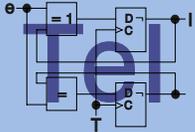


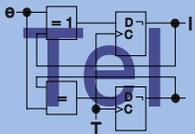
Untersuchungen zu In-Circuit Debug-Möglichkeiten für die Intel XScale Mikroarchitektur

Kai Schicktanz

`ks37@inf.tu-dresden.de`



- Motivation / Software Debugging
- In-Circuit Debugging / JTAG Debug-Schnittstelle
- Intel XScale Architektur (PXA 255)
- Universal Debug Engine (UDE) / Software-Komponenten
- Zusammenfassung



➤ Entwicklungstrend:

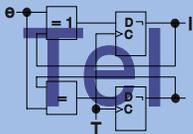
- Technische Systeme werden zunehmend umfangreicher.
- Bei gleichbleibender Fehlerrate steigt dadurch die Fehleranzahl.

➤ Fragen:

- Welche Auswirkungen hat dieser Trend auf das Verhalten aktueller Systeme? (Entdeckbarkeit eines Fehlers / Fehlerwirkung)
- Wie kann man die Fehleranzahl in einem System minimieren?

➤ Lösungsansätze:

- Vermeidung unnötiger Komplexität
- Strukturierung, Automation und Dokumentation im Entwurfsprozeß
- Standardisierung, Formale Methoden der Verifikation, ...
- Intensives und effizientes Testen zur Aufdeckung von Fehlverhalten
- Debugging zur Isolation der Ursache eines Fehlverhaltens



➤ Ziele / Forderungen:

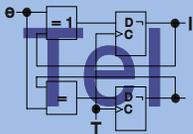
- Beobachtbarkeit eines Fehlers verbessern / Fehler lokalisieren
- Stimulierbarkeit eines Fehlers sicherstellen
- Debug-induzierte Störungen eines Programmablaufes minimieren

➤ Einsetzbare Mittel:

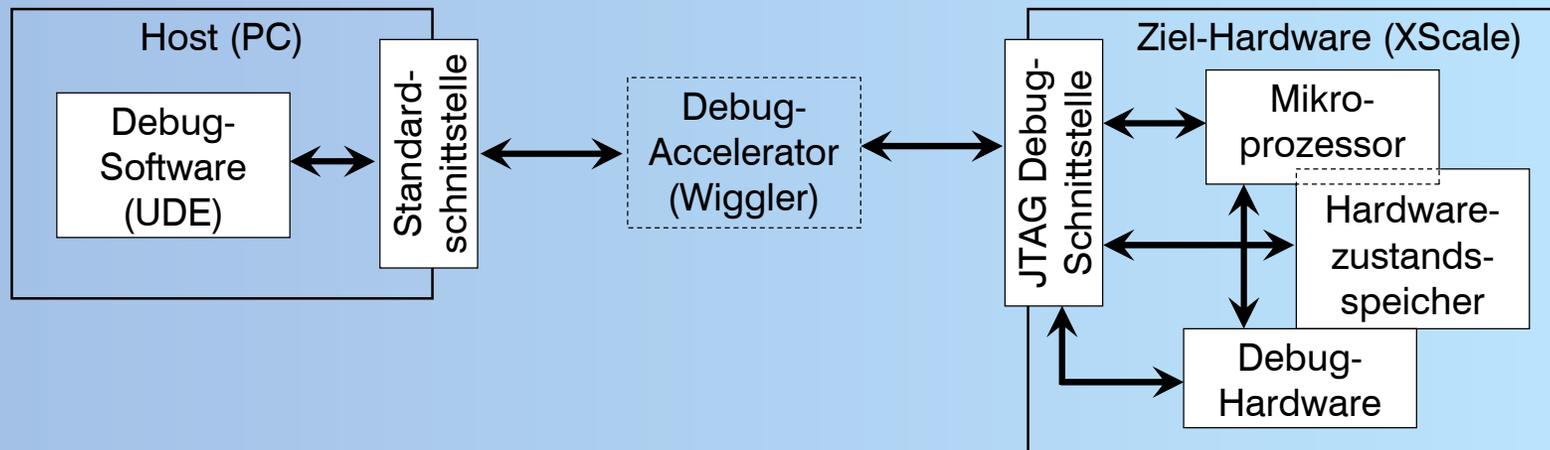
- Instrumentierung eines Programms
- Interaktive Debug-Lösungen:
 - Ausführungsgeschwindigkeit bis zum Stillstand hin reduzieren
 - Zugriff auf die relevanten Zustandsspeicher der genutzten Hardware sichern
 - Reaktionen auf definierte Ereignisse eines Programmablaufes ermöglichen

➤ Voraussetzungen:

- Ausgabekanal
- Interaktive Debug-Lösungen:
 - Eingabekanal
 - Debug-Software, oft auf spezieller Debug-Hardware aufbauend

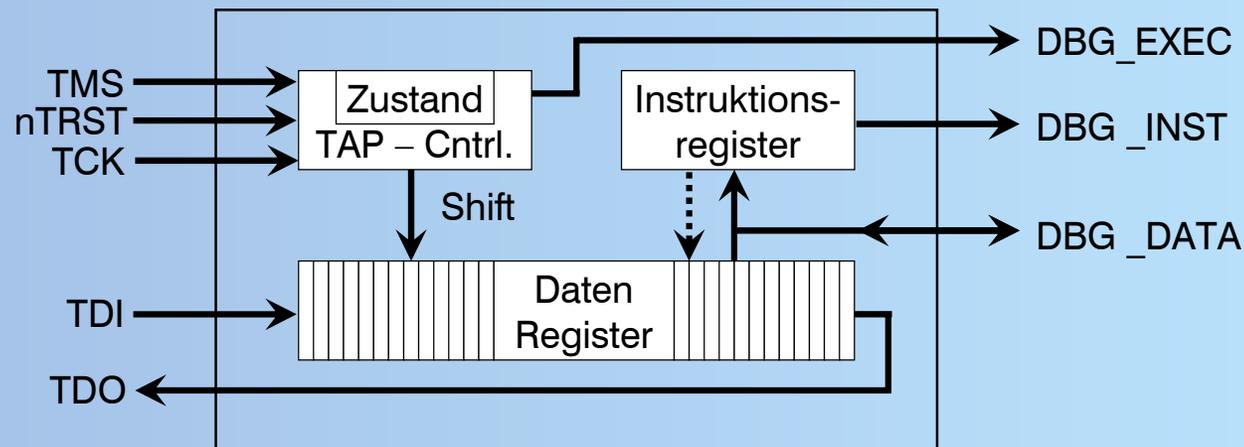


- Allgemeines Hardware-Umfeld von Mikroprozessoren:
 - Ein- und / oder Ausgabekanal häufig nicht vorhanden / nutzbar
 - Ressourcen für die Aufnahme von Debug-Software oft zu begrenzt
 - Debug-induzierte Störungen verhindern oft eine zielgerichtete Fehlersuche (Zeitverhalten, Kommunikation)
- Mögliche (Teil)Lösung:
 - Aufteilung der Debug-Lösung in mehrere unabhängige Module
 - Nutzung hardwareintegrierter Debug-Schnittstellen (JTAG)



➤ Joint Test Action Group (JTAG)

- Hardwareintegrierter Test Access Port (TAP)
- Genormt in IEEE 1149.1
- TAP-Controller (Zustandsautomat), Instruktions- und Daten-Register
- Fünf externe Signale: TCK, TMS, nTRST, TDI, TDO
- JTAG-Befehle: Boundary-Scan, Bypass, ID, ..., spezifische Befehle
- Verbindungen zur Ziel-Hardware (DBG_XXXX) nicht standardisiert



➤ Intel XScale:

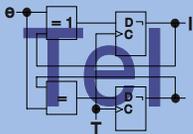
- Mikroprozessorarchitektur, basierend auf der ARM V5TE Architektur
- Baustein applikationsspezifischer Standardprodukte (ASSP)

➤ Strategische Ziele:

- Ersatz der Intel StrongARM Architektur
- Baustein für Intel-Produkte im Embedded-, Mobil- und Netzwerkbereich; z.B. Handhelds, Netzwerk- und Speichergeräte, ...

➤ Historie:

- Okt. 1999: Intel und ARM schließen Lizenzabkommen
- Aug. 2000: Intel kündigt auf dem IDF in San Jose XScale an
- Sep. 2000: Erste verfügbare XScale Hardware (Intel 80200)
- Feb. 2002: Erste, auf XScale basierende Single-Chip Produkte
- Seither : Ausweitung des XScale Produktspektrums

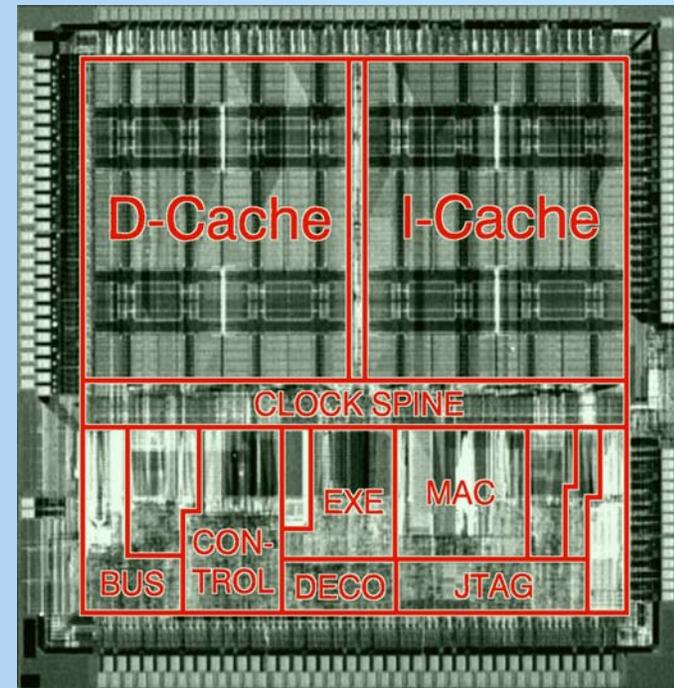


➤ Technische Daten:

- 37 Register, 8 Modi, 2 I-Sets
- Takt : 150 – 1000 MHz
- Leistung : 188 – 1250 MIPS
- P_{Aufnahme} : 100 – 1600 mW
min. 0,1 mW

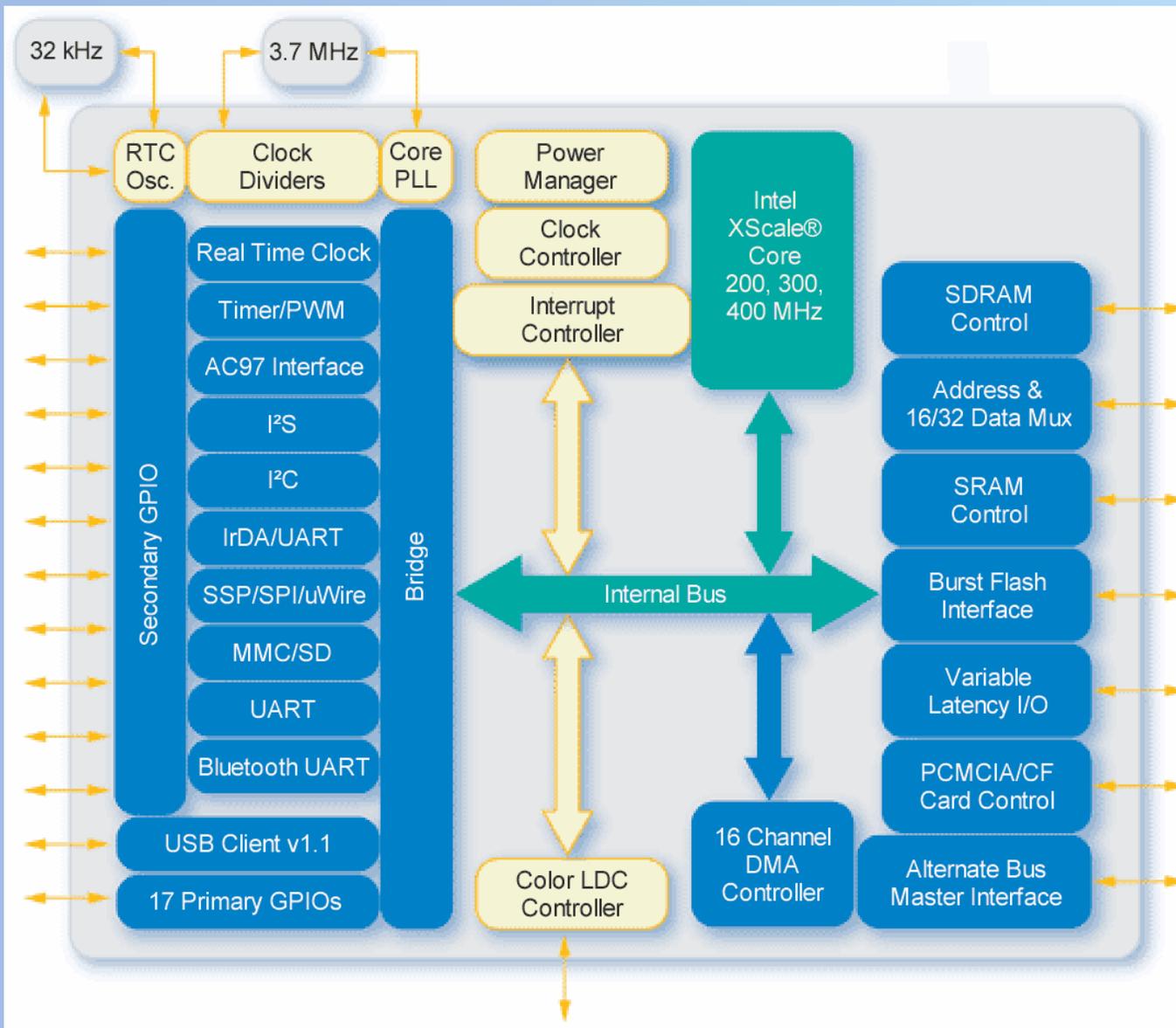
→ „extremely scalable“ → XScale

➤ XScale Prozessorserien:



Intel 80200: 6,5M Trans., 180 nm, 17 mm²

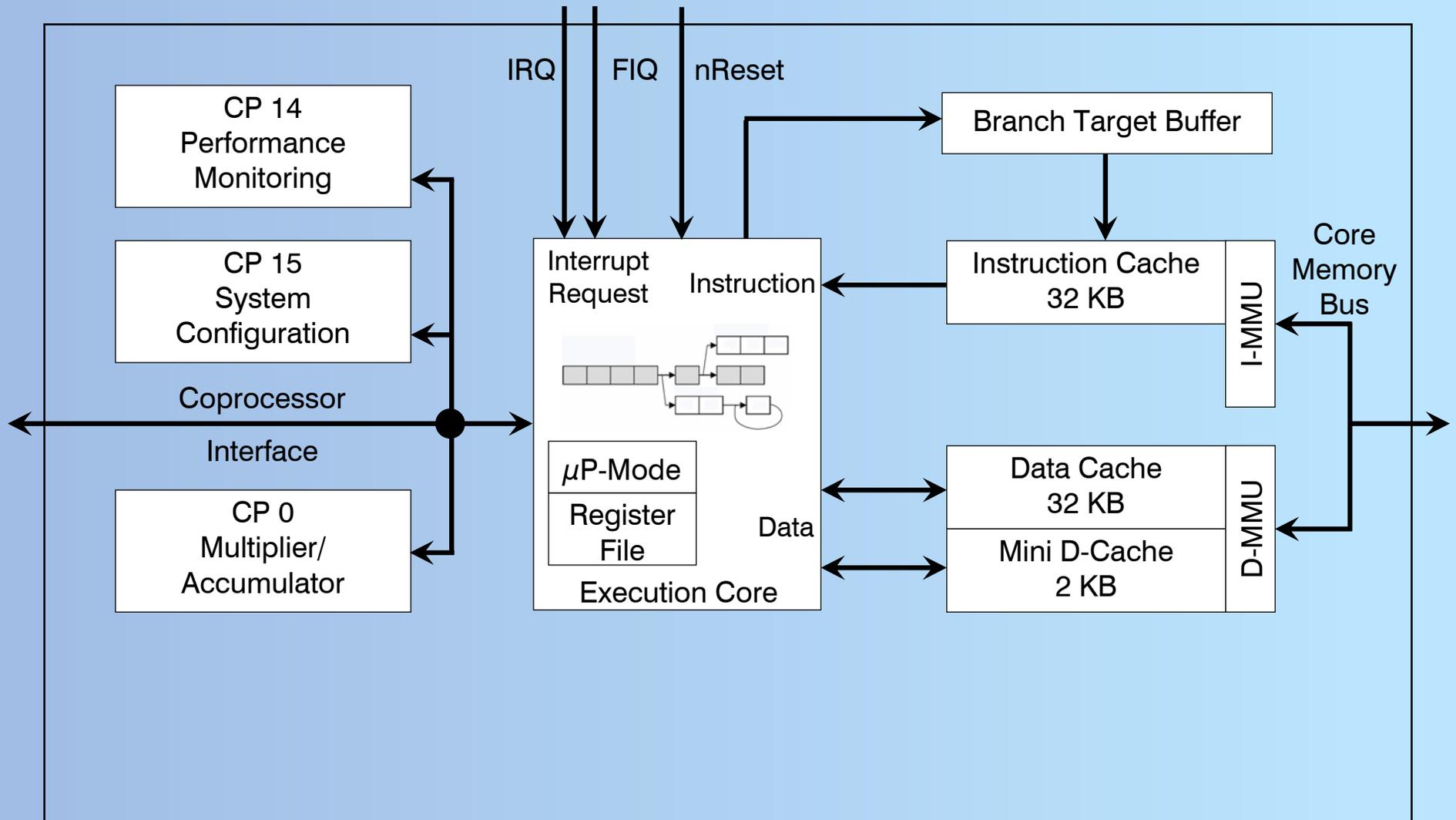
Komplexität	Serie	Einsatzgebiet
	80xxx	Universelle Mikroprozessoren für Applikationen mit hohem Rechenleistungsbedarf.
	ICX	Netzwerkprozessoren zur Paketvermittlung in der Kontrollebene.
	IOP	I/O-Prozessoren für Speicher-, Netzwerk- und Kommunikationsapplikationen.
	PCA	Single-Chip Lösungen für Mobilfunkanwendungen, PDA's und ähnliche Gerätetypen.
	IXP	Netzwerkprozessoren zur parallelen Paketverarbeitung mit Leitungsgeschwindigkeit.



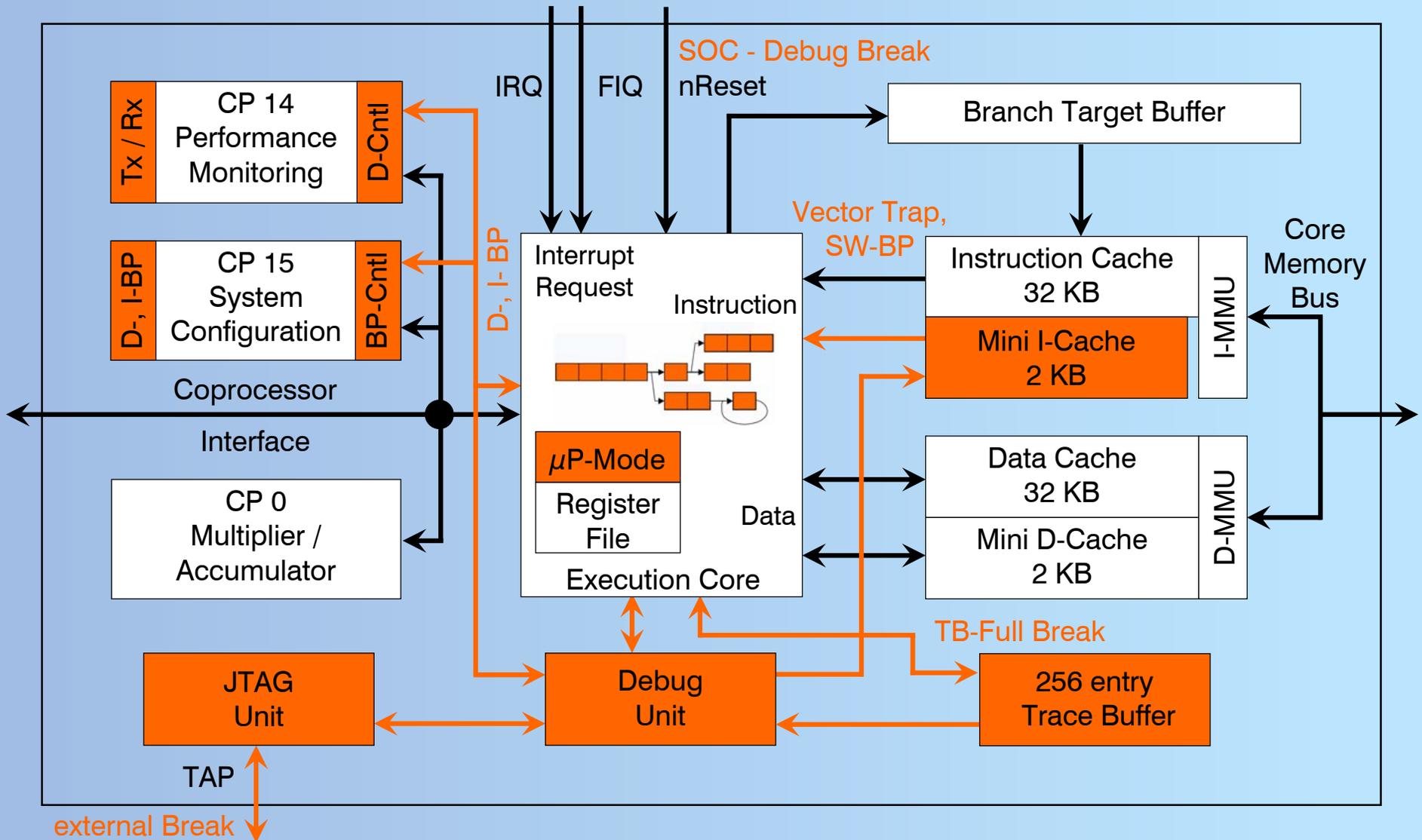
PDA auf der Basis eines PXA 255:

- ext. Speicher:
 - SDRAM
 - SRAM
 - FLASH ...
- ext. Schnittstellen:
 - USB
 - Bluetooth
 - MMC / SD
 - PCMCIA / CF
 - AC97 ...
- LCD / Touchscreen
- Taktversorgung
- Energieversorgung

XScale Prozessorkern

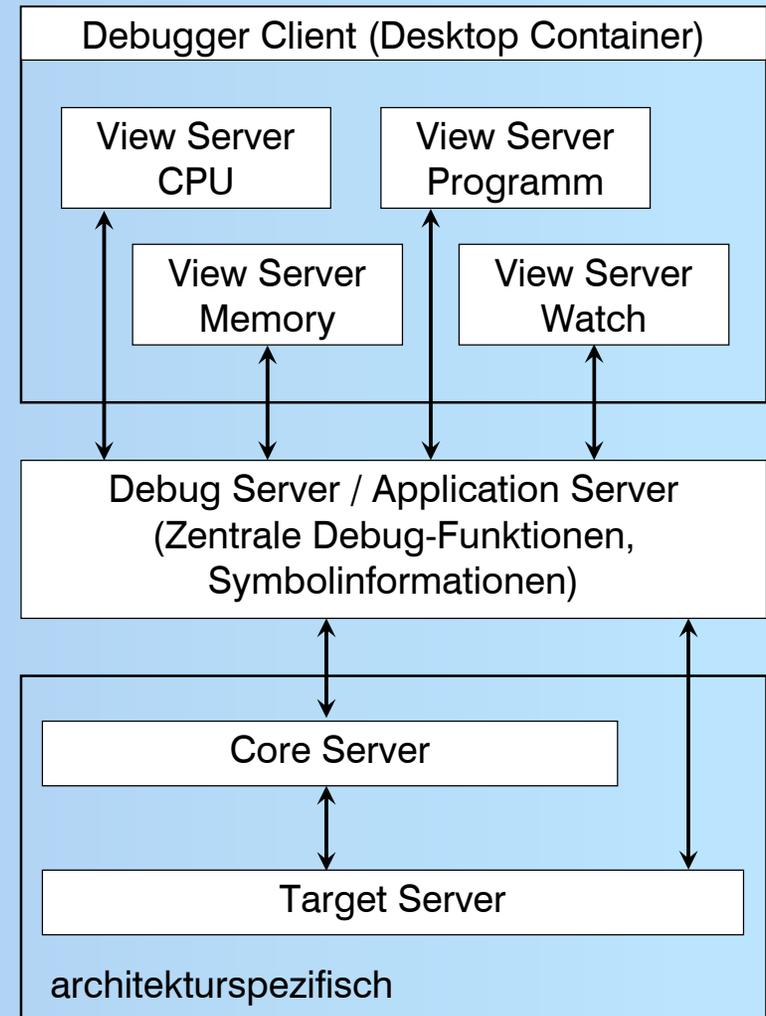


XScale Debughardware



➤ Universal Debug Engine:

- Komponenten-basierend
- Windows-Applikation, COM
- MultiCore tauglich
- Container für generische und architekturenspezifische Module:
 - Sichten auf den Architekturzustand: CPU-, Speicher-, Programm-, Watch-Window, ...
- Debug Server
- Core Server
- Target Server
- Bisher unterstützte Architekturen:
 - ARM: 7 / 9
 - Infineon: TriCore, C166, ...

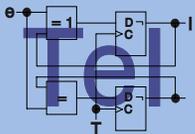


➤ Core Server:

- Hält den Architekturzustand für die Darstellung durch die View Server; Zustand wird mit Hilfe der Funktionalität des Target Servers ermittelt
- Befehlssatzanpassung des Assemblers und Disassemblers
- Ähnlichkeit von ARM und XScale ermöglicht die Wiederverwendung

➤ Target Server:

- Übernimmt die gesamte Kommunikation mit der Ziel-Hardware
- Weitgehend generische Schnittstelle zum Core und Debug Server
 - Speicher lesen / schreiben
 - Prozessorzustand sichern / restaurieren
 - Break- und Watchpoints
- anpaßbares Modul; abhängig vom eingesetzten Debug-Accelerator (Wiggler, UAD, UAD2)
- Spezifisches Protokoll für XScale verhindert die Wiederverwendung



➤ Allgemein:

- Debugging ist unerlässlich für die Softwarequalitätssicherung.
- Für eingebettete Mikroprozessoren sind spezielle Debug-Lösungen verfügbar, diese müssen jedoch architekturenspezifisch angepaßt werden.
- Die Verwendung standardisierter Schnittstellen und modularer Debug-Software reduziert dabei den Anpassungsaufwand.

➤ UDE und Intel XScale:

- Debug-Protokoll der XScale Architektur deutlich vom Protokoll der ARM Architektur verschieden:
 - Anpassung / Erweiterung des Core Servers für ARM für XScale möglich.
 - Target Server ist unter Nutzung vorhandener Schnittstellen weitgehend neu zu implementieren.
- Durch die Verwendung identischer Kerne in den verschiedenen XScale Prozessoren kann der Einsatzbereich des UDE stark erweitert werden.

