A Flexible Multi-Processor System-on-a-Chip Architecture for Safety- and Security-Critical Applications

Daniel Kliem

Summary

Modularization is a common design pattern in safety- and security-critical embedded software designs. It is mainly motivated by complexity reduction but also allows for effort and cost reduction during development. Domain segregation is a key concept to support such system partitioning. In contrast to strict isolation, segregation allows for communication between segregated components.

This thesis presents a concept of a robust, safe, secure, and efficient architecture with segregation support that is well prepared for certification. Moreover, it deals with aspects of prototype realization on an *Field Programmable Gate Array* (FPGA) platform. The goal is to host different safety and security critical functions with as few hardware components as possible: the *System-on-a-Chip* (SoC) approach.

Software solutions, i. e., operating systems with time and space partitioning, are state-of-the-art approaches to handle segregation. As an alternative to pure software solutions, and to circumvent their particular drawbacks, a novel SoC architecture is proposed.

The architecture offers hardware enforced segregation and is completely transparent to software applications. Since it targets reconfigurable platforms, the architecture is flexible and can be tailored to application specific needs at design time.

This approach follows the recent trend of chip-multiprocessing. Instead of focusing solely on software partitioning, the architecture segregates whole computer systems on a single chip. Segregation is achieved with a hierarchical connection of memory busses by secure bus bridges. Different bridge designs are evaluated. Special attention is paid to performance evaluation and avoidance of temporal conflicts. The architecture is evaluated by dedicated bus observers using simulation and hardware prototypes. It is finally able to run multiple isolated off-the-shelf Linux systems.

A Flexible Multi-Processor System-on-a-Chip Architecture for Safety- and Security-Critical Applications

Daniel Kliem

Zusammenfassung

Für sicherheitskritische Software-Anwendungen im Embedded-Bereich werden häufig modulare Design-Strategien verwendet. Die zuverlässige Trennung (domain segregation) von sicherheitsrelevanten Bereichen ist dabei von hoher Bedeutung. Sie ermöglicht es, komplexe Systeme in einzelne Komponenten zu partitionieren, obwohl diese gemeinsame Schnittstellen besitzen.

Diese Arbeit präsentiert eine robuste, sichere und effiziente Rechnerarchitektur mit eingebauter Bereichstrennung unter besonderer Berücksichtigung von Zertifizierungsbelangen. Die Architektur verfolgt den Ansatz eines *System-on-a-Chip* (SoC). Ein zentrales *Field Programmable Gate Array (FPGA)* beherbergt verschiedene sicherheitskritische Anwendungen und benötigt dazu wenige zusätzliche Bauteile. FPGAs werden als konfigurierbare Plattform verwendet, da so die Architektur in der Entwurfsphase anwendungsspezifisch angepasst werden kann.

Etablierte Verfahren zur Bereichstrennung sind z. B. Software-Lösungen wie partitionierende Echtzeitbetriebssysteme. Durch den Einsatz von konfigurierbarer Hardware besitzt die hier vorgestellte Architektur mehr Flexibilität als Software-Lösungen und bietet dennoch die Sicherheit einer in Hardware realisierten Bereichstrennung. Die vorgeschlagene Lösung arbeitet nicht auf der Ebene der Software-Partitionen. Sie separiert stattdessen komplette, in sich abgeschlossene Rechnersysteme, inklusive Peripherie, auf einem einzelnen Chip. Für die Software-Komponenten ist die Bereichstrennung dabei völlig transparent.

Die Architektur sieht einen hierarchisch angeordneten Verbund von Speicherbussen vor. Diese Busse werden mittels Busbrücken verbunden, welche die räumliche und zeitliche Trennung der Sicherheitsbereiche ermöglichen.

Im Rahmen dieser Arbeit werden verschiedene Busbrückenentwürfe vorgestellt und evaluiert. Dabei wird insbesondere die Skalierbarkeit der Architektur und das Laufzeitverhalten untersucht. Dies geschieht mittels eigens entworfener Bus-Monitore, die die Auslastung der Speicherbusse vermessen. Unter anderem zeigt die Arbeit, dass mehrere unmodifizierte Linux-Systeme parallel und getrennt ausgeführt werden können.