Safety und Security in Multicore-Systemen: So gelingt die Implementierung

Die Anforderungen an sicherheitsrelevante Steuerungen steigen stetig. Multicore-Architekturen bewältigen diese Aufgaben am besten und werden deshalb vermehrt angeboten und eingesetzt.

Die Sicherheitsnormen definiert man in den verschiedenen Industriezweigen je nach der möglichen Gefährdung zum Beispiel durch ein Gerät, eine Maschine oder ein Fahrzeug. Damit eine funktionale Sicherheit des Systems oder der Maschine gewährleistet werden kann, werden bei einem höheren Gefährdungsgrad umso mehr Überwachungsmechanismen eines Systems gefordert. Bekannte Beispiele hierfür sind die SIL- und ASIL- (Automotive SIL) Spezifikationen.

Die neuen **Multicore-Mikrocontroller-Architekturen** bieten mehrere Rechenkerne (CPUs) auf einem Chip und verfügen über gemeinsame globale Ressourcen, die sie sich teilen.

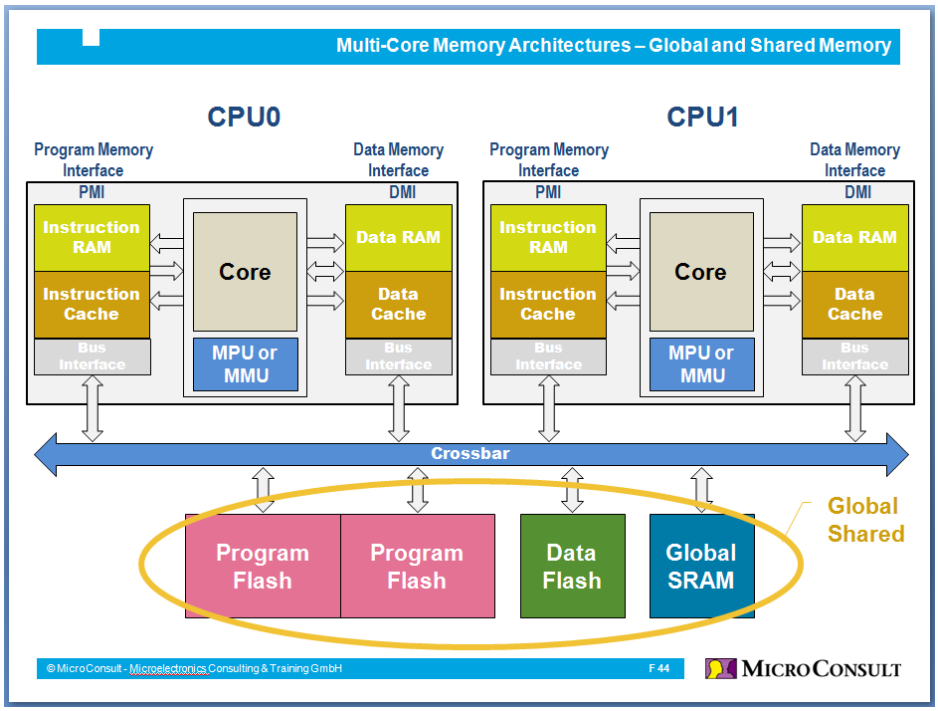

Bild 1: Globale und gemeinsam verwendete Ressourcen in einer Multicore-Mikrocontroller-Architektur

Die Rechenkerne können Aufgaben mit unterschiedlichen Safety-Anforderungen parallel bearbeiten. In solchen Applikationen gilt es, die jeweils benötigten Ressourcen gegenseitig voneinander abzugrenzen und vor unberechtigten Änderungszugriffen zu schützen. Wir müssen also die Integrität der Applikationsdaten, deren Verarbeitung und Übertragung garantieren, um sichere Applikationen zu gewährleisten.

Die Multicore-Mikrocontroller-Bausteine enthalten daher eine Reihe von Überwachungsmechanismen und Funktionsblöcken, die das korrekte Arbeiten von sicherheitsrelevanten Funktionen überwachen. Im Falle eines erkannten Fehlers oder einer fehlerhafter Funktionalität muss das System in einem fest definierten Zeitraum (z.B. innerhalb von 10 ms) eine vordefinierte Fehlerreaktion (Error Response) auslösen.

Die **Überwachungsmechanismen** umfassen:

- Überwachung **fehlerfreier Systemfunktionalitäten** (vital System Parts):
    - **Spannungsüberwachung** für Über- und Unterspannungserkennung

    - **Betriebsfrequenzüberwachung** für Über- oder Unterschreitung
    - **Temperaturüberwachung** des Siliziums
- Überwachung von Daten:
    - Durch **Fehlererkennungs**- und **Fehlerkorrektur-Mechanismen** (**Error Detection** und **Error Correction Mechanismen**) durch den Einsatz von **Error Correction Codes** (ECC) für die Daten in allen SRAMs und Flashes
- Überwachung der **internen Datenkommunikation**:
    - Durch Übertragen von Daten Error Correction Codes ECC und Prüfung bzw. Korrektur im Datenziel
- Überwachung der **Zugriffsberechtigung** auf **Speicher**- und **Peripheriemodule mit Zugriffsschutzeinheiten** zur Überwachung:
    - Schreib- und Lesezugriffe auf Datenspeicher durch **Memory Protection Units** (MPUs)
    - **Schreibzugriffsberechtigung** auf (Peripherie-) **Module** durch **Access Control Units**
- **Überwachung** der **Softwareverarbeitung** (Software-Ablaufsicherung):
    - Befehlsverarbeitung sicherheitsrelevanter Software in der **Safety CPU** (CPU mit Checker / Lockstep Core)
    - **Softwareablaufsicherung** durch Core-spezifische **Watchdog Timer** und **Safety-relevante Software** durch globalen **Safety Watchdog Timer**
    - Differenzierung zwischen sicherheitsrelevanter und nicht sicherheitsrelevanter Software durch **Safety Status Bits**

Zentrale Fehlermeldeeinheit mit **individuell programmierbaren Fehlerreaktionen**

(Fault Collection and Control Unit FCCU bzw. Safety Management Unit SMU):

Alle Fehler werden an diese zentrale Einheit gemeldet. Der Anwender kann für jeden Fehlertyp individuelle Fehlerreaktionen programmieren:

- Aufruf einer Interrupt-Funktion mit wählbarer Interrupt-Priorität
- Aufruf einer Trap-Funktion
- Auslösen eines Resets zum Neustart der Software
- Signalisieren des Fehlerzustandes nach extern über spezifische Signalisierungs-Ports bzw. –Pins

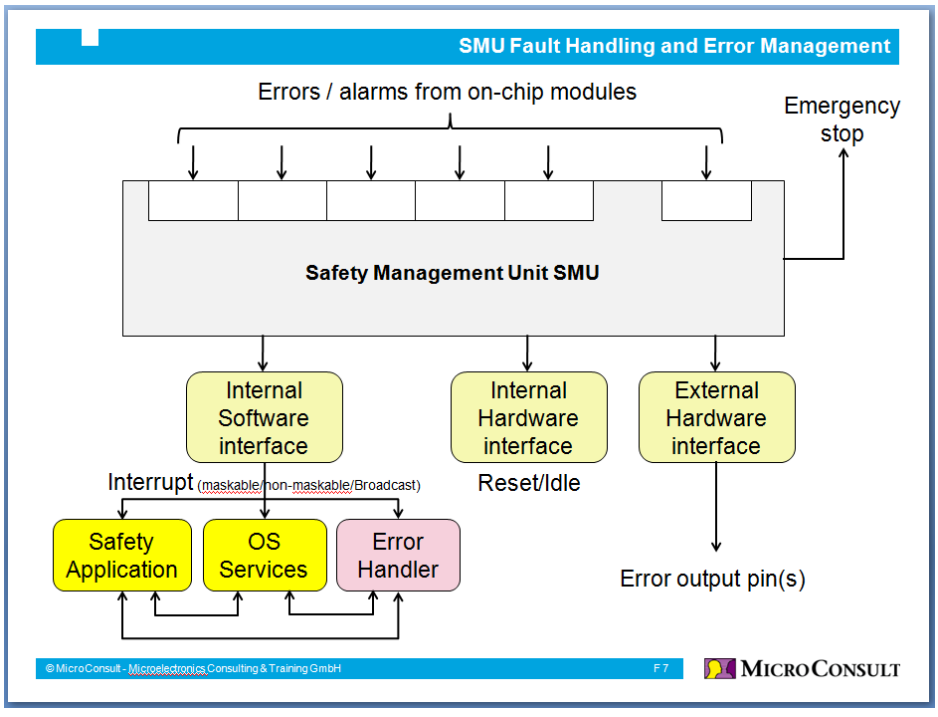

Bild 2: Zentrale Fehlermelde- und Fehlerresponseeinheit (SMU)

Welches Know-how wird für die Fehlerkorrektur-Mechanismen benötigt?

Der Vorteil dieser Safety-Implementierung liegt in der Möglichkeit der softwareunabhängigen Fehlerreaktion: Die Fehlerantwort kann vom Anwender für jeden Fehlertyp individuell eingestellt werden und reagiert auf Fehler der vital Parts des Systems (z.B. Fehler in der Spannungs- und Taktversorgung) hardwaregesteuert, es wird also keine Softwareverarbeitung benötigt. Die Summe aller implementierten Safety-Methoden garantiert, dass das System im Fehlerfall in einen sicheren Zustand überführt werden kann und dient damit zum Schutz vor Gefahren, zur Verhinderung von Schäden und zur Minimierung der Risiken.

Für die Erstellung von Systemen mit Multicore-Mikrocontroller, die den unterschiedlichen Safety-Klassen entsprechende Fehlerüberwachungen und die benötigten Fehlerreaktionen auslösen können, ist das folgende Wissen erforderlich:

- Kenntnisse des **Requirements Engineerings** und **Requirements Managements**
  - Wie definiere und verwalte ich korrekte und sinnvolle Anforderungen an ein Safety-relevantes System?
- Kenntnisse für die richtige **Bausteinauswahl** von **Multicore-Mikrocontroller-Architekturen**
  - Wie wähle ich den richtigen Multicore-Mikrocontroller aus?
- Kenntnisse der **Software-Architektur** für **Hard-Realtime Multicore-Systeme**
  - Wie bilde ich eine Software-Architektur, die die Herausforderungen einer Multicore-Architektur für Hard-Realtime-Applikationen richtig berücksichtigt?
- Kenntnisse des **Ressourcen-Managements**
  - Wie teile ich die vorhandenen Multicore-Ressourcen passend zu den Anforderungen des Systems zu?
- Kenntnisse des **Multicore**-**Debuggings**, **Multicore**-**Tracens** und des **Software-Tests** von **Safety-relevanten Multicore-Systemen**
  - Was muss ich beachten, wenn ich Safety-relevante Multicore-Systeme debuggen will?
  - Welche Herausforderungen stellen Multicore-Systeme beim Debuggen?
  - Wie kann ich sicherstellen, dass mein Safety-relevantes Multicore-System die zeitlichen Anforderungen erfüllt und dies durch Trace-Aufzeichnungen nachweisen?
  - Wie gehe ich bei Trace-Aufzeichnungen mit den übertragungsbandbreitenlimitierten Trace-Interfaces um?
  - Was muss ich beim Test von Multicore-Systemen alles wissen und beachten, damit die Tests auch wirklich umfänglich durchgeführt werden?

Security-Aspekte von Multicore-Systemen

Abhängig von der Art eines Systems werden unterschiedliche Anforderungen an die Sicherheit (Vertraulichkeit, Verfügbarkeit und Datenintegrität) gestellt. Jährlich entstehen Schäden in Millionenhöhe:

- In der Automobilindustrie durch Veränderungen der Steuerungssoftware im Automobil (sogenanntes Software Tuning, z.B. zur Steigerung der Motorleistung)
- In der Welt des IoT (Internet of Things): Wie können wir unsere Konten vor nicht autorisiertem Zugriff über das Internet schützen?

In die neuen Multicore-Mikrocontroller werden eigene **Security-Cores** implementiert. Diese sind innerhalb des Multicore-Mikrocontrollers durch eine **interne Firewall** gegen unerlaubte Zugriffe geschützt. Diese Security-Architekturen enthalten **Ver-** und **Entschlüsselungsmodule**, z.B.:

- **AES-Module** – Advanced Encryption Standard Module
- **Random Number Generator** (RNG)
- **Zugriffsgeschützte Speicher** für den Security-Core (gesichertes RAM zum Verwahren von Sicherheitsschlüsseln, geschützten Programme bzw. Funktionen und Daten).

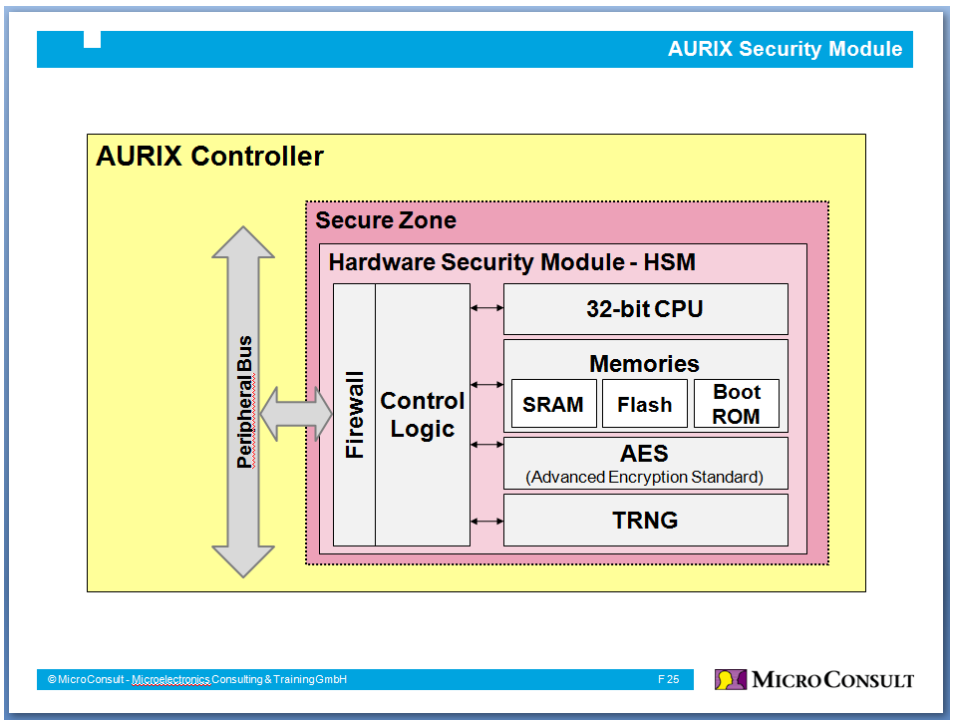

Bild 3: Beispiel für ein Security Module in einem Infineon AURIX Multicore Mikrocontroller

Durch die strikte Trennung dieser Sicherheitsdomäne vom Rest des Systems (Trust Zone for Security Application) lassen sich Angriffe von außen besser abwehren.

Autor:

Dipl.-Ing. (TU) **Marcus Gößler** schloss das Studium der Elektrotechnik an der Technischen Universität Graz ab. Seine berufliche Laufbahn begann als Field Application Engineer für analoge und digitale Produkte im Bereich Luft-und Raumfahrt. Weitere Applikationsfelder umfassten Audio/Video, portable Systeme und Infotainment im Automobil. Er leitete Applikationsorganisationen in Zentral- und Osteuropa und zeichnete Verantwortung für große Halbleiterhersteller im Vertriebskanal und Marketing. Bei MicroConsult ist er heute als Trainer und Coach im Bereich Embedded Systems tätig, mit Schwerpunkten in sicherheitsrelevanten Anwendungen und Multicore-Bausteinen.

Veröffentlicht:

2017

Weiterführende Informationen

MicroConsult Fachwissen zum Thema Multicore & Mikrocontroller

MicroConsult Training & Coaching zum Thema Multicore

MicroConsult Training & Coaching zum Thema Safety & Security