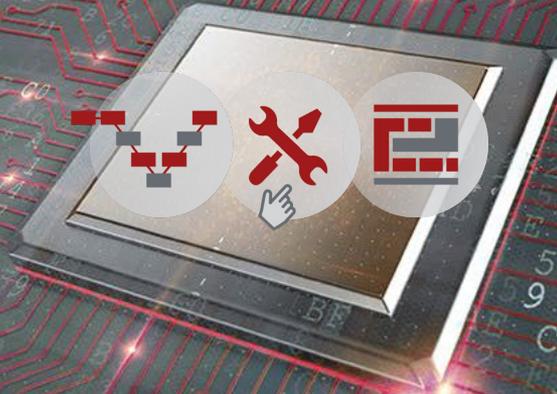


ARAMiS II Abschlussveranstaltung
20.09.2019 Stuttgart



Vorstellung der Use Cases

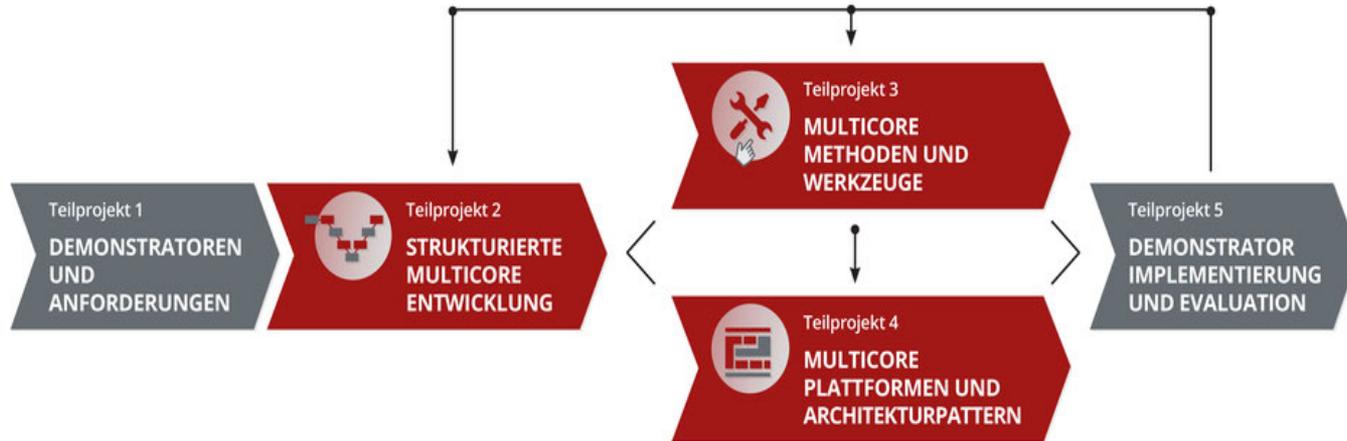
Automotive, Avionik und Industrie Automatisierung

Hatem Miled, Audi AG

GEFÖRDERT VOM



Bundesministerium
für Bildung
und Forschung



Die Demonstration sind aus Automotive, Avionik und Industrieautomatisierung entstanden.
Validierung des Entwicklungsprozesses inkl. Methoden, Tools und Plattformen.

Übersicht-Demonstratoren

AUTOMOTIVE – Demonstratoren

CHASSIS 

POWER-TRAIN 

E-ANTRIEB 

AP5.1
AP5.2
AP5.3

AVIONIK – Demonstratoren

MULTICORE OBSERVER 

TRIEBWERKS-
STEUERUNG 

FLUG-
STEUERUNG 

ECAM 

AP5.4
AP5.5

INDUSTRIEAUTOMATISIERUNG – Demonstratoren

MOBILE
MASCHINEN-
SENSORIK 

PUMPEN-
REGELUNG 

AP5.6

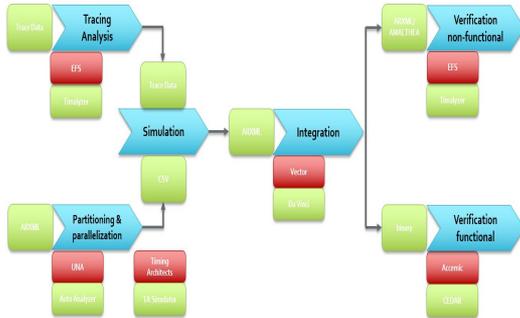


Automotive Chassis demonstrator

Beschreibung des Vorhabens

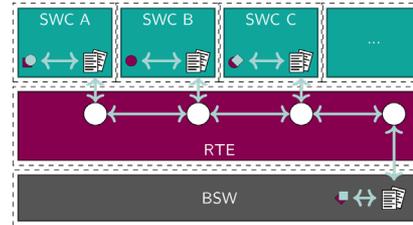
Anwendung von Tools

- Simulation
- Parallelisierung
- Verifikation



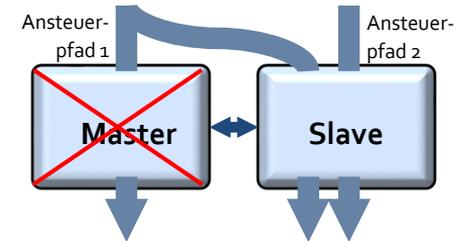
Dynamic RTE Design

- Dynamischer Austausch von Software-Modulen
- Kompatibel zur Classic AUTOSAR RTE



Fail-Operational Konzepte

- Software-Integrations-Plattform
- Zwischen Steuergeräten
- Leistungselektronik-Plattform

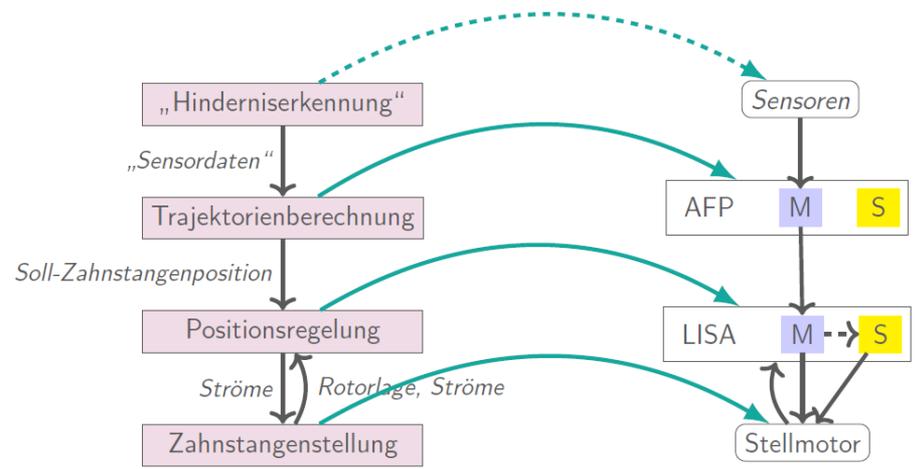
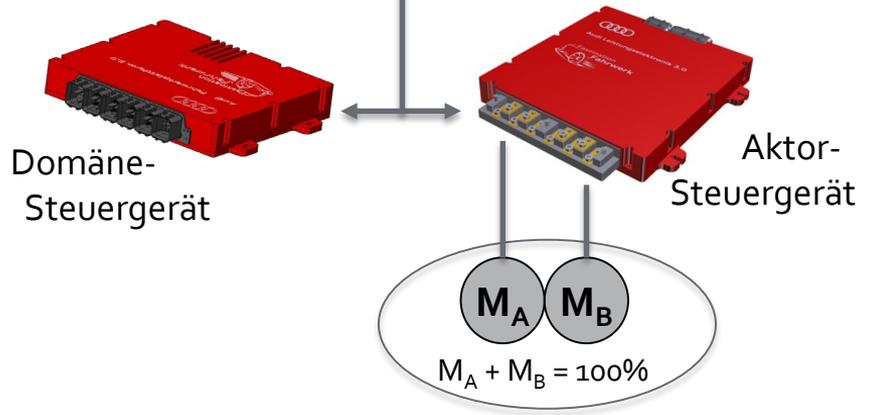


Vorstellung des Demonstrators

manuell



automatisiert



- AP2.3 Timing Analysis as an Aspect for Bottom-Up Approaches
 - Entscheidungskriterien zur Auswahl der Prozessor Architektur
- AP3.6 On Schedule Validation and Load Optimization
 - Begleitende Timing-Absicherung in der Serienentwicklung von Steuergeräte
 - Laufzeitüberwachung von sicherheitskritischen Software Komponenten

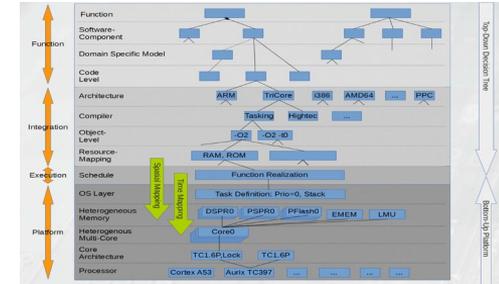


Bild: AP2.3 Entscheidungsbaum

- AP4.1/4.3 Parametric Runtime Environment
- AP4.4 Fail-Operational with Hot-Standby Spare Paths
 - Data Monitoring als Überwachungsmodul im Hot-Standy Konzept

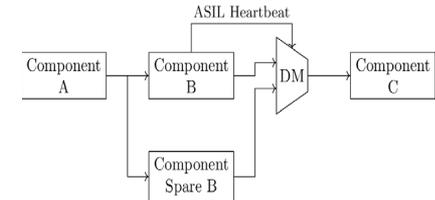


Bild: AP4.4 Hot-Standby Spare Path Konzept



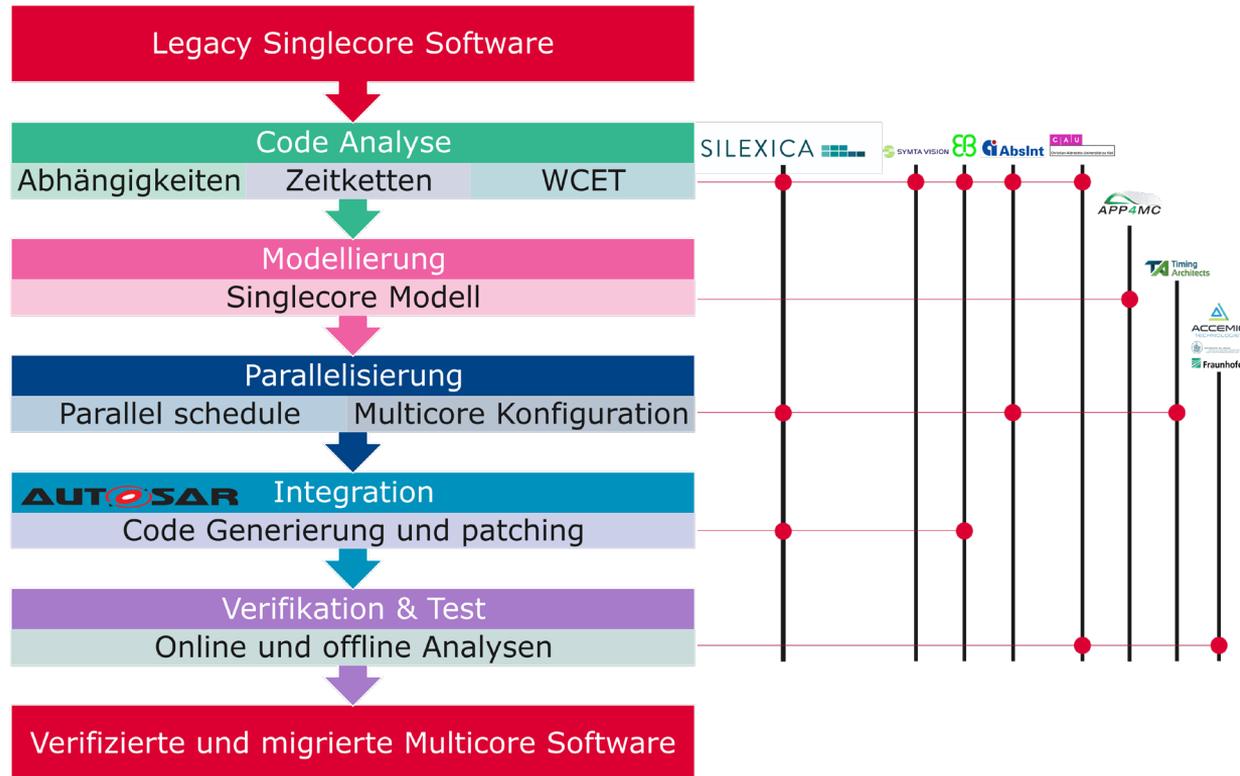
Automotive

Powertrain demonstrator

Beschreibung des Vorhabens

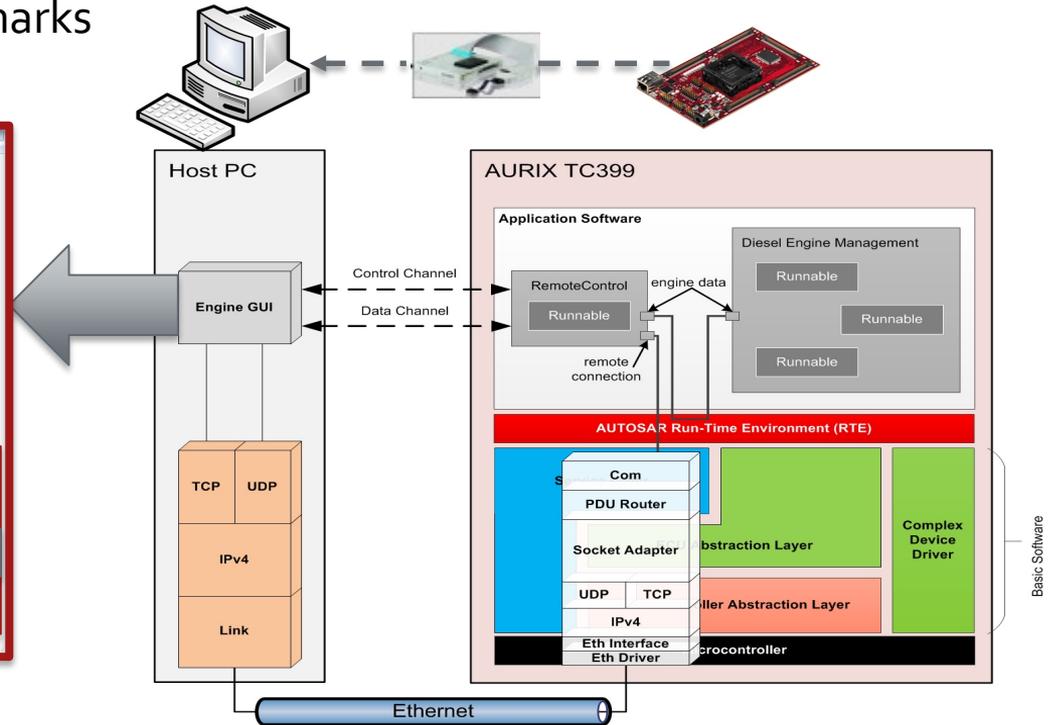
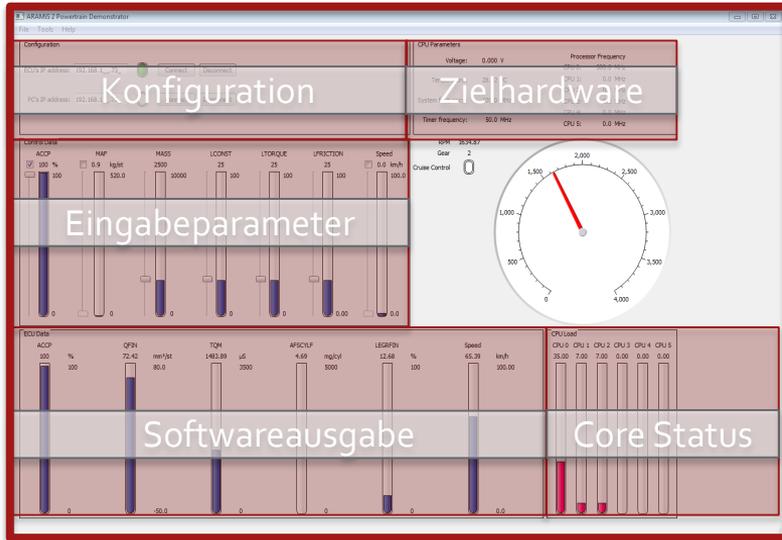
Migration von singlecore zu multicore Software

- Systematischer Ansatz: V-Modell Entwicklung mit Use-Case spezifischer Werkzeugkette
- Anwendung von reverse Engineering Methoden zur multicore Migration von legacy Applikationen
- Nutzerunterstützter, teilautomatisierter Ablauf der Werkzeugkette
- Datenfluss identisch zu singelcore Software
- Deterministische Ausführung der multicore Software

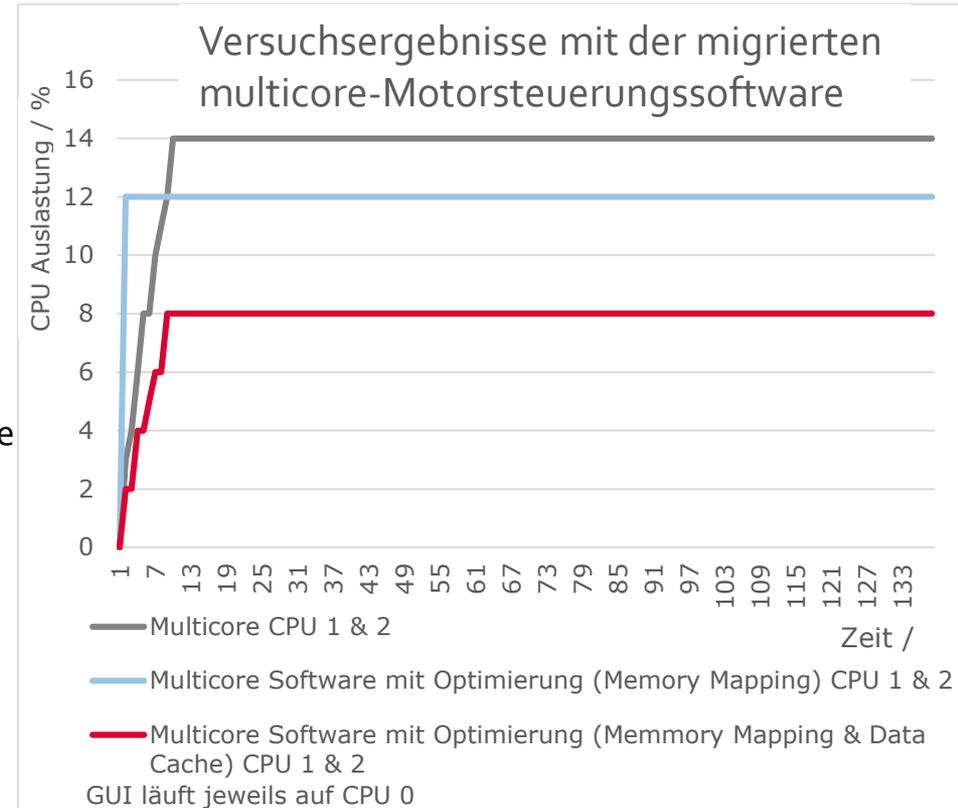


Vorstellung des Demonstrators

Kontrollinterface für Softwarebenchmarks und Versuchsszenarien



- Ergebnisse im Use-Case Powertrain
 - Produktive Werkzeugkette mit Automatisierungsumfängen zur Nutzerunterstützung
 - Erfolgreiche multicore Software Migration einer automotive legacy Steuerungssoftware
 - Erfolgreiche Automatisierung der Runnable-Level-Parallelisierung (Runpar-Algorithmus)
 - Optimierungen der erzeugten Multicore Software
- Identifizierte Entwicklungsbedarfe
 - Interoperabilität Tool, Modellen
 - Standardisierung humanzentrierter Vorgänge
 - Werkzeugkettenautomatisierung
 - Verifikation





Automotive

E-Antrieb demonstrator

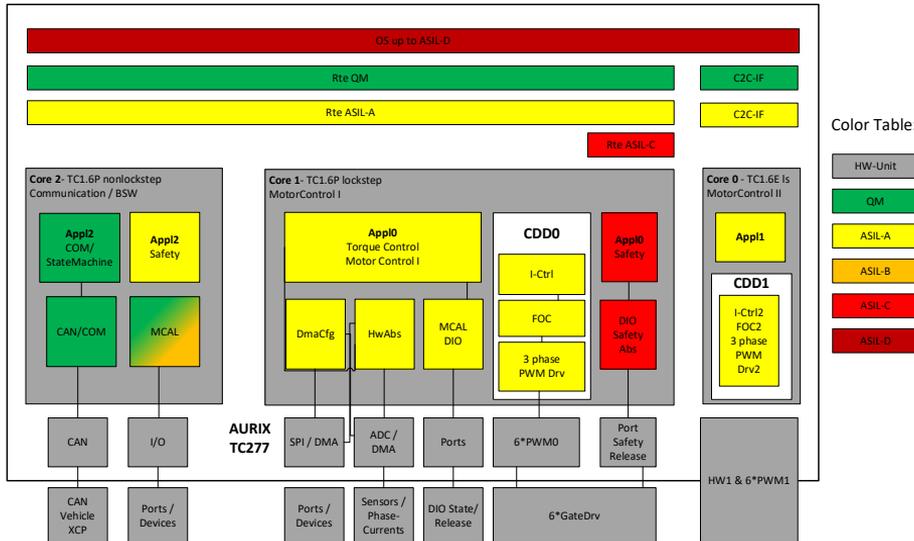
Beschreibung des Vorhabens

- AP5.3 wendet Tools, Prozesse und Best Practices aus ARAMiS II für Anwendungen im Bereich eMobility an
 - Radnabenantriebe
 - Hybridmodule
 - eAchsen
 - in Kombination mit einer zweiten Motorregelung auf gleichem Multicore μ C



Vorstellung des Demonstrators

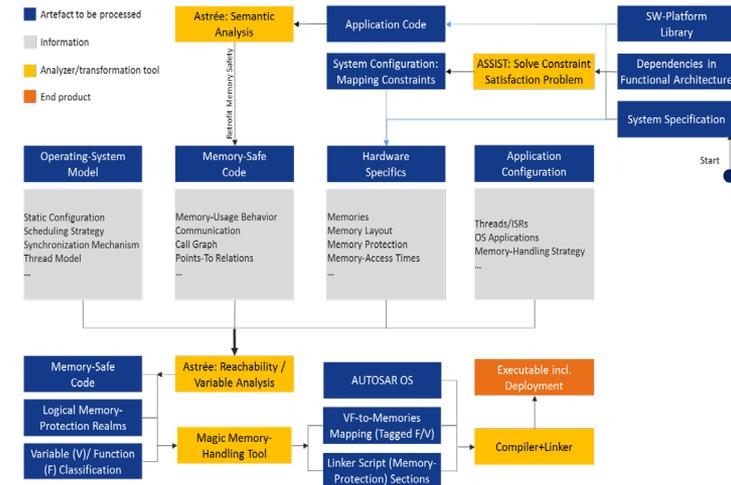
Triple Core – Dual PMSM Configuration



Charakteristika des Demonstrators

- Infineon AURIX TC277 (statt projektspezifischer HW wurde ein Triboard verwendet)
- AUTOSAR 4.0 Architektur
- modellbasierte ASW (Matlab/Simulink/TargetLink)
- Sicherheitskritisches System (ASIL-C), mehrere Speicherpartitionen
- ASW basiert auf generischem Motor Control Baukasten

- Deployment- und Partitionierungsoptimierungen im Single/Dual/Triple-Core Setup
- Halbautomatische Speicherallokierung durch semantische Codeanalyse
- Top-Down Ansatz nach Architekturvorgaben
- AUTOSAR-Erweiterungen der eingesetzten Tools
- Durchführung von WCET-Analysen und –Messungen auf der Demonstrator-Software
- Aufbau eines Fail Operational Demonstrators in AP4.4 zur Verfügbarkeitserhöhung von E-Wheel-Drives





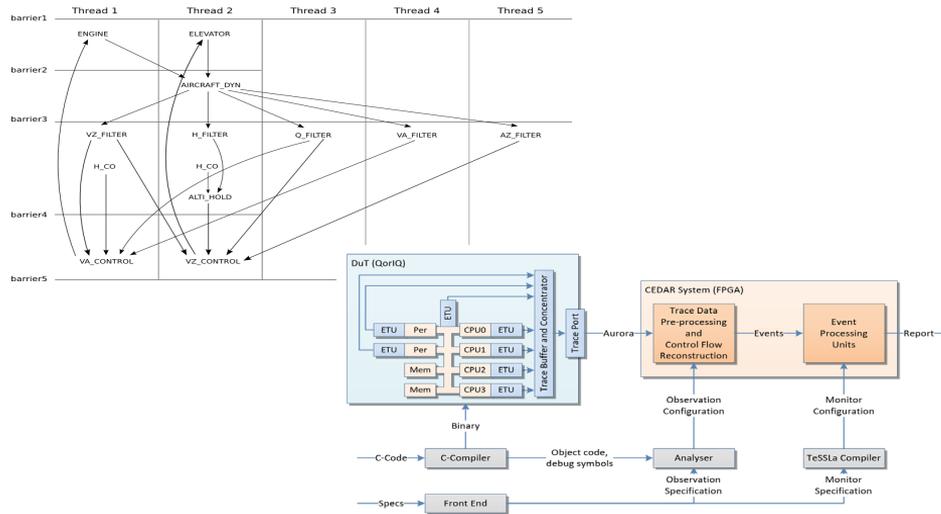
Avionik

Multicore Observer

Beschreibung des Vorhabens

- AP5.4 zielt auf Fortschritte bei der Multicore-Avionik-Zertifizierung ab
- Fokus ist hierbei die begleitende Entwicklung und Evaluierung eines Werkzeuges zur Multicore-Beobachtung
- Herausforderungen:
 - Längerfristige Beobachtung eines Kerns (Funktion und Zeitverhalten)
 - Automatische Bewertung der Beobachtungen
 - Simultane Beobachtung zweier Kerne
- Die Verwendung zweier Kerne geht weit über die aktuellen Ideen zur Multicore-Zertifizierung hinaus, da diese nur einen aktiven Kern vorsehen

Vorstellung des Demonstrators



Charakteristika des Demonstrators

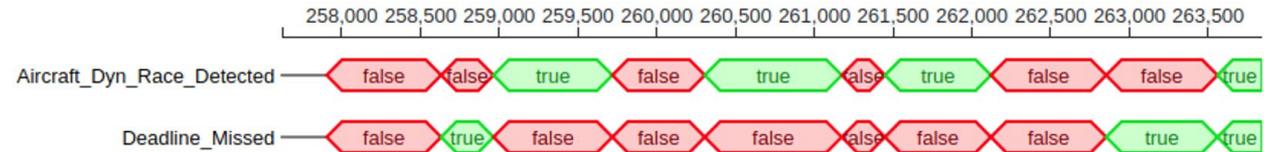
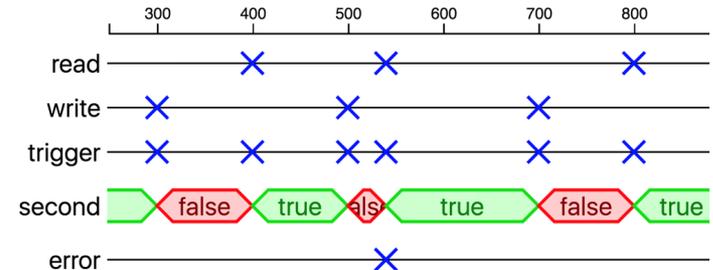
- NXP T1042 Quad-core Prozessor
- Evaluierungsboard von NXP erlaubt Zugriff der Projektpartner
- 2 Prozessoren verwendet und beobachtet
- Generische Benchmark Suite erlaubt Beobachtung von Fehlerfällen



Automatische Höhenkontrolle

- Herkömmlich generierter Scade-Code
- Manuell parallelisiert
- Beobachtung mittels *CEDAR* und *TeSSLa* Spezifikation

- Funktionale Beobachtung möglich
- Zeitliche Beobachtung im Rahmen der Möglichkeiten des Prozessors möglich (eingeschränkte zeitliche Auflösung)
- Erkennung von Fehlerfällen
- Beobachtung zweier Kerne
 - Unabhängige Ereignisse und
 - Beziehungen zwischen Ereignissen beider Kerne möglich
- Vielversprechendes Werkzeug für die zukünftige Multicore-Zertifizierung



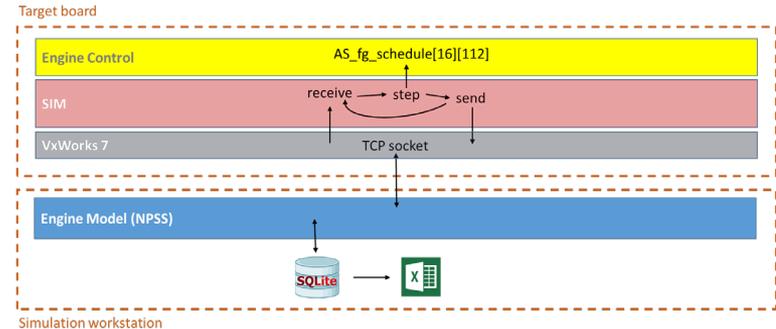


Avionik

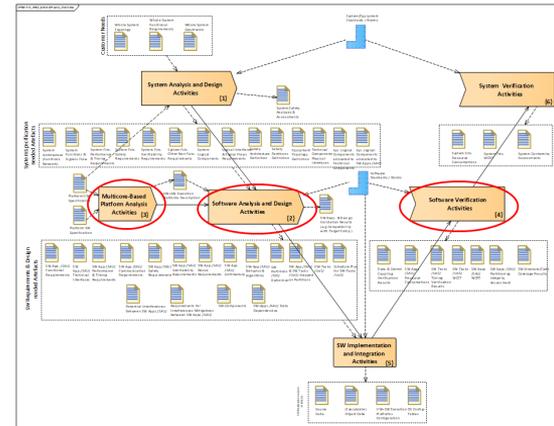
Triebwerkssteuerung

Beschreibung des Vorhabens

- Migration einer bestehenden Triebwerkssteuerung auf einen multi-core
- Fokus auf
 - Plattformaspekte (Interferenzen)
 - Partitionierung, deployment und scheduling
 - Zeitanalyse (WCET)
- Evaluierung der ARAMiS II Lösungen auf einer Demonstrator Plattform



Demonstrator
Funktionsweise



ARAMiS II
Entwicklungsprozess

Vorstellung des Demonstrators

AS	(AS)	(AS)	N/C
VxWorks 7 SMP Kernel			
Core 0	Core 1	Core 2	Core 3
NXP T1040			



Power PC T1040

INTF	INTF	INTF	INTF
Xilinx BSP			
Core 0	Core 1	Core 2	Core 3
ARM A53			



Xilinx US+ ZCU102 Experimentelle Plattform

Power PC T1040 (Triebwerksteuerung)

- 1 x Power PC T1040
- VxWorks 7 SMP
- Applikation: Triebwerkssteuerung (1 – 3 Kerne)
- V+V: TimeWeaver (AbsInt) + Triebwerkssimulation

Xilinx US+ (Plattformanalyse)

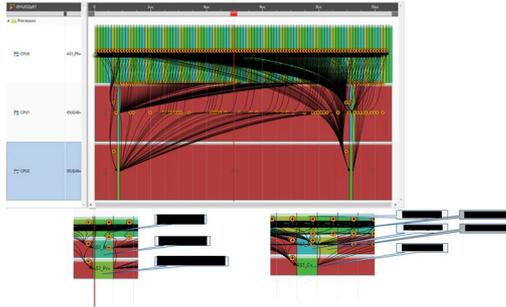
- 1 x ZCU102 (XCZU9EG)
- Bare metal oder SAG Jailhouse
- Applikation: Benchmarkprogramm
- V+V: proprietäre Zeitmessung

Experimentierumgebung:

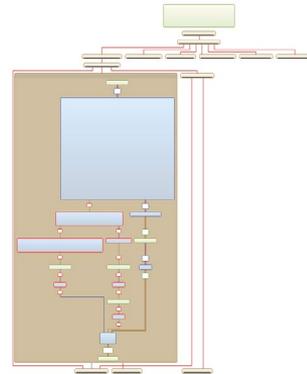
- Reduzierte Triebwerkssteuerung zur ARAMiS II Werkzeuganalyse

ARAMiS II Werkzeuge:

- Jailhouse – Partitionierung, Interferenzanalyse
- AutoAnalyze (UNA) - Partitionierung
- Gropius (CAU) - Partitionierung
- SLX Automotive Prototype (SLX) – Partitionierung, Deployment, Scheduling
- ASSIST (DLR) – Deployment, Scheduling
- TimeWeaver (AbsInt) - WCET



Links: ECS Analyse mit SLX Automotive (Software Design)



Rechts: WCET Analyse mit TimeWeaver (Verifikation)

Analyse der Triebwerkssteuerung mit ARAMiS II Tools
3 verschiedene Multi-core Varianten
Verifikation (funktional + WCET)

- AP2.2 Anwendung des generischen Prozesses auf den Demonstrator
- AP2.3 Methodik zur Analyse von Interferenzen
- AP3.1 Partitionierung von C Code
- AP3.6 Methodik zur Verwendung von TimeWeaver
- AP5.4 Triebwerkssteuerung Demonstrator



Avionik

Flugsteuerung/ECAM

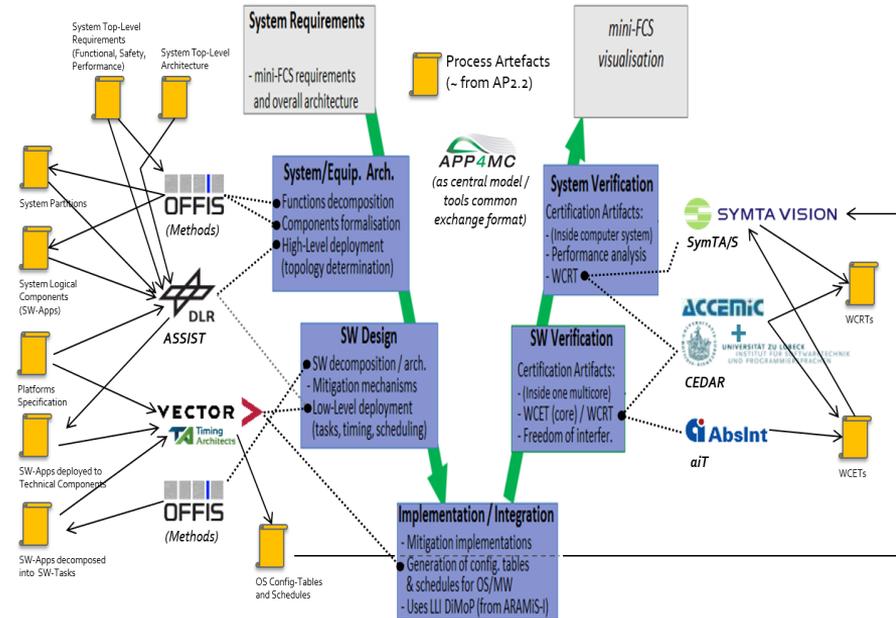
Vorhaben und Ergebnisse - Liebherr

Vorhaben

- Übertragung und Erprobung der Methoden, Prozesse und Tools aus TP2, TP3 und TP4 in einem integrierten, werkzeuggetriebenen Avionik-tauglichen Entwicklungsprozess zur sicheren und effizienten Nutzung von Multicore-Prozessoren
- Entwicklung einer prototypischen Multicore-Plattform mit einem safety-orientierten Multicore-Microcontroller

Ergebnisse

- Adaptierung des ARAMiS-II generischen Prozesses zu einem realistischen Avionik-Entwicklungsprozess
- Selektion, Integration und Erprobung von Software-Tools zur Unterstützung des Entwicklungsprozesses
- Realisierung einer Firmware zur deterministischen Inter-Core Kommunikation; Integration mit dem Liebherr Realtime-OS

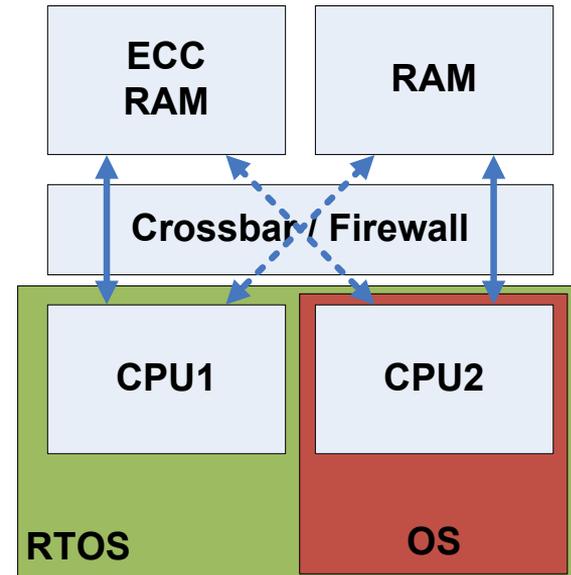


Vorhaben

- Interferenzanalysen von komplexen Multicore-SoC
- Validierung der Prozesse und Methoden aus TP2
- Entwicklung einer prototypischen Multicore-Plattform mit einem safety-orientierten performanten Multicore-SoC

Ergebnisse

- Adaptierung des ARAMiS-II generischen Prozesses zu einem realistischen Avionik-Entwicklungsprozess
- Realisierung einer Firmware für das entwickelte Plattformboard mit einem TI Sitara Multicore-SoC
- Integration von PikeOS inkl. ELinOS von Sysgo auf dem Plattformboard mit Segregation und Partitioning



Vorstellung des Demonstrators

Poster:

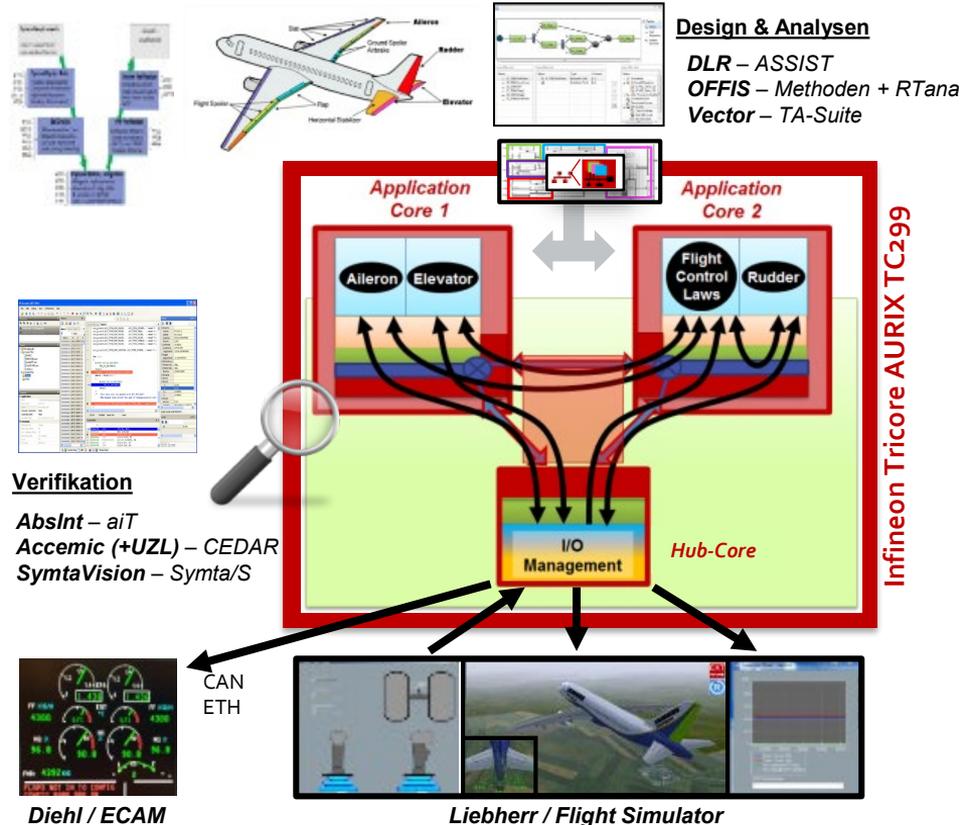
- Darstellung des evaluierten DO178C-kompatiblen Entwicklungsprozesses mit Tool-Unterstützung

PC:

- Demonstration der Unterstützung des Entwicklungsprozesses mit Partner-Tools

Target:

- Evalboard Infineon Tricore AURIX TC299 mit LAOS-MC (Liebherr-Aerospace Operating-System for Multicore)
- Kommunikation der vier mini-FCS Applikationen, auf verschiedene Partitionen und Kerne allokiert (Liebherr)
- Darstellung der Berechnungen der verteilten Software im PC-Flugsimulator FlightGear und in ECAM von Diehl.
- Trennung von sicherheits- und nicht-sicherheitskritischen Informationen auf einzelne Kerne auf Diehl Plattform

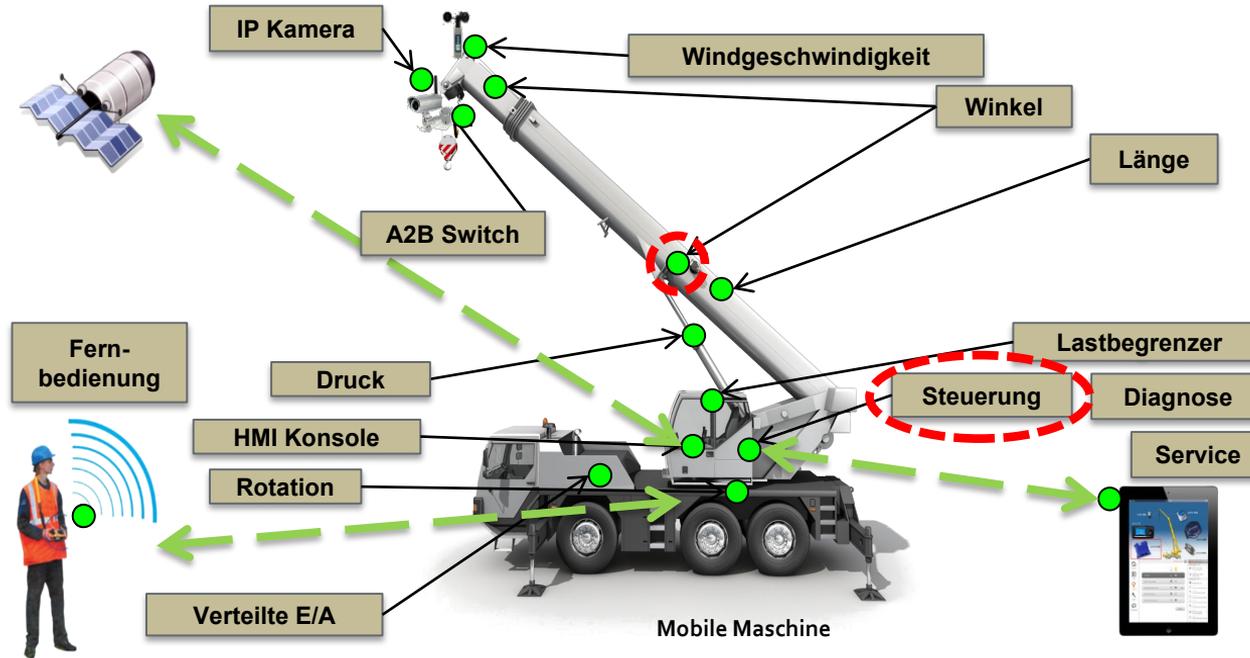




Industrie Automatisierung

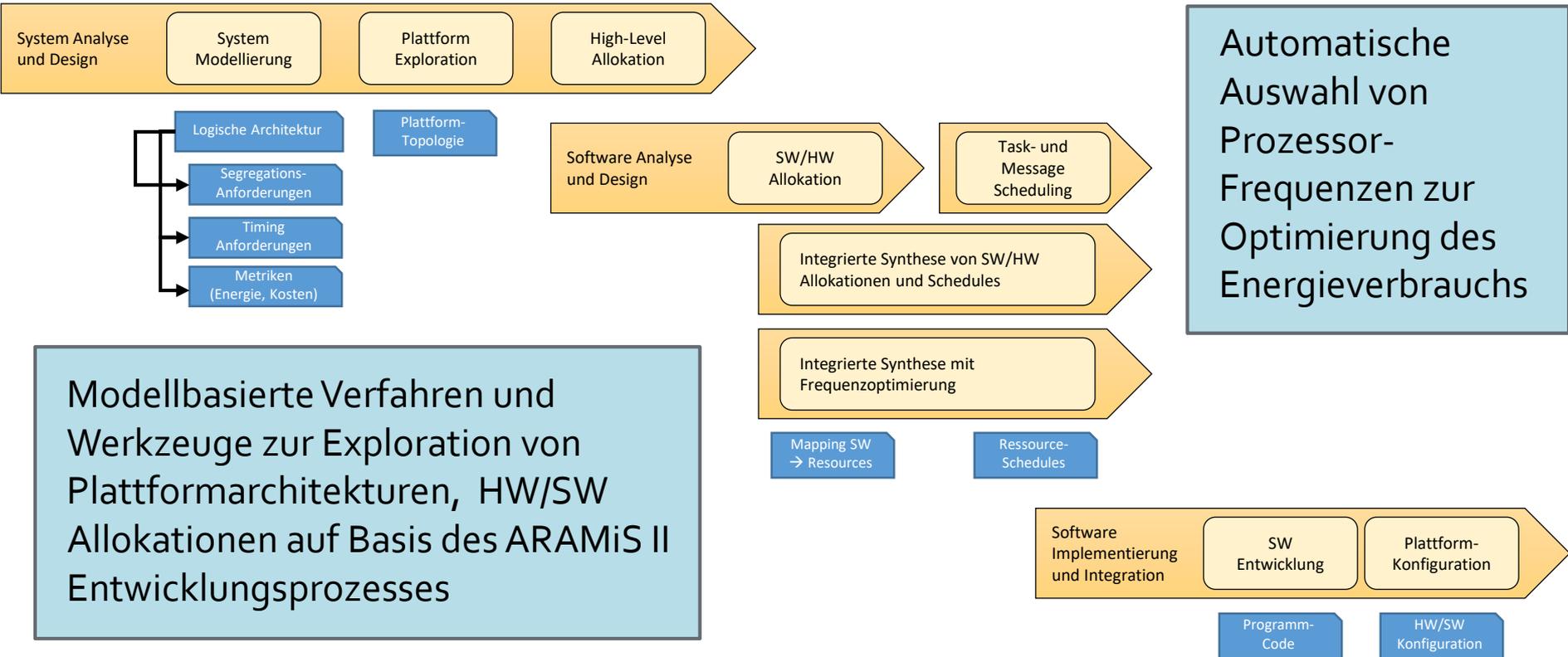
Maschinensensoren

Überblick und Ziele



Ziele von UC56.B

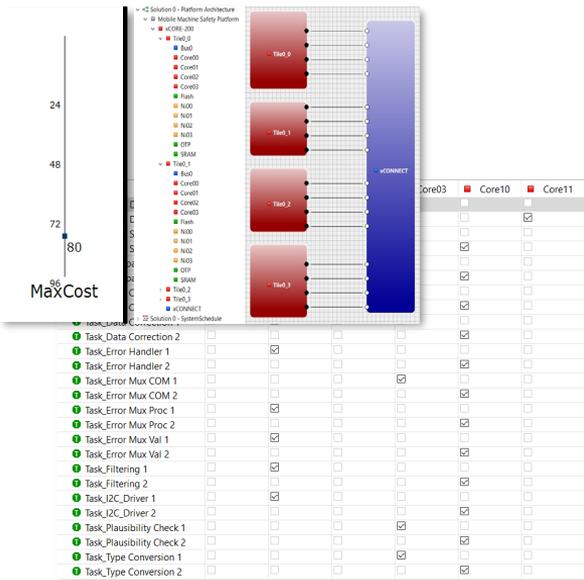
- Deterministische Low Power Multicore-Plattform für zukünftige batteriegestützte Sensorsysteme
- Untersuchung energieeffizienter Systemarchitekturen und Konfigurationen



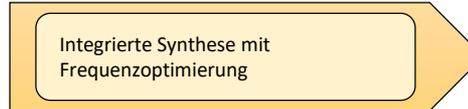
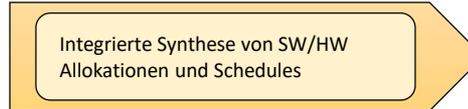
Ergebnisse



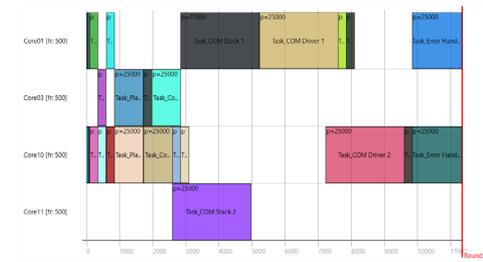
Synthetisierte Plattformarchitektur



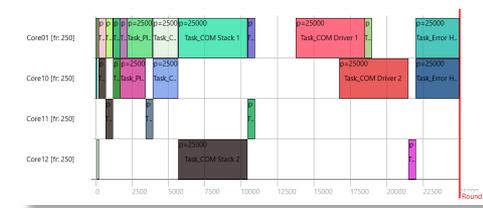
Task → Core Allokation



- Evaluation von modellbasierten Explorationsverfahren und des ARAMIS II Multicore-Prozesses
- Evaluation der XMOS-Plattform (Eignung für Safety-Produkte)



Schedule (f=500 MHz, T= 11,3ms)



Schedule (f=250 MHz, T= 24,8 ms)



```
int main() {
  int i;
  for (i=0; i<10; i++) {
    on t11e[0].core[0]; task1_A(i);
    on t11e[0].core[1]; task2_A(i);
    on t11e[0].core[2]; task3_A(i);
    on t11e[0].core[3]; task4_A(i);
    on t11e[1].core[0]; task1_B(i);
    on t11e[1].core[1]; task2_B(i);
    on t11e[1].core[2]; task3_B(i);
    on t11e[1].core[3]; task4_B(i);
  }
}
```



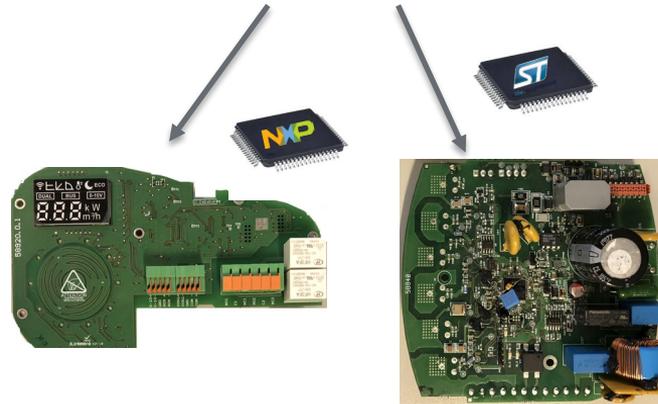
Industrie Automatisierung

Pumpenregelung

Pumpenregelung – Motivation und Ziele

Vision:

- Nächste Generation sicherheitskritischer Pumpenregelungen ist bereit für die Migration auf Multicore



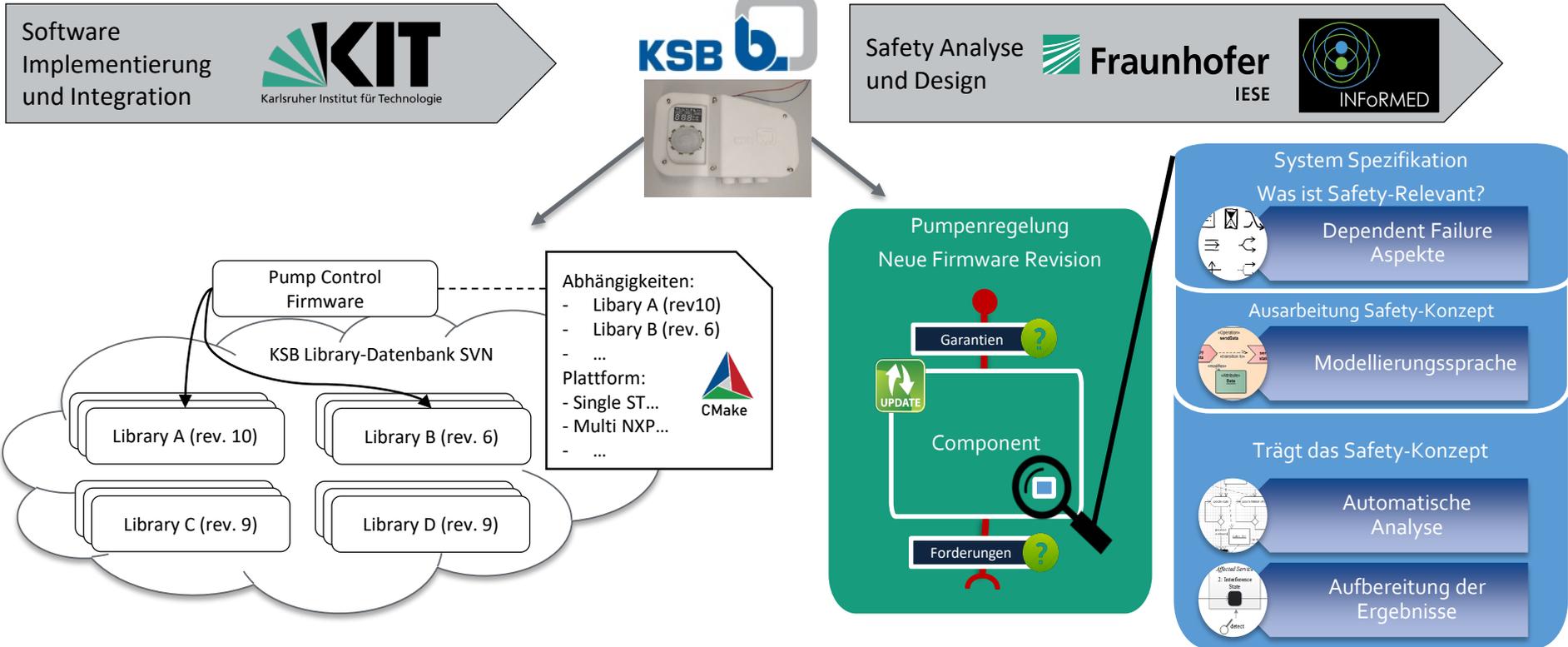
Fokus:

- Komplexitätsherausforderungen einer neuen heterogenen Multi-Controller / Multicore Plattform

Ziele:

- Optimierung der KSB Embedded Plattform
- Aufbau einer Plattform-unabhängigen Werkzeugkette
- Safety Konzept unter Betrachtung von: Dependent Failure und Interferenzen

Vorgehen



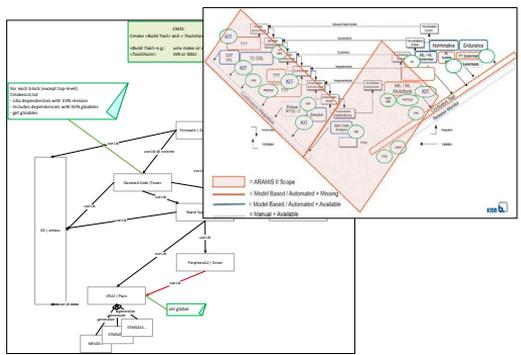
Software Implementierung und Integration



Safety Analyse und Design

Konzept und Prototyp für eine Plattform-unabhängige Werkzeugkette

Prototypischer Safety Konzept incl. Dependent Failure- und Interferenz-Betrachtung



Demonstrator als „Proof of Concept“ der KSB Embedded Plattform V3

