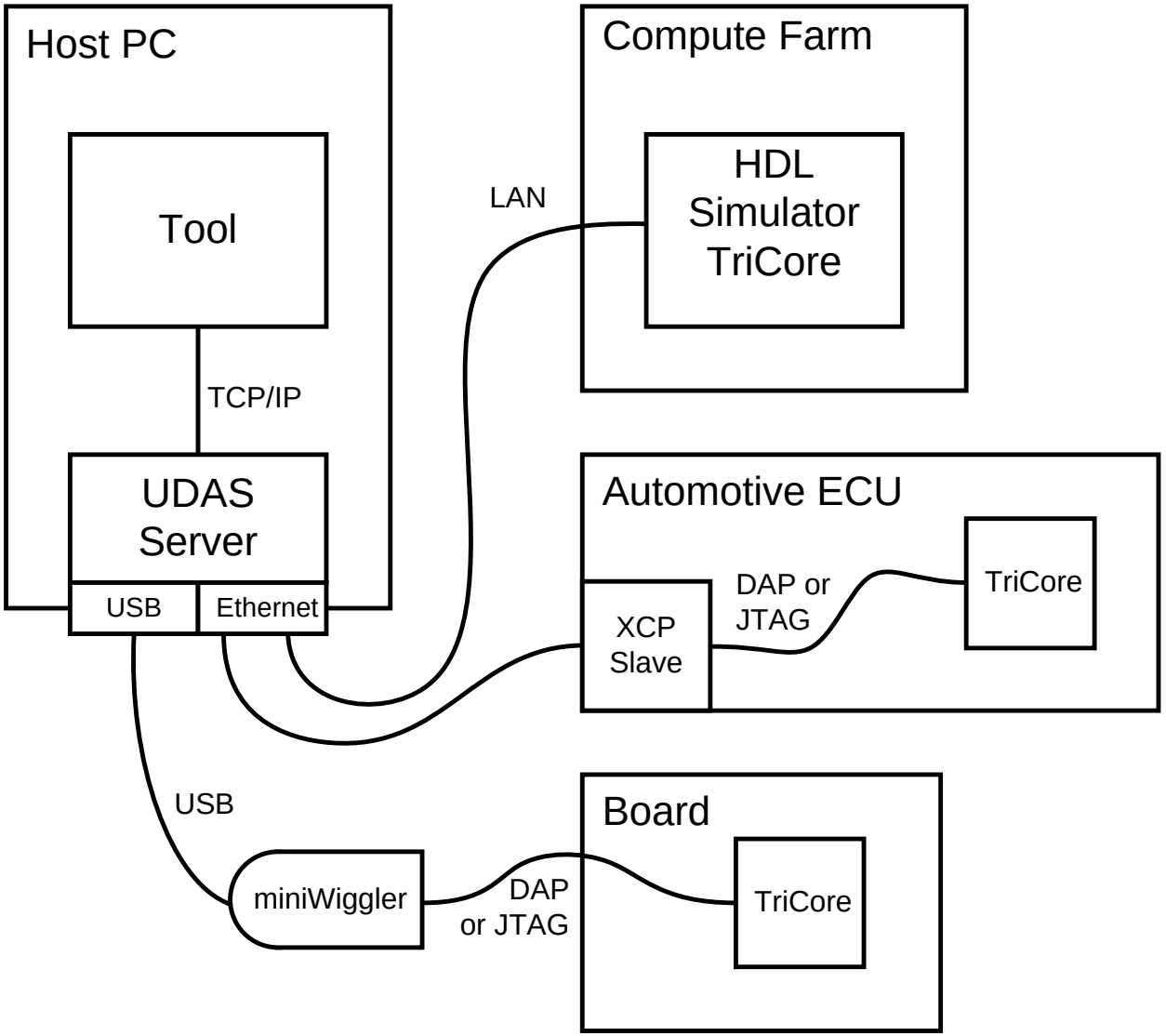
Remote Debugging Use Case Example (1/2)



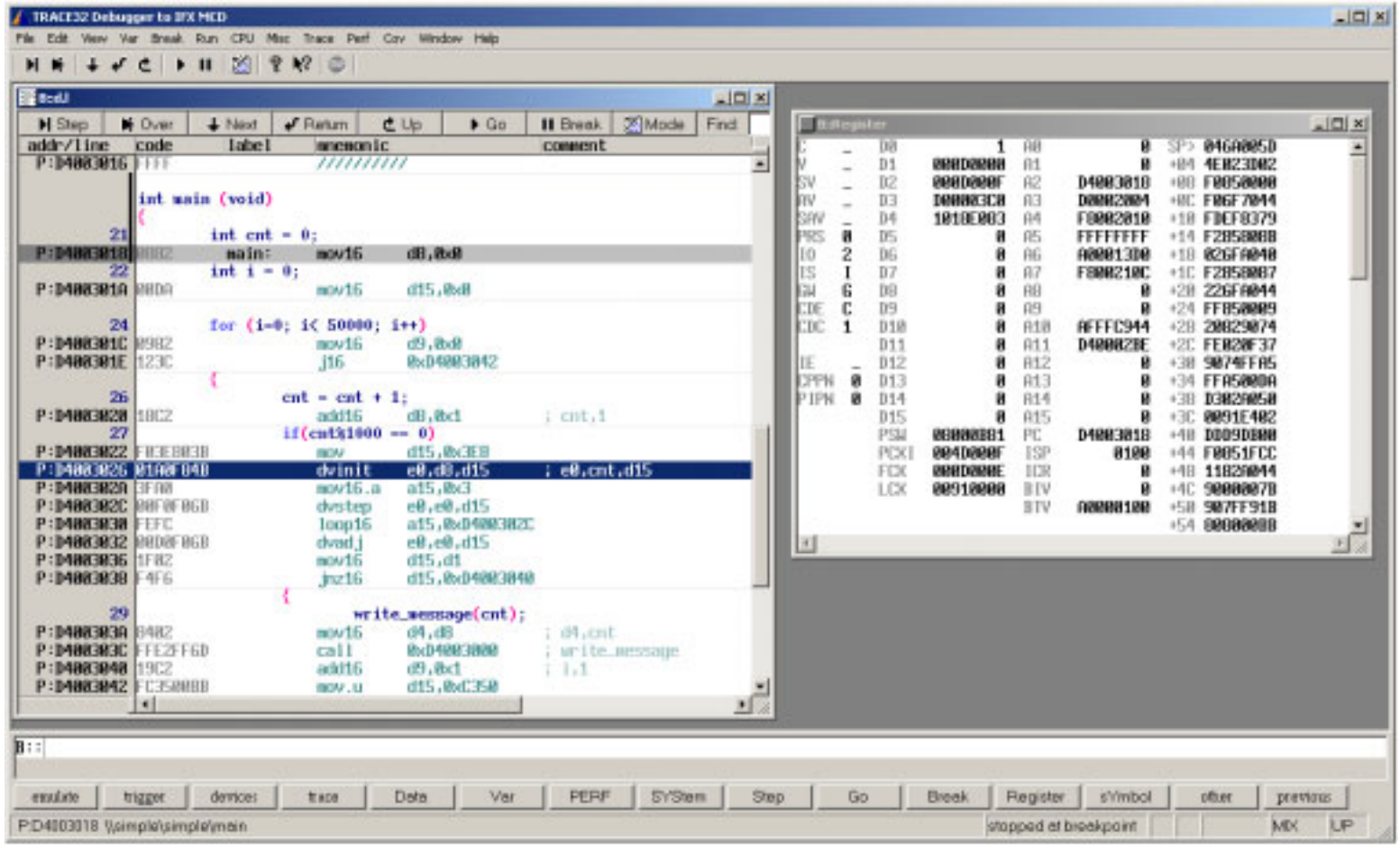
Remote Debugging Use Case Example (2/2)

- 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



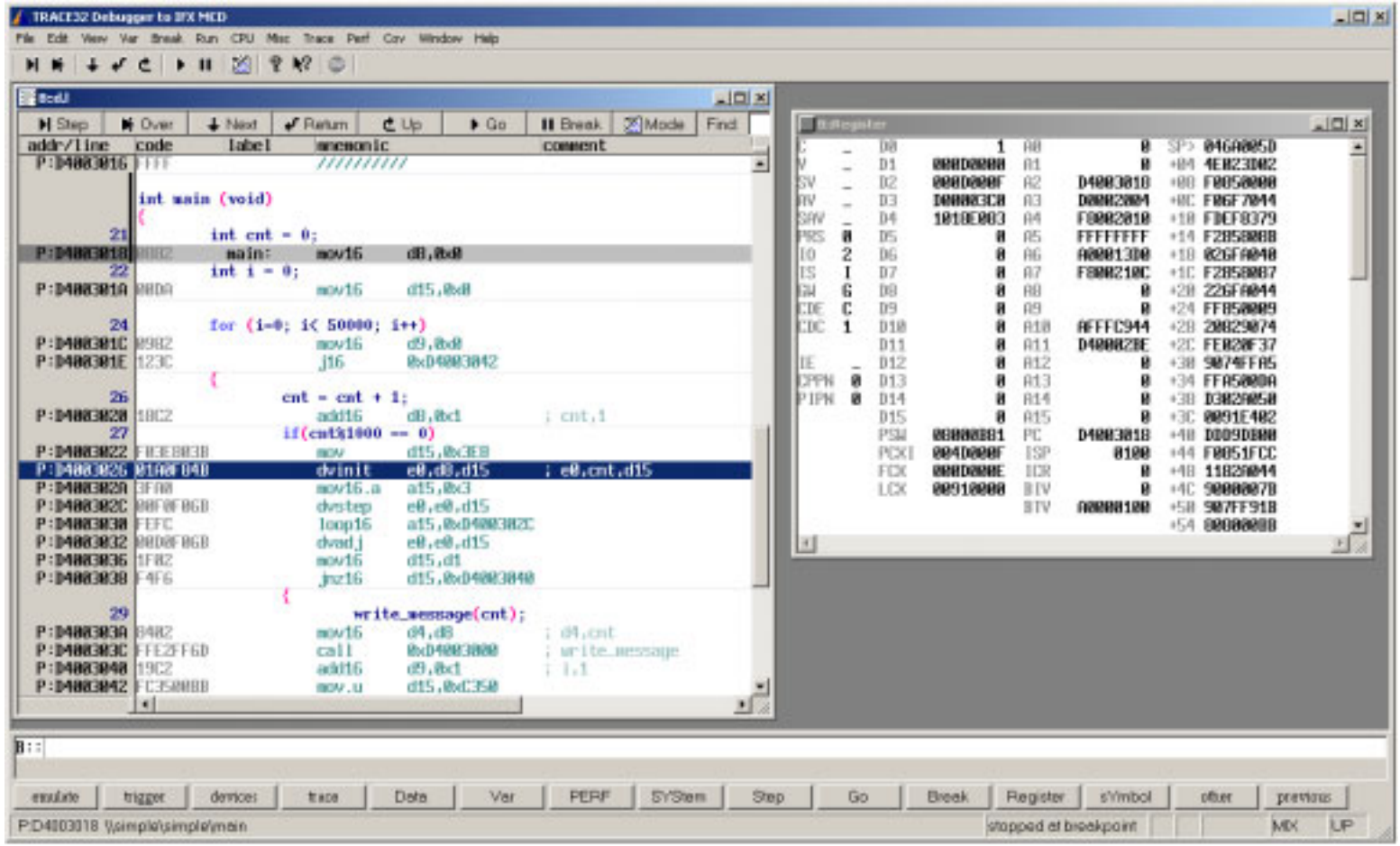
Debugger with ESL Model



TRACE32 Config String:

sys.mcd McdServerName=“C-Model TriCore"

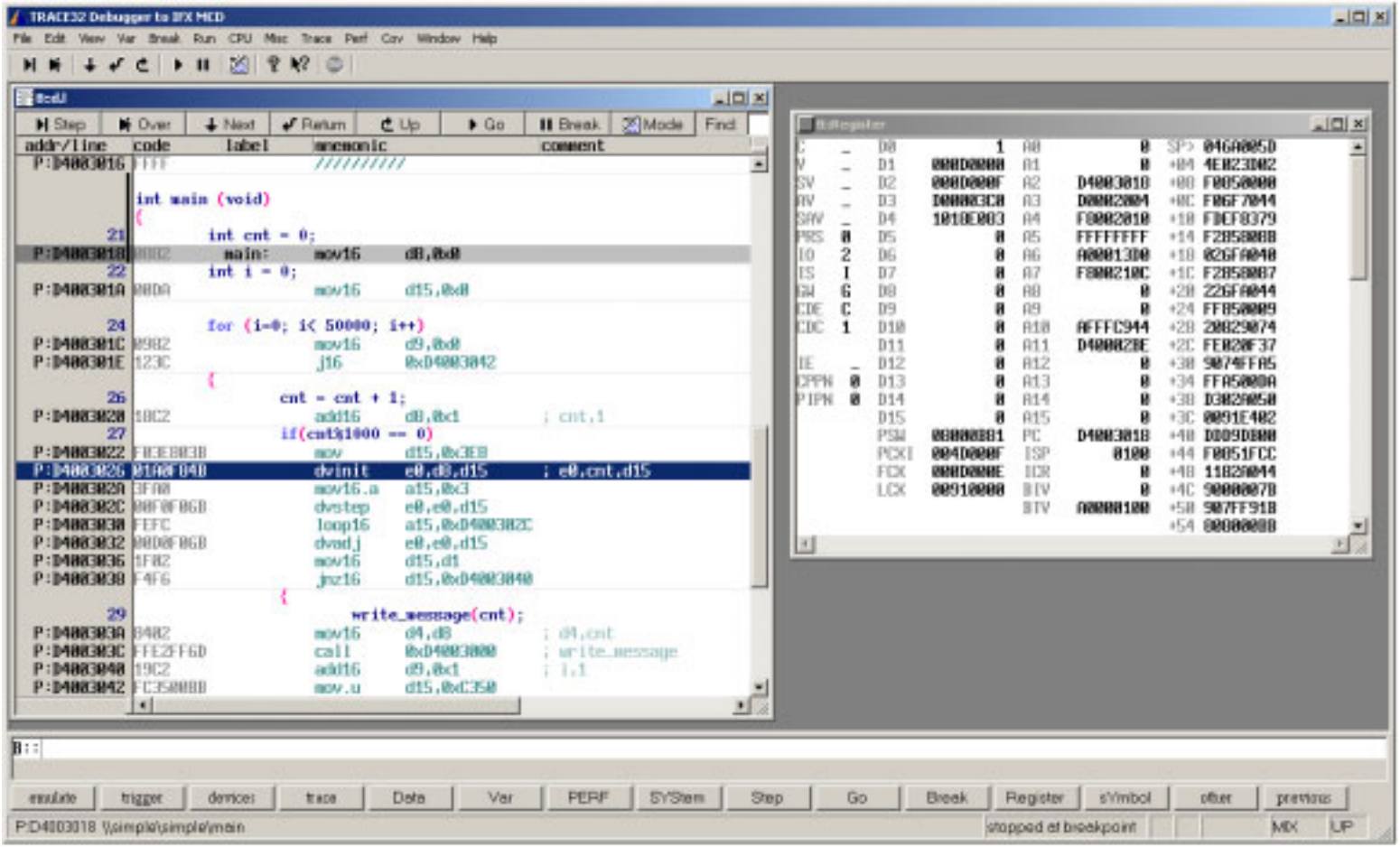
Debugger with Real Device



TRACE32 Config String:

`sys.mcd McdServerName="UDAS"`

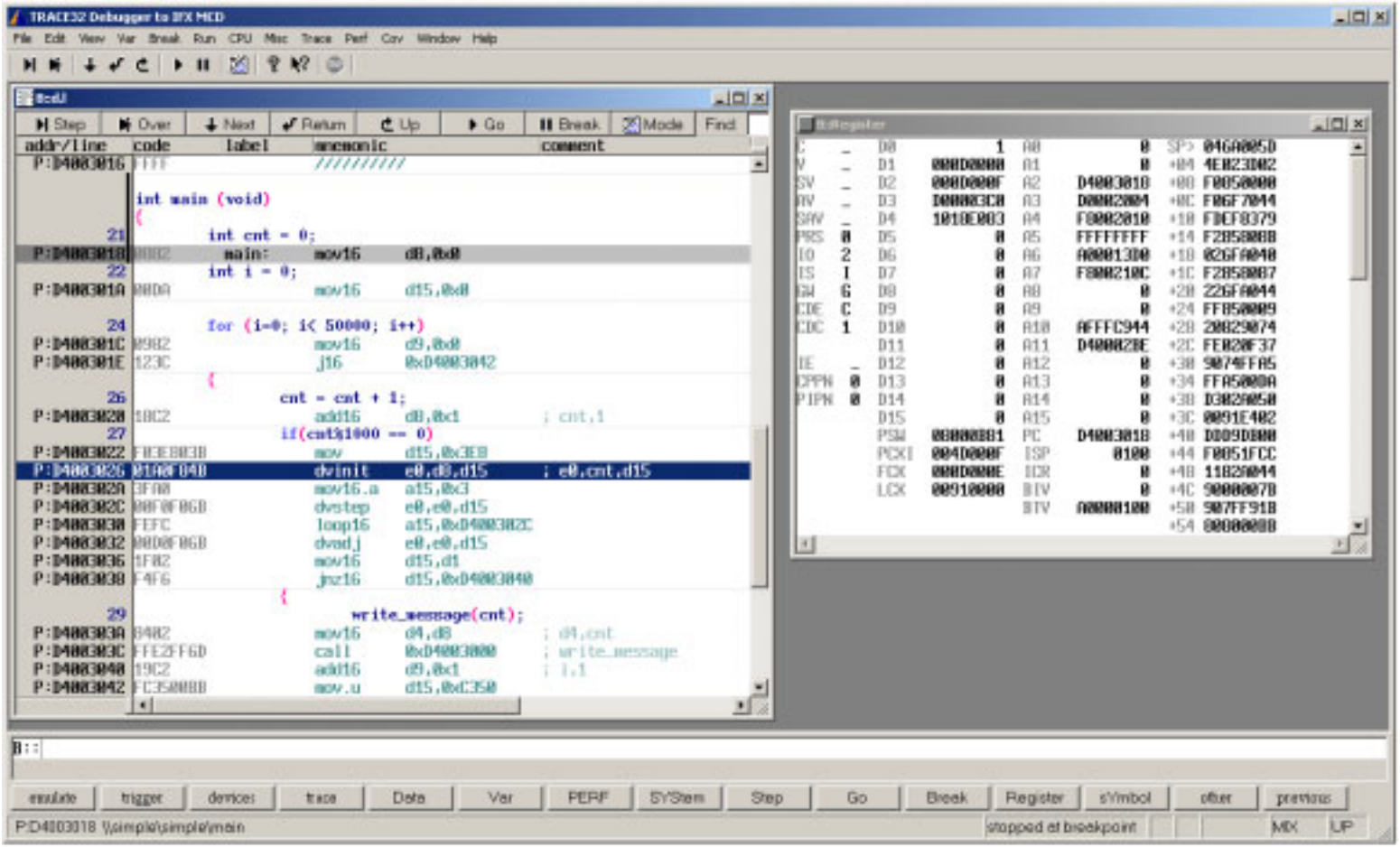
Debugger with Real Device over XCP



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”

Never stop thinking