Entwicklungsvorschläge für ISO 26262 konforme MCUs in sicherheitskritischer Avionik

Georg Seifert, Sebastian Hiergeist, Andreas Schwierz

Technische Hochschule Ingolstadt

Agenda

- Motivation
- Aktuelle Integration von AS-MCUs in der Luftfahrt
- Entwicklungsvorschläge für AS-MCUs in sicherheitskritischer Avionik
- Zusammenfassung

Motivation

Allgemeine Anforderungen der Avionik

Sicherheit

- Hohe Integrität der Hardware
  - Kein katastrophales Ereignis
    Fehlerrate = 10^{-9}
- Flugzeug muss sich verhalten wie beabsichtigt
  (continuous safe flight and landing)

Echtzeit

- Deterministisches Verhalten der Applikation
- Quantifizierung von Störeinflüsse

→ SW/HW sind kritische Komponenten

Kaufkraft der Flugzeugindustrie

- Auslieferungen 2013:

AIRBUS **626**

BOEING **648**

- Bisher ausgeliefert:

Eurofighter **543**

- Verkauft in 2017:

Audi **1.878.100**

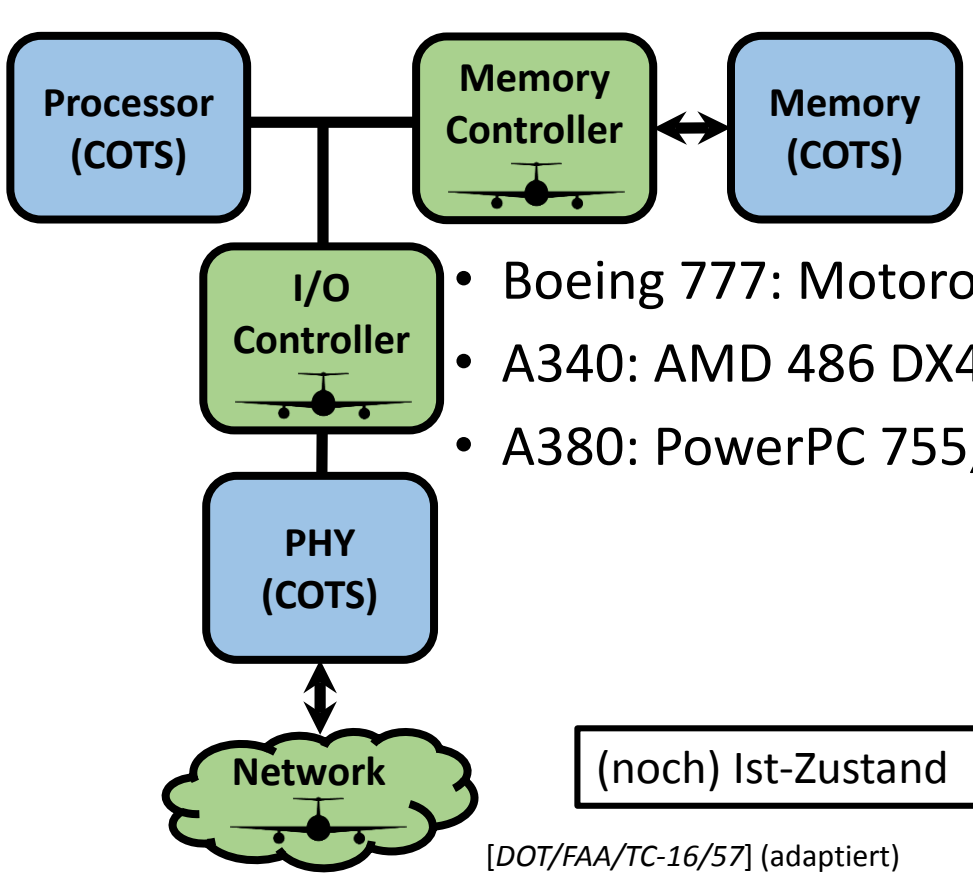
COTS-Komponenten

- “**C**ommercial **O**ff-**T**he-**S**helf (COTS) Component - Component, integrated circuit or subsystem developed by a supplier for multiple customers, whose design and configuration is controlled by the supplier's or industry specification.” [*RTCA DO-254*]

- COTS-Komponenten gibt es nicht umsonst
  - Im Einkauf „billig“,
  - in der Zusicherung des korrekten Verhaltens „teuer“

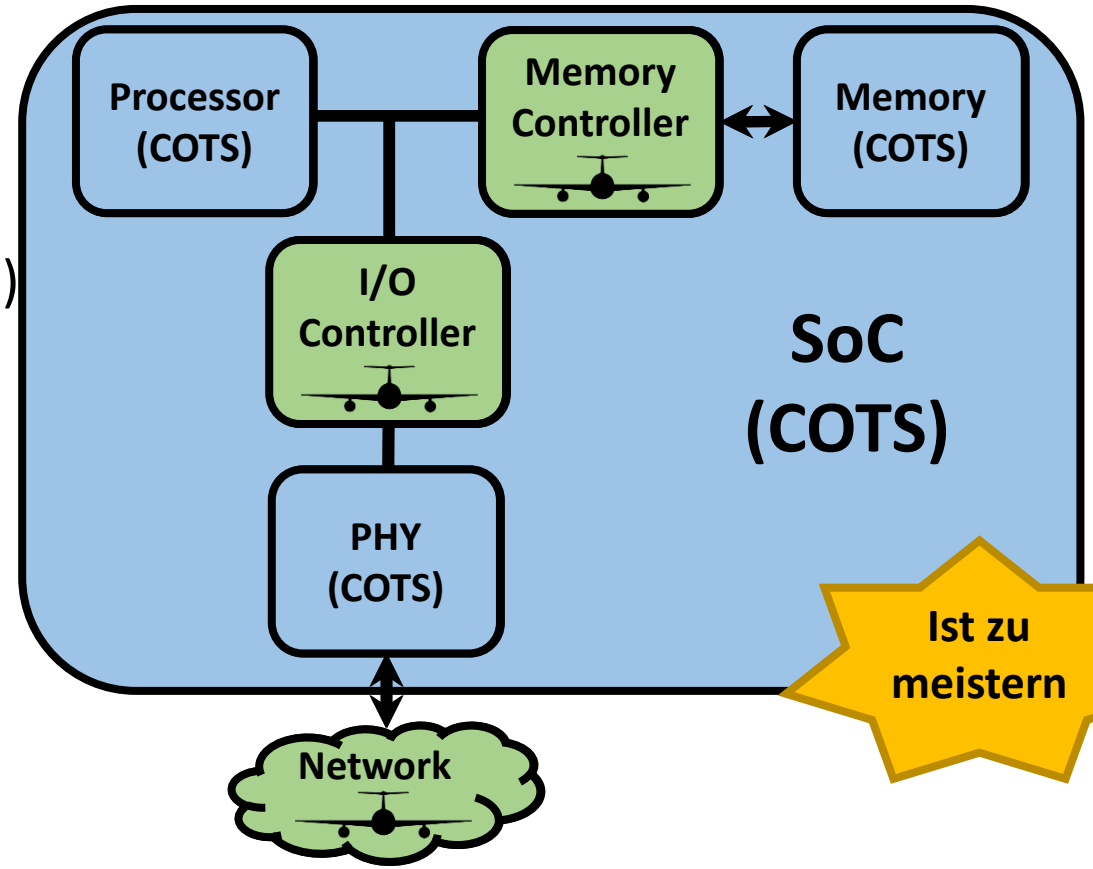
Hintergrund: COTS-Hardware-Komponenten Situation auf dem Halbleitermarkt

- „Einfache“ MPUs in hoch-sicherheitskritische Avionik
- Zunahme der funktionalen Integration → Komplexe MCUs



- Boeing 777: Motorola 68040, (1990)
- A340: AMD 486 DX4, (1995)
- A380: PowerPC 755, (2000)

(noch) Ist-Zustand



[*DOT/FAA/TC-16/57*] (adaptiert)

Wahl eines COTS-MCU

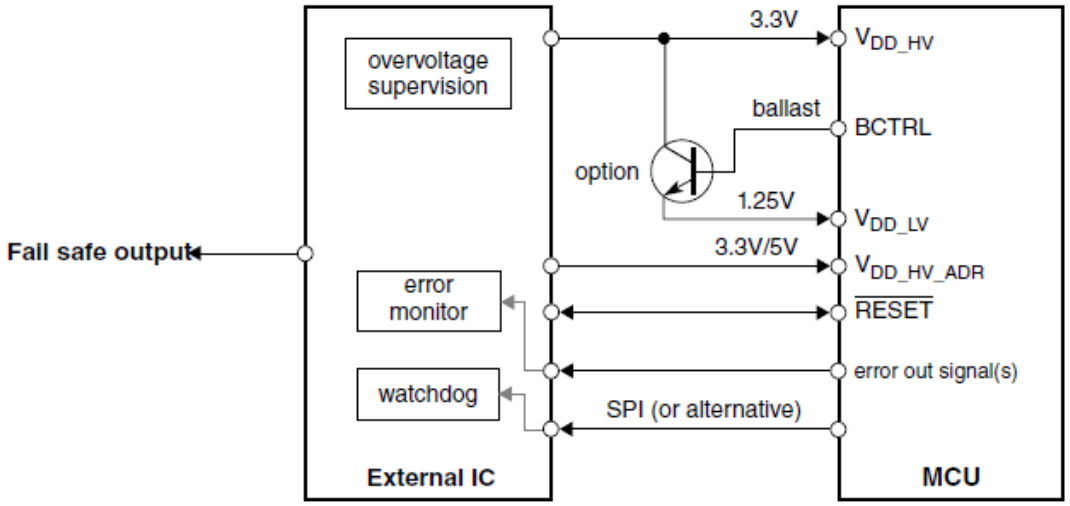
- Best Fit: Komponente entwickelt für
  - Safety-Domäne mit höherer Kaufkraft
  - Echtzeitkritische Anwendungen
  - Langzeitverfügbarkeit der Komponenten
- Automotive Safety MCU (AS-MCU)
  - integrierte Sicherheitsarchitektur
  - geeignet für Anwendungen nach ISO 26262 ASIL-D

➢Integration des AS-MCU in den sicherheitsrelevanten Avionik-Systemkontext?

Aktuelle Integration von AS-MCUs in der Luftfahrt

Automotive
Fail Safe: Fail Stop

- Zielanwendungen des MCU: Airbag, ABS, ESC



[MPC5744PSM Safety Manual for MPC5744P]

MCU mit externer Überwachung

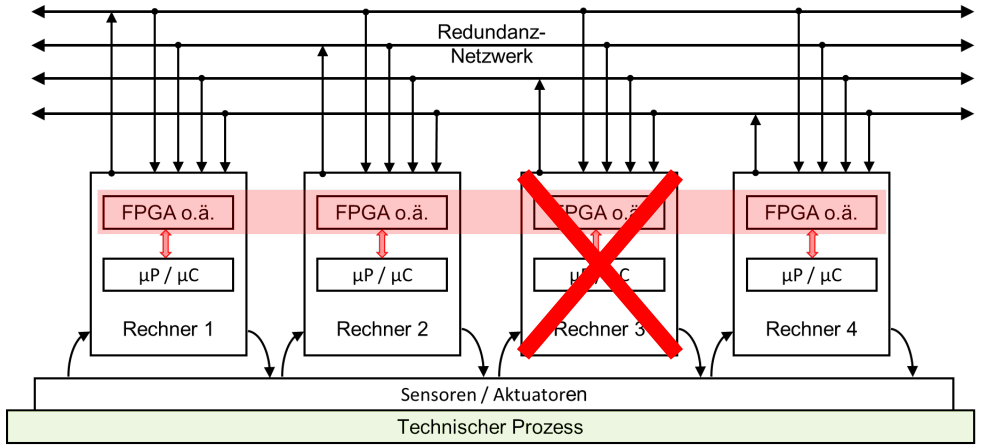
Pannenbucht

Erkennen von Fehlerzuständen

→ Überführen in einen sicheren Zustand

Luftfahrt
Fail Safe: Fail Operational

- Zielanwendungen des MCU: FCC (Safety & Echtzeit)



Redundanznetzwerk basierend auf FPGA

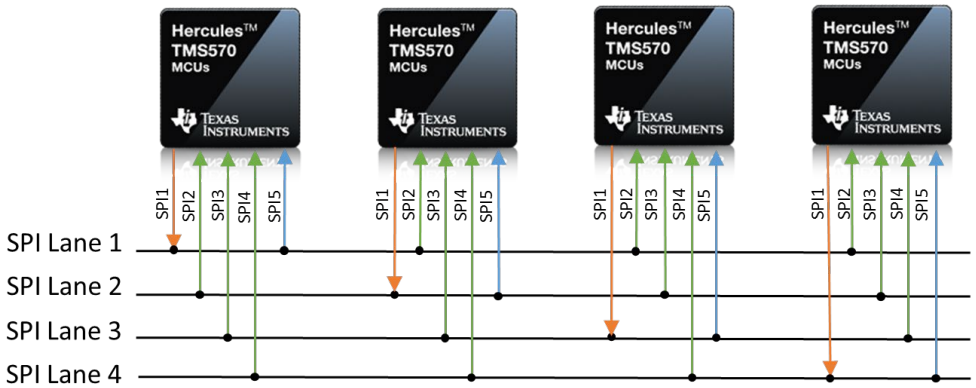
Durchgehender Betrieb auch bei
Ausfall einzelner Recheneinheiten

Maskieren von Fehlerzuständen

→ Durchgehender Betrieb durch Mehrheitsentscheid

Möglichkeiten zur Steigerung der Ausfallsicherheit

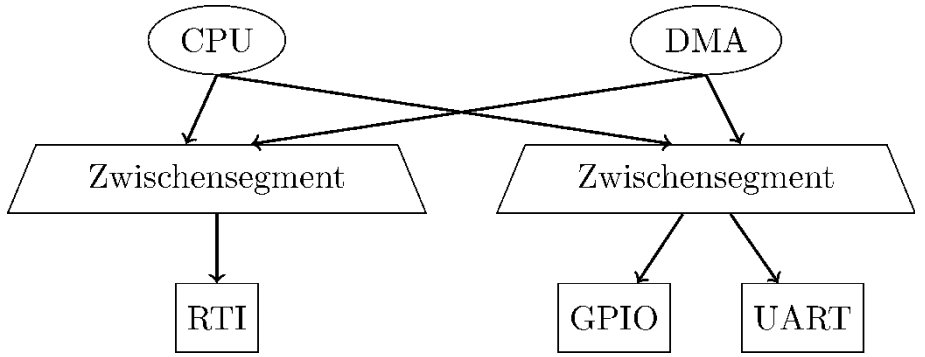
- Redundante Auslegung der Hardware
  - Mindestens Triplex-Systeme (2oo3) zur Fehlermaskierung
  - Keine Umschaltzeiten (im Gegensatz zu 1oo2D)
  - Leistungsanforderungen: Bandbreite, Synchronität
- Redundanznetzwerke werden in FPGA/ASICs realisiert
- **Bestreben**: Implementierung mit Standardschnittstellen
  - Anbindung von FPGA an internes Netz zukünftig problematisch
  - Steigerung des internen Datenaufkommens



Möglichkeiten zur Zusicherung der WCET

- Statische Analysen für CPU und CPU-nahe Komponenten möglich
  - Single-Core CPU
  - Modelle für wenige ausgewählte CPUs verfügbar
  - Cache-Hierarchie
  - Konfliktfreier Zugriff auf Speicher

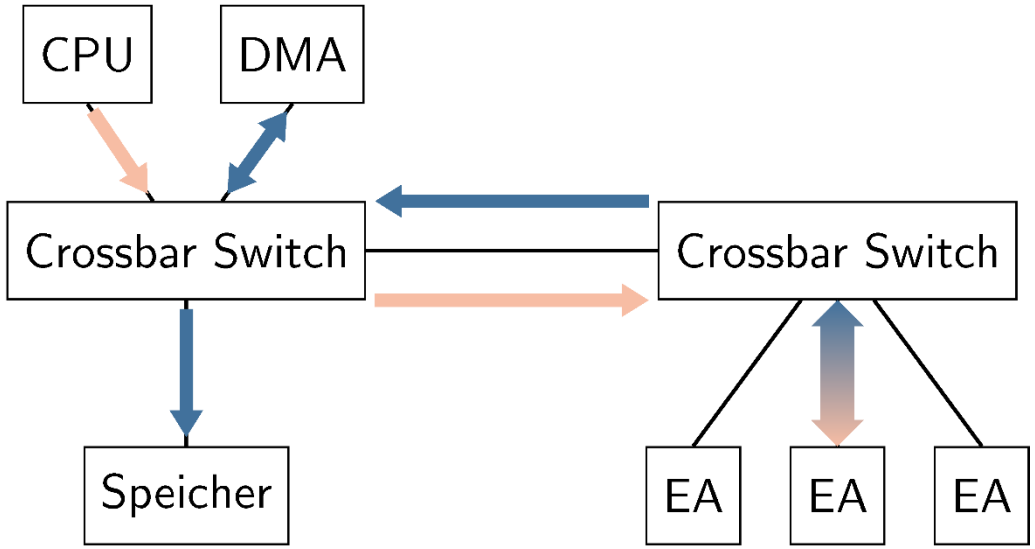
- **Bestreben**: Erweiterung auf IO-Subsystem-Ebene
  - Einsatz von DMA-Controller oder IO-CPU unumgänglich
  - Messbasierte Ansätze nicht immer ausreichend
  - Modellierung der Interferenzen im IO-System



Entwicklungsvorschläge für AS-MCUs in sicherheitskritischer Avionik

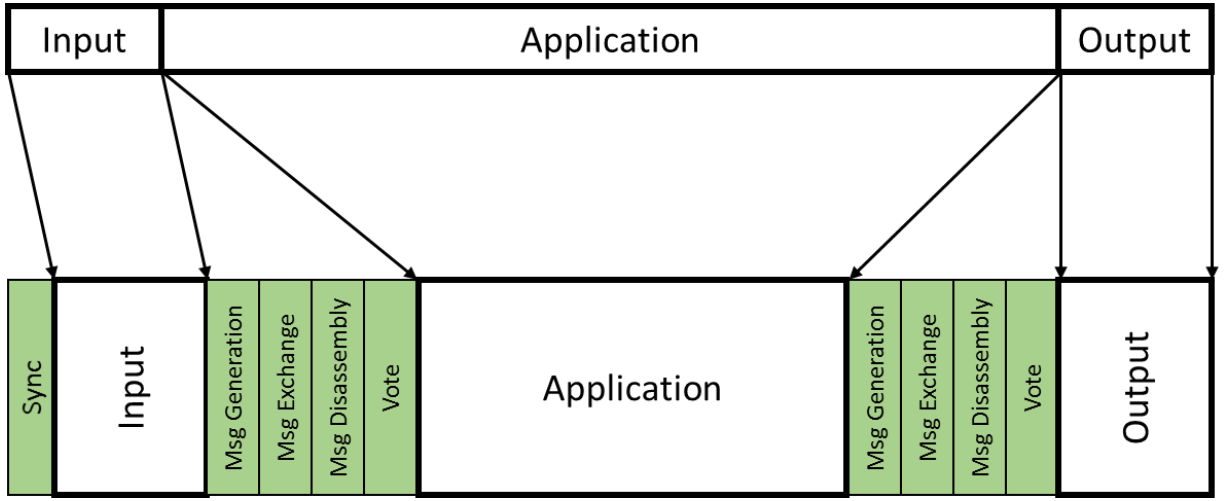
Verbesserung der Echtzeitbedingungen

- Echtzeitfähige CPU-Architekturen
  - Deterministisches Pipelining und Hardware-Threading
  - Deterministische Cache-Verdrängungsstrategien
  - Zeitschranken via Instruction Set Architecture
- Deterministische Verbindungsnetzwerke
  - Dedizierte Anbindung der IO an Crossbar Switch
  - Nutzen von Network-on-Chip Systemen
  - Optimierung auf Worst Case Durchsatz
- Autarke Peripherie
  - Eigenen Speicher
  - Abarbeiten von mehreren Nachrichtenblöcke



Schaffung nativer Grundlagen für Redundanzkonzepte

- Hardwareseitige Implementierung von Redundanzschnittstellen
- Vorverarbeitung der Redundanzdaten (Performance-Problem)
  - Codierung
  - Decodierung
  - Voting-Algorithmus
- Standardisierung der Redundanzschnittstellen



Zusammenfassung

Zusammenfassung

- Automobilindustrie hat bereits Anstrengungen unternommen
  - hohen Grad an Fehlerdiagnose
  - aktuelle Szenarien der Automobil-Branche
- Aktuelle Schwierigkeiten bei der Integration
  - WCET-Analysen sind sehr komplex und eingeschränkt möglich
  - Fail Operational nur mit Zusatzaufwand möglich
- Potenzial für weitere Verbesserungen
  - spezialisierte Schnittstellen
  - Fokussierung auf Determinismus

Zusammenfassung

- Weitere zulassungsrelevante Aufgaben
  - Lebenszyklus von Produkten
  - Errata-Management
  - Einsicht in das Hardware-Design

→ Aspekte auch interessant für autonomes Fahren

Fragen?
Kommentare!