



AUTOMOTIVE · RAILWAY · AVIONICS

MULTICORE SYSTEMS

ARAMiS Schlussbericht

Freescal e Halbleiter GmbH

Version	0.1
Laufzeit des Vorhabens	01.12.2011 - 30.11.2014
Förderkennzeichen BMBF	01IS110352
Verbreitung	Öffentlich
Fälligkeitsdatum	31.05.2015
Datum	17.02.2015

GEFÖRDERT VOM



Bundesministerium
für Bildung
und Forschung

Versionen

Version	Datum	Beschreibung
0.1	17.02.2015	Erste Version , Krutsch Robert

Projektkoordination

Prof. Dr.-Ing. Jürgen Becker / Dr.-Ing. Oliver Sander
Karlsruher Institut für Technologie (KIT)
Institut für Technik der Informationsverarbeitung (ITIV)
Engesserstr. 5
76131 Karlsruhe
Telefon: +49 721 - 608 - 42502 / - 42512
Telefax: +49 721 - 608 - 42511
Email: becker@kit.edu / sander@kit.edu

© Copyright 2015 ARAMiS
Kordinator: Karlsruher Institut für Technologie (KIT)

Autoren

Krutsch Robert, Freescale Halbleiter GmbH

Dr. Schlagenhaft Rolf, Freescale Halbleiter GmbH

Inhaltsverzeichnis

1.1 Aufgabenstellung	1
1.2 Voraussetzungen, unter denen das Vorhaben durchgeführt wurde	2
1.3 Planung und Ablauf des Vorhabens	3
1.4 Wissenschaftlicher und technischer Stand, an den angeknüpft wurde.....	5
1.5 Zusammenarbeit mit anderen Stellen.....	6
2.1 Verwendung der Zuwendung und des erzielten Ergebnisses im Einzelnen, mit Gegenüberstellung der vorgegebenen Ziele	8
2.1.1 TP1 Szenarien und Anforderungen	8
2.1.2 TP2 Systementwurf.....	9
2.1.3 TP3 Hardware.....	10
2.1.4 TP4 Software	14
2.1.5 TP5 Durchgängige Entwicklungsmethodik und Werkzeuge.....	19
2.1.6 TP6 Demonstratoren.....	20
2.2 Wichtigste Positionen des zahlenmäßigen Nachweises ...	21
2.3 Notwendigkeit und Angemessenheit der geleisteten Arbeit	22
2.4 Voraussichtlichen Nutzens, insbesondere der Verwertbarkeit des Ergebnisses im Sinne des fortgeschriebenen Verwertungsplans	22
2.5 Erfolgskontrollberichten beschrieben Während der Durchführung des Vorhabens dem ZE bekannt gewordenen Fortschritts auf dem Gebiet des Vorhabens bei anderen Stellen	23
2.6 Veröffentlichungen des Ergebnisses	23

1. Kurzdarstellung

1.1 Aufgabenstellung

ARAMIS hatte zum Ziel, durch den Einsatz von Multicore-Technologien in den Mobilitätsdomänen Automobil, Avionik und Bahn die technologische Basis zu erweitern und Lösungen zu erarbeiten die zu der Erhöhung von Sicherheit und Verkehrseffizienz beitragen. Multicore-Systeme sind Stand der Technik in vielen Bereichen (e.g. Consumer, Networking, vielen Automobil Applikationen); ARAMIS hat nicht versucht einen Plädoyer für diese Systeme zu geben (die positiven Eigenschaften sind sehr bekannt) sondern diese Systeme zu analysieren in Hinblick zu Applikationen in den drei genannten Domänen und Verbesserung vorzuschlagen so dass kritische Anforderungen von Bereichen wie Echtzeitfähigkeit, Funktionssicherheit, Zuverlässigkeit etc. erfüllt oder besser erfüllt werden können.

Im Einzelnen waren folgende Teilprojekte geplant (Abb. 1):

- **TP0 Gesamtprojektkoordination:** Aufgabe der Projektleitung war die Koordination des Gesamtprojektes.
- **TP1 Szenarien und Anforderungen:** In TP1 sollten die relevante Szenarien für den zukünftigen Einsatz von Multicore-Technologie in den Mobilitätsdomänen Automotive, Avionik und Bahn erarbeitet werden.
- **TP2 Systementwurf:** Ziel von TP2 war eine Systemarchitektur für Multicore-Systeme zu erarbeiten basierend auf dem TP1 definierten Szenarien.
- **TP3 Hardware:** TP3 sollte die Hardware definieren, beschreiben und implementieren (wo möglich) basierend auf den in TP1 ermittelten Anforderungen und der in TP2 spezifizierten Systemarchitektur.
- **TP4 Software:** Im TP4 sollten Software-Architekturen entwickelt werden für den Einsatz von Multicore-Technologie und Virtualisierung im Rahmen der Anforderungen und des Systementwurfs aus TP1 bzw. TP2. Eine enge Zusammenarbeit mit TP3 war geplant.
- **TP5 Durchgängige Entwicklungsmethodik und Werkzeuge:** TP5 hat sich der Betrachtung von Werkzeugen und dem Entwurf durchgängiger Methoden die dem Design von Multicore-Systemen unterstützen gewidmet.
- **TP6 Demonstratoren:** TP6 hatte die Validierung der Architekturvorlagen für System, Hard- und Software aus TP2, TP3 und TP4 und der entsprechenden Methoden und Tools aus TP5 zum Ziel gesetzt. Demonstratoren sollten die wesentlichen Erkenntnisse implementieren und validieren.

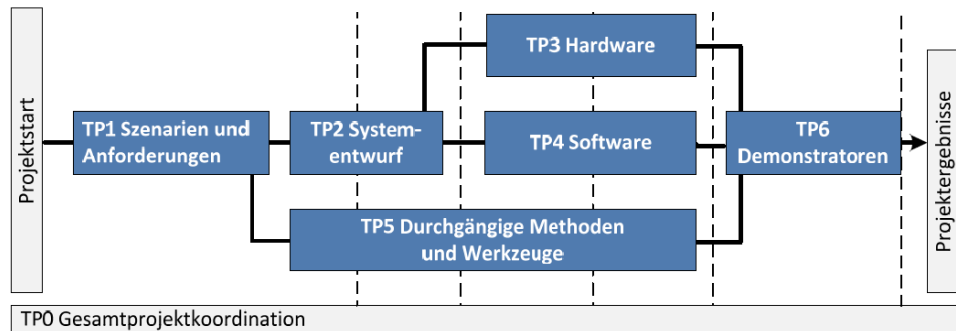


Abb. 1 – Teilprojekte ARAMIS

1.2 Voraussetzungen, unter denen das Vorhaben durchgeführt wurde

Das Konsortium bestand aus einer erheblichen Anzahl Partner, die viel Erfahrung mitgebracht haben in den jeweiligen Domänen. Durch diese Zusammensetzung war eine optimale Abdeckung und Aufteilung der Projektinhalte und -ziele möglich.

Für die einzelnen Arbeitspakete konnte auf folgende Vorarbeiten aufgebaut werden:

TP1 - Beim Konsortialpartner wie Daimler, BMW, Audi, Cassidian, Airbus, Liebherr etc. gab es Konzepte für zukünftige Systeme (Autos, Flugzeuge) und gesammelte Erfahrungen mit Werkzeugen (e.g. Enterprise Architect, die eine Modellierung der Anforderung und Szenarien erlauben). Freescale hat in der Automobil Domäne verschiedene Multicore Plattformen im Einsatz und verschiedene Systeme in der Avionik Domäne. Intel hat sehr viele Erfahrungen gesammelt im Bereich Virtualisierung von General Purpose Hardware. Software Firmen wie Sysgo, OpenSynergy etc. haben und hatten gute Kenntnisse und Vorstellungen von zukünftigen Applikationen und Erweiterungen der existierenden Betriebssysteme.

TP2 – In Rahmen von TP2 wurde auf dem Modellierungseinsatz zurückgegriffen. Partner, die Erfahrung mit Anforderung Modellierung in anderen Projekten hatten, haben mit Beispielen und Erklärungen sehr viel weitergeholfen. Existierende System Architekturen im Bereich Avionik und Automobil wurden als Startpunkt benutzt. Wichtige Informationen (gesammelt während der Realisierung existierender Plattformen) sind von Partnern wie Daimler, Bosch, Continental, Cassidian, Liebherr etc. die konkreten Anforderungen gebracht haben, runtergebrochen auf Subsystemen Ebene.

TP 3 – In Rahmen von TP3, die drei Halbleiter Firmen (Intel, Infineon, Freescale) haben existierende Hardware Lösungen

vorgestellt und zu verfügunggestellt. Existierende Projekte in der Industrie (e.g. PCI SIG wo I/O Virtualisierung angegangen wird) oder Lösungen in der General Purpose Domäne (e.g. DirectX und Windows Graphics Processing Units preemption) sind als Startpunkte berücksichtigt worden. Hardware von anderen Firmen die nicht teil des Projektes waren sind mitberücksichtigt worden (e.g. ARM Virtualisierung Extension).

TP 4 - Existierende Betriebssysteme und Hypervisor Software (von Sysgo, OpenSynergy, GreenHills oder open source Lösungen wie Xen und Linux) sind als Startpunkt benutzt worden. Industrie Standards und Kommerzielle Lösungen (e.g. OpenGL, EGL von Khronos, DirectX von Microsoft) worden als Stand der Technik in Betracht gezogen.

TP 5 – Freescale hat sich mit der Thematik, in Rahmen von ARAMIS, nicht befasst.

TP 6 – Die Halbleiter Unternehmen haben existierende Hardware Systeme den Konsortialpartnern zu verfügunggestellt. Existierende Betriebssystem, Hypervisors und Treiber sind als Startpunkt der Demonstratoren Benutzt worden.

1.3 Planung und Ablauf des Vorhabens

Die Kompetenz für die Thematiken betrachtet in ARAMIS ist in Freescale präsent aber nicht alle Thematiken kontern abgedeckt werden mit den Mitarbeitern die in Standort München arbeiten. Um Lücken abzudecken haben wir Mitarbeitern aus anderen Standorten nach Deutschland transferiert.

Auf dem Projekt haben folgende Ingenieure gearbeitet:

Name	Kernkompetenz
Dr. Rolf Schlagenhaft	Safety
Markus Baumaister	System Modellierung
Dr. Reiner Markowitz	Projekt Management
Robert Krutsch	Graphik und Virtualisierung
Manfred Thanner	System Modellierung
Juergen Frank	Security

Tabelle 1 – Freescale Mitarbeitern in ARAMIS

Um einen besseren Überblick auf dem Gesamtprojekt zu bekommen haben wir die TPs nicht zu einzelnen Ingenieuren

aufgeteilt, sondern Thematik orientiert die Aufgaben aufgeteilt (e.g. Virtualisierung in Software in Hardware sind von derselben Person verfolgt worden).

Durch internen Umstrukturierungen konnten wir nicht alle geplanten Aktivitäten machen oder zu Ende bringen (e.g. wir mussten die Leitung von TP 2.1.4 abgeben und die Zeit für Beiträge in Ergebnisdokumente zu benutzen). Unsere Priorität war die Konsortiumspartner zu unterstützen und haben uns sehr viel darauf konzentriert Schulungen und Workshops für unsere, in Projekt benutzten, Hardware Plattformen zu organisieren. Nach der Umstrukturierung haben wir verstärkt Virtualisierung und insbesondere Graphik Karten Virtualisierung untersucht. Die Konzepte und Untersuchungen in Bereich Graphik und Speicherschutz Virtualisierung sind sehr wichtige Anforderungen geworden für die kommenden Architekturen.

Der online Portal und Configuration Management Werkzeuge (SVN) die Anfang des Projektes aufgesetzt worden sind haben sich als zehr hilfreich dargestellt. Wir konnten schnell Meetings aufsetzen, Dokumente zu Prüfung geben und Teilprojektmitarbeiter ausfindig machen.

Eine sehr gute Kooperation hatten wir mit den Graphik Karten Architektur Team von Vivante, die uns viele Details über die internen Elemente der GPU gegeben haben. Wir hatten verschiedene Workshops wo wir detailliert über die Virtualisierung de Vivante GPUs gesprochen haben und wo wir viele Fragen über der Laufzeitabschätzung stellen konnten.

Die meisten Erkenntnisse, in ARAMIS, haben wir in Bereich GPU Virtualisierung bekommen. Die Zusammenarbeit mit Daimler, Sysgo und OpenSynergy für die Demonstratoren und in Rahmen von TP4 ist sehr wichtig für uns gewesen.

Ein Problem, mit dem dieses Projekt fertig werden musste ist die Breite der Thematiken die adressiert worden sind. Wir haben sehr schnell begreifen, nach TP1, das die Anforderungen (in den verschiedenen Domänen und auch in verschiedenen Applikationen in derselben Domäne) sehr schwer unter eine unifizierte Betrachtung bringbar sind. Wirtschaftlich und wegen Gründen der Energieeffizienz sind die optimalen Ansetze sehr unterschiedlich (e.g. Infotainment Applikationen die auf sehr neuwertigen Architekturen aufbauen können Virtualisierungsansetze benutzen aber Steuergeräte die kleine aber Energieeffiziente Microcontroller benutzen können nur Isolierung Konzepte implementieren).

1.4 Wissenschaftlicher und technischer Stand, an den angeknüpft wurde

Freescall, als Nummer zwei (nach Renesas Electronics) der Hardware Halbleiter Firmen in Bereich Automotive hat eine langjährige Erfahrung in der Entwicklung von Plattformen die in Safety relevanten Applikationen benutzt sind. Freescall entwickelt auch Multicore Netzwerk Hardware Systeme die komplette Virtualisierung in Hardware unterstützen. In ARAMIS, haben Ingenieure aus den beiden Domänen zusammengearbeitet um auf existierende Kenntnisse aufzubauen und um schnell Lösungen ausfindig zu machen.

Freescall bietet zwei Hypervisor Lösungen in Bereich Networking (Type 1 und Type 2) die als Referenz benutzt worden sind. Die Software Lösungen von Sysgo und OpenSynergy waren ein guter Startpunkt und worden als Bases für die Demonstratoren eingesetzt. Open source Lösungen (zum Beispiel Xen) worden als Stand der Technik in Betracht gezogen.

Um die Erarbeiteten Ansätze zu testen und auch um die Demonstratoren zu implementieren haben wir existierende Mikroprozessoren benutzt:

- i.mx 6 , für Graphik Applikationen , GPU Virtualisierung und Demonstratoren
- P4080, Plattform die komplett Virtualisierung in Hardware unterstützt
- MPC5746M, Safety Mikrocontroller

Für Security Thematiken haben wir unsere Akzeleratoren vorgestellt. Bereits in existierende Hardware sind dies Co - Prozessoren mit Virtualisierungs Ansätze ausgestattet.

Für die Safety Thematik haben wir als Startpunkt die folgenden ISO Standards benutzt:

- **ISO 11428** - Ergonomic requirements for the perception of visual danger signals.
- **ISO 26262** - Risk-based assessment of potentially hazardous operational situations and of safety measures.
- **ISO 15005** - Requirements to prevent impairment of the safe and effective operation of the moving vehicle.
- **ISO 16951** - Priority-based presentation of messages.
- **ISO 2575** - Symbols for controls and indicators.
- **ISO 15408-2** - Security in IT systems.

Als Stand der Technik worden auch Lösungen von Firmen in Betracht gezogen die nicht teil des Konsortium waren (e.g. Virtualisierungs Lösungen von ARM und AMD).

1.5 Zusammenarbeit mit anderen Stellen

Wir haben innerhalb des Projektes kleinere Arbeitsgruppen gebildet mit Firmen die gleiche Interessen gezeigt haben. Eine sehr gute Zusammenarbeit hatten wir mit Daimler, Sysgo und Opensynergy für die Demonstrator Platforms (Abb. 2).



Abb.2 – GPU Virtualisierung Demonstrator

Wir haben in Bereich System Modellierung und Anforderung Modellierung eine enge Zusammenarbeit mit der Firma Liebherr die viel Vorwissen mitgebracht hat von anderen Forschungsprojekte (Abb. 3).

Eine gute Zusammenarbeit hatten wir auch mit Vivante (nicht ARAMIS Partner), die Graphik Karten Zulieferer für i.mx 6. In verschiedenen Workshops haben wir die GPU Treiber Architekturen besprochen, die Virtualisierung Schnittstelle und Laufzeitabschätzungen diskutiert.

2. Eingehende Darstellung

In Projekt ARAMIS wurde insbesondere Echtzeitfähigkeit, Zuverlässigkeit, Safety und Security von MultiCore Systeme Studiert. Der meistbenutzte Ansatz in ARAMIS, der sich auch in anderen Industriedomänen bewährt hat, ist Virtualisierung gewesen. Es sind viele Erkenntnisse gewonnen worden in Bereich I/O Virtualisierung, GPU Virtualisierung und Echtzeitfähigkeit auf System Ebene. Es wurde schnell klar das die echten Herausforderungen in Bereich Virtualisierung nicht auf der Ebene der Rechenkerne auftreten (diese Probleme sind in anderen Domänen gelöst worden, e.g. Networking) sondern bei Hardwareblocken die von mehreren Anwendungen gemeinsam genutzt werden müssen (e.g. SoC Interconnect, Akzeleratoren, I/O Schnittstellen). Die Studien die in ARAMIS gemacht worden sind haben gezeigt das Isolierung (zeitlich und Speiche bezogen) von Applikationen ist sehr Schwierig (wenn nicht unmöglich) wen keine Hardware Unterstützung vorliegend ist (e.g. Bandbreiten Garantie in einen System das keinen QoS unterstütz). Wichtig war auch zu analysieren ob die Konsolidierung der Applikationen Wirtschaftlich effizient ist (e.g. Infotainments Systeme haben geringe Safety Anforderungen in Vergleich zu Driver Assist Systems; blinde Implementierung aller Safety Maßnahmen wurde zu Preislich unattraktiven Endprodukten führen).

2.1 Verwendung der Zuwendung und des erzielten Ergebnisses im Einzelnen, mit Gegenüberstellung der vorgegebenen Ziele

2.1.1 TP1 Szenarien und Anforderungen

In dieses Teilprojekt worden verschiedene Szenarien und Anforderungen auf höre ebene von OEM Konsortiumpartner präsentiert und in Enterprise Architect Modellen eingebracht (e.g Daimler hat das Konzept „in Auto wohnen“ vorgeschneilt).

2.1.1.1 Aufgabenstellung in TP1

Als Halbleiter Unternehmen hat Freescale sich nur Qualitätprüfung der modele und Dokumente vorgenommen. Die Ergebnisse von TP1 worden als wichtig angesehen für TP2 und für TP6 (wo

Szenarien mit Demonstratoren gezeigt werden sollten). Für Freescale ist es wichtig gewesen zu verstehen welche Hardware System benötigt werden um die notwendigen Schritte zu machen für die Anschaffung der Boards.

2.1.1.2 Ablauf und Ergebnisse von TP1

Freescale hat keine direkten Beiträge zu TP1 Ergebnisdokumente und Modelle geliefert. Wir haben die Modelle und Dokumente qualitätsgeprüft damit ein unifizierter Ansatz zwischen TP1 und TP2 sichergestellt werden kann. In Rahmen von TP6 Meetings sind wir mit den Partnern die Szenarien und Anforderung durchgegangen und haben identifiziert welche Hardware System benötigt werden um die Demonstratoren zu realisieren. Wir haben die Anschaffung der Boards in die Wege geleitet so dass genügend Zeit für die Entwicklung bleibt.

Der Model basierte Ansatz (mit Hilfe von Enterprise Architect) ist auch in TP2 widerbenutzt worden.

2.1.2 TP2 Systementwurf

In TP2 worden mehr detaillierte Anforderungen spezifiziert, Logische Architektur modele und technische Architektur modele worden realisiert und analysiert.

2.1.2.1 Aufgabenstellung in TP2

Wir haben uns in Rahmen von TP2 die folgenden Ziele gesetzt:

- Technische Leitung von TP2.1.4
- Qualitätprüfung der Security und Safety Teilprojektergebnisdokumente
- Präsentation der existierenden Freescale Multicore Systeme die relevant für die Safety und Security Applikationen die in ARAMIS betrachtet werden
- Qualitätprüfung von Ergebnisdokumenten die die Logische Rechnerarchitektur beschreiben
- Beiträge zu Ergebnisdokumenten die die Technische Rechnerarchitektur beschreiben

2.1.2.2 Ablauf und Ergebnisse von TP2

Wir haben die technische Koordinierung von TP 2.1.4 bis zu der internen Umstrukturierung, die uns leider in der Position gebracht hat das keine Ressourcen für die Aktivität verfügbar waren. Wir haben vorgezogen die technischen Beiträge zu

dem Ergebnisdokument abzuliefern und die Leitung abzugeben. Während der Zeit wo wir die Leitung hatten haben wir Meetings mit den Partnern in zwei Wochen Takt organisiert, Dokument Muster vorbereitet und Workshops organisiert wo der Stand der Arbeiten in Detail besprochen worden ist. In den Zeitpunkt wo wir die Leitung abgegeben haben waren die ersten Versionen der Ergebnisdokumente verfügbar und erste Versionen der Enterprise Architect Modellen realisiert.

Basierend auf SPES 2020 Enterprise Architect Modellen haben wir ein Model erstellt für Freescales McKinley Prozessor. Die SPES 2020 Meta-Modelle wurden erweitert mit spezifischen Charakteristiken die eine bessere Darstellung von Hardware Architekturen ermöglichen. Unsere McKinley Modelle wurden von den Partnern als Vorlagen benutzt und in Ergebnisdokument 2.1.4.1 untergebracht. Wir haben auch zu mehreren Kapitel des Ergebnisdokumentes E2.1.4.1 beigetragen.

Freescale hat einem Security Workshop organisiert wo Zusammen mit Infineon existierende Lösungen für security präsentiert worden sind. In einer Brainstorming Session worden Anforderungen aufgesammelt die Später in TP 2.2 eingeflossen sind.

Wir haben mit Qualitätprüfung beigetragen bei mehreren TP 2.1.4 Dokumenten und TP 2.1.3.

2.1.3 TP3 Hardware

In TP3 worden verschiedene Virtualisierung Ansätze studiert und implementiert. Lösungen die in anderen Domänen zu Stand der Technik gehören sind adaptiert worden für den Automotive Bereich (e.g. Single Root Virtualisierungs Konzepte die bei PCIe benutzt sind worden für CAN Schnittstellen adaptiert). Die Problematiken der Coprozessoren Virtualisierung, IO und Speicherschutz Virtualisierung sind angegangen worden. Safety und Security Aspekte worden in Rahmen von TP3.3 und TP 3.2 studiert und Hardware Lösungen worden vorgeschlagen.

2.1.3.1 Aufgabenstellung in TP3

Wir haben uns in Rahmen von TP3 die folgenden Ziele gesetzt:

- Qualitätprüfung der Ergebnisdokumenten in TP 3.1 (Heterogene Hardware Architekturen)
- Qualitätprüfung und Beiträge zu den Ergebnisdokumenten in TP 3.2 und TP 3.3 wo Safety und Security Lösungen die in Freescale Produkten implementiert sind vorgestellt werden

- Technische Leitung von TP3.4.1 und Beiträge zu afferenten Ergebnisdokumenten
- Technische Leitung von TP3.3.1
- Technische Leitung von TP 3.4.5 und Beiträge zu E3.4.51

2.1.3.2 Ablauf und Ergebnisse von TP3

In Rahmen von TP3 haben wir das erste Ergebnisdokument erstellt, E3.4.1. Wir haben Regelmäßige Konferenzen organisiert die Beiträge von unseren Partnern aufgesammelt und das Ergebnisdokument editiert und qualitätgeprüft. Unsere Beiträge zu TP3.4.1 sind im Bereich GPU Virtualisierung, ARM Virtualisierungs Extension, Power PC Virtualisierung und IOMMU/System MMU. Die Evaluierung der verschiedenen Architekturen konnte nur qualitative gemacht werden, zusammen mit unseren Partnern haben wir Benchmark Matrix definiert und ausgeführt (Abb 5).

State of the Art Virtualization Evaluation					
Criteria:		Use case:			
		Single core:		Multi core:	
		Single OS	Multi OS	Single OS per core	Multi OS per core
Architecture:	QorIQ, Pjazz	Legend:	fully supported, <comments>, <cross reference to text>		
Author:	Rolf Schlagenhaft, Robert Kru		partially supported, <comments>, <cross reference to text>		
Date:	6/26/2013		not supported, <comments>, <cross reference to text>		
Virtualization	Emulation	n.a.	not reason	not reasonable	not reasonable
	Para-virt.	n.a.	possible	possible	possible
	Full virt.	n.a.	possible	possible	possible
Processing	Goldberg instr. set	yes	yes	yes	yes
	Privilege levels	2x2	2x2	2x2	2x2
	Debug support	supported	supported	supported	supported
	Interrupt routing	supported	supported	supported	supported
Special Process	DMA	supported	supported	supported	supported
	GPU	has no GF	has no GF	has no GPU	has no GPU
	Security module	supported	supported	supported	supported
RAM	Static allocation	supported	supported	supported	supported
	Dynamic allocation	not reason	not reason	not reasonable	not reasonable
	Local RAM	no	no	no	no
	Global RAM	supported	supported	supported	supported
	Tightly coupled RA	no	no	no	no
	Cache coherency	supported	supported	supported	supported
	MMU regions	yes	yes	yes	yes
	MMU granularity	yes	yes	yes	yes
	MPU regions	no	no	no	no
	MPU granularity	no	no	no	no
31 Bus arbitration	Determinism	supported	supported	supported	supported
	Fairness	supported	supported	supported	supported
	Bandwidth	supported	supported	supported	supported
Periphery	Simple GPIO	supported	supported	supported	supported
	Packet oriented IO	supported	supported	supported	supported
	IOMMU	partially	partially	partially	partially
	Emulation	supported	supported	supported	supported
	Sharing	supported	supported	supported	supported
	Fixed assignment	supported	supported	supported	supported
Ecosystem	Compiler, linker	supported	supported	supported	supported
	Debugger, IDE	supported	supported	supported	supported
	Recompil. for virt.	not needs	not needs	not needed	not needed
	Libraries	supported	supported	supported	supported
Miscellaneous	...				

Abb. 5 – Benchmark Matrix definiert in TP 3.4.1

Nach der internen Umstrukturierung haben wir nicht mehr die notwendigen Ressourcen zu Verfügung gehabt um die Leitung von TP 3.3.1 zu behalten. Wir haben den Kick off Meeting organisiert,

die entsprechenden Partnern informiert über den wechsell der Leitung. In Rahmen von TP 3.3 haben wir MPC5746M Boards für Lieberr besorgt und ein Workshop organisiert für die Inbetriebnahme der Hardware. Verschiedene Safety relevante Hardware Elemente auf den Board sind diskutiert worden um die Partner in deren Investigation zu helfen.

In Rahmen von TP 3.4.4 Haben wir Beiträge abgeliefert für GPU und Security Coprozessor Anbindung. Unsere Beiträge basieren auf der gesammelten Erfahrung mit i.mx 6 (Abb. 6). Wir haben auch eine Variante der GPU Virtualisierung präsentiert, basierend auf Multiple Contexts, die wir in für die kommenden i.mx SoC in Betracht ziehen. Nach der Zusammenlegung vor TP3.4.4 und 3.1.4 haben wir mit Qualitätprüfung Aktivitäten geholfen.

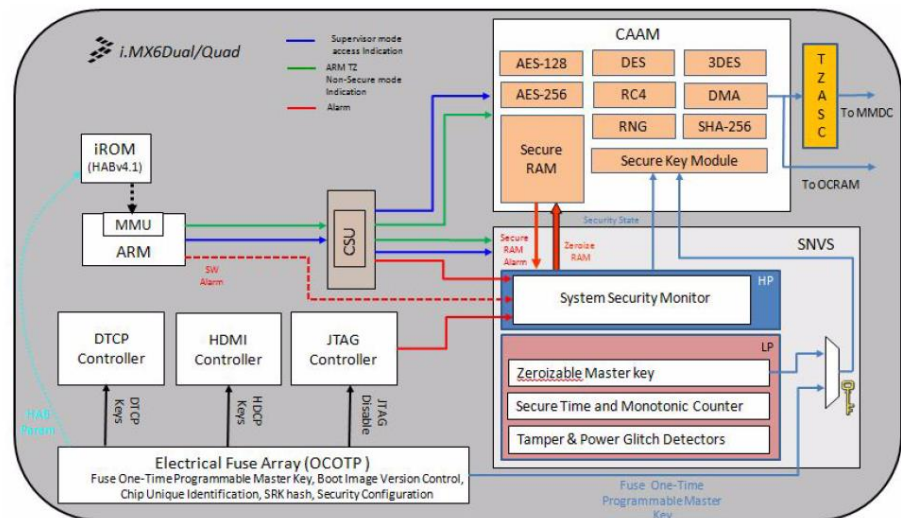


Abb. 6 – Security Subsystem in i.mx 6, Security Elemente sind verteilt in der SoC

In Rahmen von TP3.2 haben wir Beschreibungen des CAAM Cryptographic Akzelerators als Beitrag zu Dokumenten abgegeben (Abb. 7). Die Architektur dieses Akzelerators ist mit Hinblick zu Virtualisierung realisiert worden wegen seine Benutzung in Networking Bereich (eine simplifizierte Variante ist auch in Automotive SoCs).

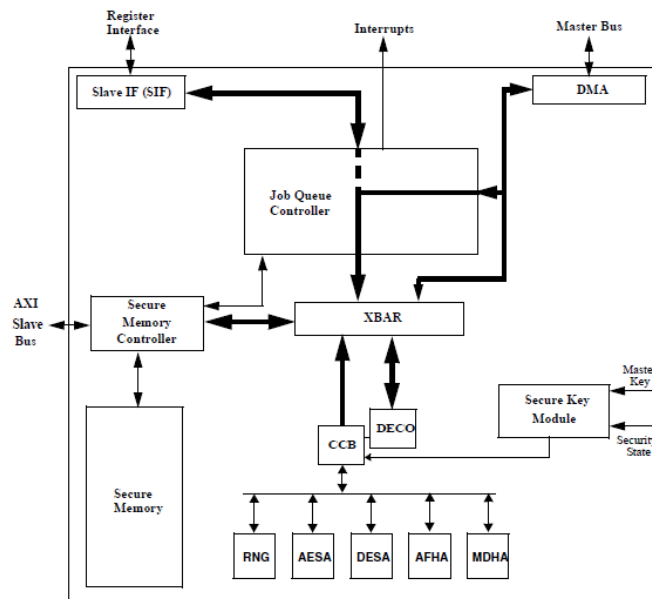


Abb. 7 - Cryptographic Acceleration and Assurance Module

In Rahmen von TP3.4.5 haben wir Regelmäßige Konferenzen organisiert die Beiträge von unseren Partnern aufgesammelt und das Ergebnisdokument editiert und qualitätgeprüft. Die drei Ergebnisdokumente worden in einen Dokument unifiziert wegen der geringen Inhalte in E3.4.5.2 und E3.4.5.3. Die Beiträge die wir geliefert haben basieren auf den Erfahrungen die wir mit verschiedenen SoC gesammelt haben (mit ARM, PowerPC Rechenkernen und verschiedenen Coprozessoren) und beschreiben Lösungen basierend auf Memory Protection Units, Memory Management Units, I/O Memory Management Units, etc. auf mehreren Ebenen. Wir haben eine Methode für Adaptive QoS präsentiert, für SoCs die Bandbreiten Profiling und Priority QoS unterstützen aber nicht Bandbreiten für jeweilige Bus Master allokkieren können (Abb. 8).

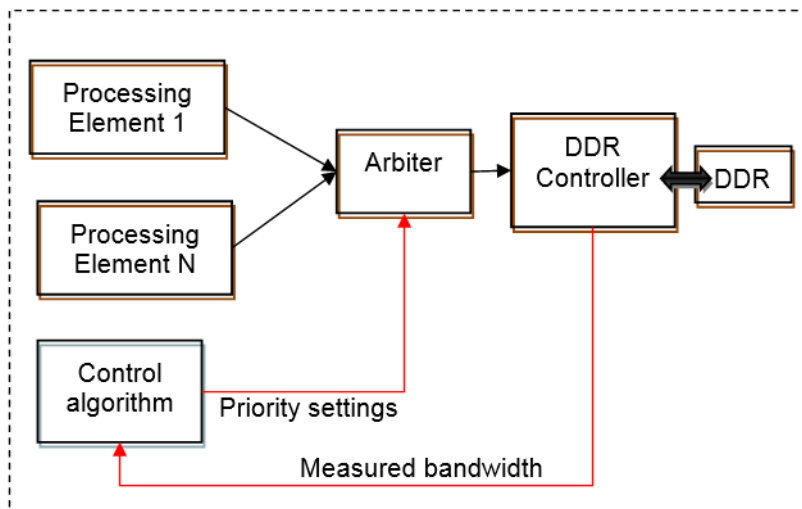


Abb. 8 – Regelschleife für Bandbreiten allokiere

In Rahmen von TP 3.4.3 haben wir ein Konzept präsentiert für IO Schnittstellen Virtualisierung das für zukünftigen i.mx SoC berücksichtigt wird (wir haben typische „low speed“ Schnittstellen berücksichtigt). Unsere Architektur kann mehrere IO Schnittstellen gleichzeitig virtualisieren und benötigt kein neuer Controller für die Schnittstellen (Abb. 9).

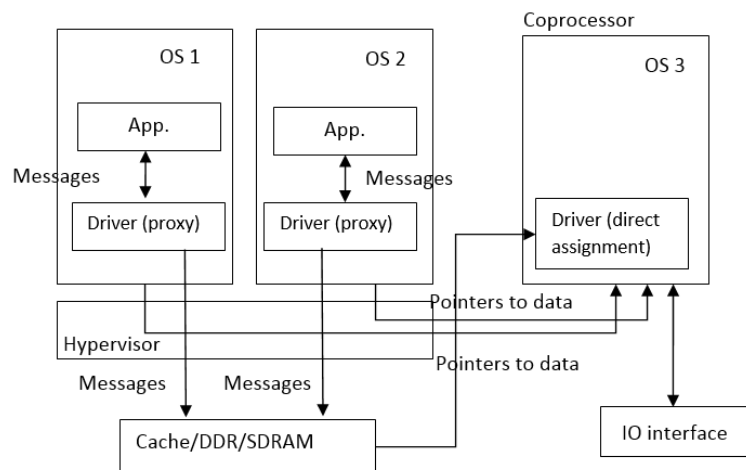


Abb. 9 – IO Coprozessor

2.1.4 TP4 Software

Im TP4 sind Software Architekturen und Implementierungen entstanden die Virtualisierungs Lösungen unterstützen. In Rahmen

von TP4 haben viele Partner GPU Virtualisierung Konzepte studiert. Ein beliebtes Forschungsthema ist WCET (Worst Case Execution Time) gewesen; verschiedene Ablaufplanungskonzepte (im English „scheduling“) und verfahren worden entwickelt und implementiert.

Die Partner die Hardware Blöcke in TP3 implementiert haben sind in TP4 mit Treiber Entwicklung und Software Stack Implementierung beschäftigt gewesen.

2.1.4.1 Aufgabenstellung in TP4

Wir haben uns in Rahmen von TP4 die folgenden Ziele gesetzt:

- Die Laufzeit von GPU Applikationen untersuchen und die Elemente die diese Laufzeit signifikant beeinflussen ausfindig machen
- GPU Virtualisierung Lösungen untersuchen
- GPU scheduling Konzepte untersuchen

2.1.4.2 Ablauf und Ergebnisse von TP4

In Rahmen von TP 4.1 haben wir GPU Laufzeitanalysen durchgeführt. Dieser Aspekt ist ein sehr wichtiger Teil des Gesamtkonzeptes der Vorhersagbarkeit. In der Embedded Welt, und auch in der general purpose compute Branche GPUs sind nicht preemptive (oder haben spezifische punkte in der pipeline wo preemption möglich ist). Es ist möglich die GPU zu unterbrechen mit einen Reset aber ein task-switch wie auf den Rechenkernen ist nicht möglich. Um die GPUs komplett preemptive zu machen müssten Zwischenergebnisse gespeichert werden; das kann zu nicht akzeptablen Lösungen führen (e.g. nur ein GPU Rechenkern hat 512 128bit Register auf GC2000 GPU) wegen Bandbreiten, Latenz und extra SoC Fläche. Um schneller die wichtigen Hardware Elemente die Laufzeitbeeinflussend sind ausfindig zu machen habe wir verschiedene Applikationen (Surround View, kleine OpenCL und OpenGL Applikationen) getestet und mit den GPU Profiler analysiert (Abb. 10).

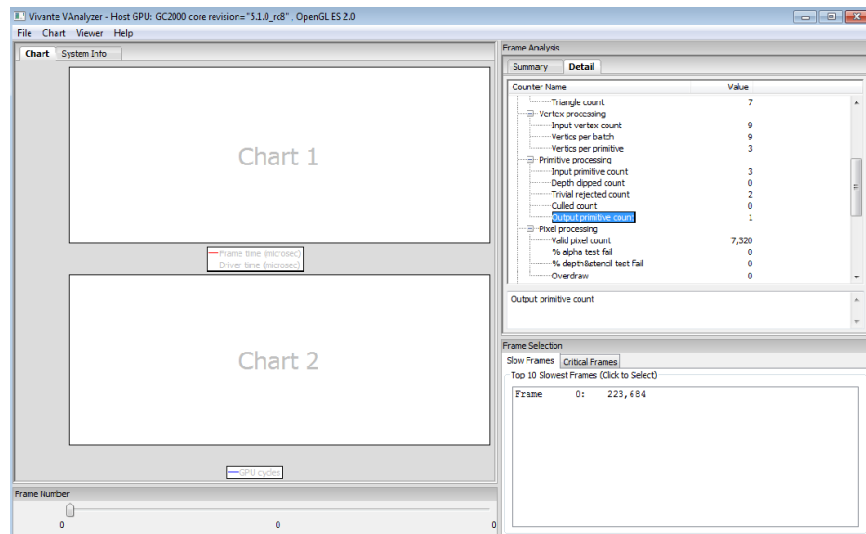


Abb. 10 – Vivante GPU profiler

Basierend auf der Analyse haben wir die folgenden Faktoren als sehr wichtig für die Vorhersehbarkeit der Laufzeit von GPU Applikationen:

- Eingangs Daten zum Vertex Shader
- Eingangs Daten zum Fragment Shader
- Laufzeit benötigt in Vertex Shader
- Laufzeit benötigt von Fragment Shader

Um die Laufzeit der shader zu untersuchen haben wir den GPU Compiler mit Assembly Code Generierung ausgestattet. In der GPU Domäne sind selbstverständliche Dinge wie Documentation des Instruction Sets nicht gegeben. Die GPU Anbieter versuchen die interne Architektur so viel wie möglich zu verbergen und zu Abstrahieren. Basierend auf den Assembly Code konnten wir die Laufzeit der jeweiligen Instruktionen testen und haben den Einfluss von verschiedenen load/store Instruktionen messen können. Es ist offensichtlich geworden das die load/store Instruktionen eine Erhebliche Einfluss haben für Vivante Architekturen und das eine Laufzeitabschätzung diese Effekte unbedingt berücksichtigen muss.

Um den Bandbreiten Einfluss zu bestimmen haben wir offline Messungen für verschiedene load/store Instruktion Typen vorgeschlagen. Um mit der Dynamik der Applikation schrittzuhalten für Architekturen die DDR Bandbreiten nicht allokkieren können eine Regelschleife mit Messung auf der DDR Controller ebene ist im E4.1.2 vorgeschneilt worden (Abb. 11).

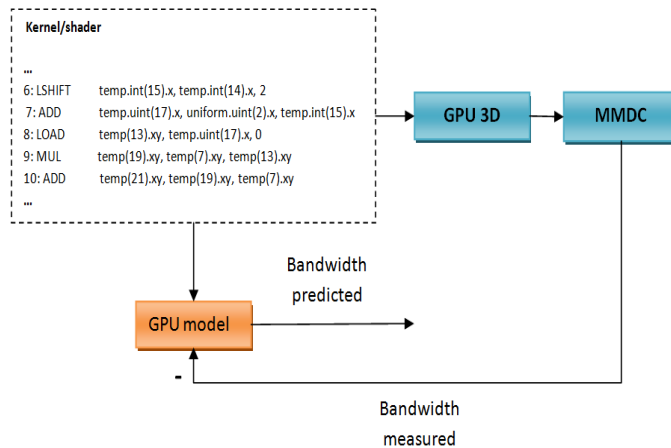


Abb. 11 – Feedback schleife für Bandbreitenabschätzung

Während den Analysen und Diskussionen mit Partnern von Universität Stuttgart und Daimler hat sich Kristallisiert das die härteste Herausforderung die Abschätzung der Laufzeit des Fragment Shader ist. Der Fragment Shader ist mehrere mahl abgerufen, abhängig von der Anzahl Fragmente die bearbeitet werden müssen. Die Anzahl Fragmente ist abhängig von den anderen Pipeline Schritten; eine genauere Bestimmung der Nummer der Fragmente auf dem Rechenkernen wäre unpraktisch (die GPU ist ein Akzelerator genau für diese Aufgabe). Unser Ansatz für dieses Problem ist „bounding box“ basiert; die 3D Elemente werden in „bounding boxes“ enkapsuliert und die Transformationen (Rotation, Translazon, etc.) werden nicht auf jeden Vertex gemacht sondern direkt auf der „bounding box“ (Abb. 12).

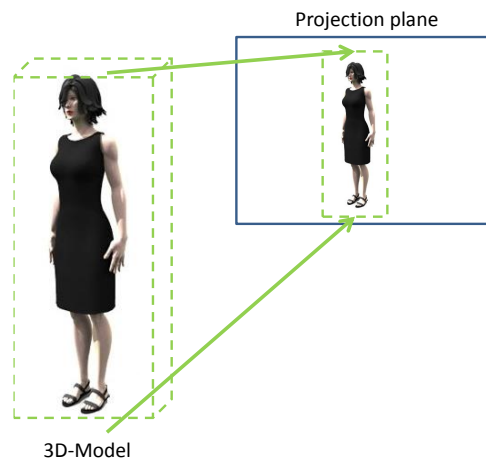
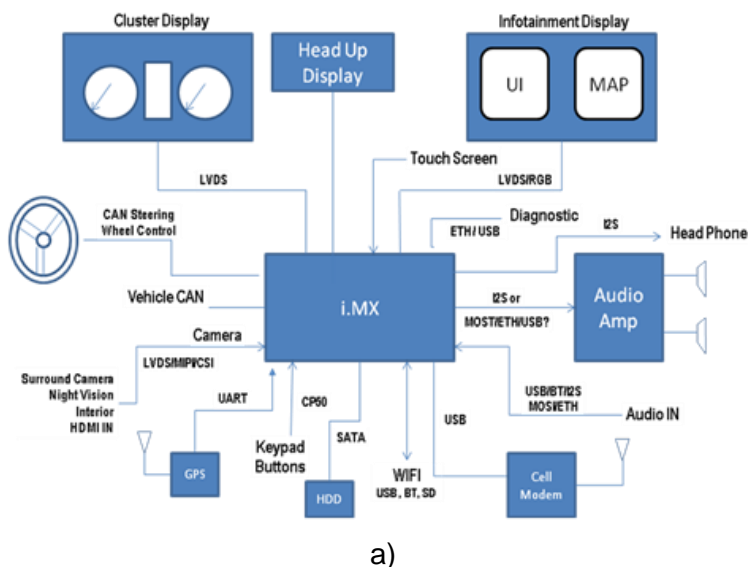
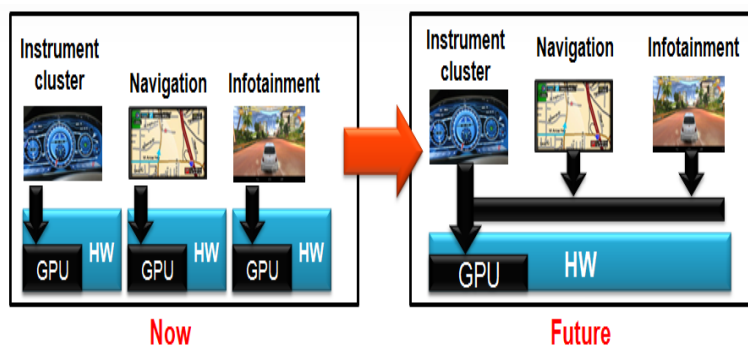


Abb. 12 – Bounding Box Fragmentabschätzung

In Rahmen von TP4.4 haben wir mögliche Virtualisierung Ansetze studiert. Das primere Szenario ist ein unifizierte Instrument Cluster und Infotainment System (Abb. 13).



a)



b)

Abb. 13 - unifizierte Instrument Cluster, Infotainment System und Navigation System; a) System detailliert; b) Konsolidierte Ansatz für Graphik Applikationen

Wir haben zwei Vorschläge für GPU Virtualisierung:

- a. Virtualisierung an OpenGL Ebene (Abb. 14)
- b. Virtualisierung an kernel Treiber Ebene (Abb. 15)

Die Vor und Nachteile sind in E4.4.2 diskutiert worden und die Variante **a.** ist auch von unseren Partnern Universität Stuttgart und Daimler implementiert in den TP6 Demonstratoren. Um diese Implementierung zu ermöglichen haben wir Demoapplikationen fuhr die Partner implementiert (2D GPU Composition, Extended

Desktop, Virtual Framebuffer etc.) und ihnen geholfen die zu integrieren.

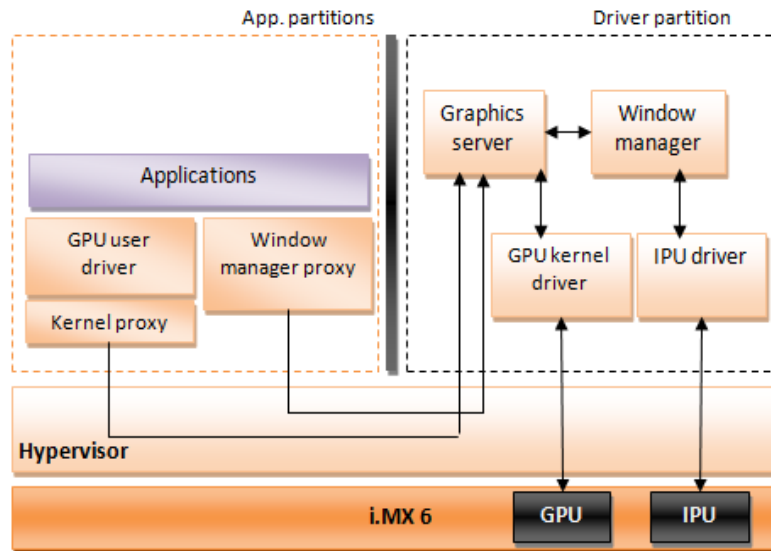


Abb. 14 - Virtualisierung an OpenGL Ebene

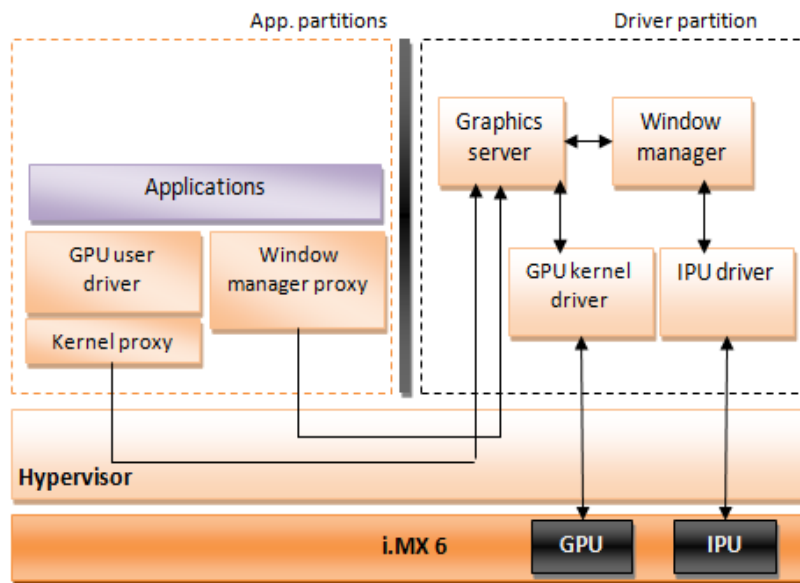


Abb. 15 - Virtualisierung an kernel Treiber Ebene

2.1.5 TP5 Durchgängige Entwicklungsmethodik und Werkzeuge

Freescale hat keine Beiträge in Rahmen von TP5 gehabt.

2.1.5.1 Aufgabenstellung in TP5

Freescale hat keine Beiträge in Rahmen von TP5 gehabt.

2.1.5.2 Ablauf und Ergebnisse von TP5

Wir haben zu Konsortiumpartnern Dokumentation zu Verfügung geachtelt die hilfreich für den design von Laufzeitabschätzung und Scheduling Werkzeugen ist. Dieser Detaillierungslevel ist meist nur intern verfügbar.

2.1.6 TP6 Demonstratoren

In Rahmen von TP6 sind Demonstratoren gemacht worden die die Erarbeiteten Konzepte in den vorherigen TPs, implementieren. Freescale hat primer bei den Automobil Domäne Demonstratoren beigetragen.

2.1.6.1 Aufgabenstellung in TP6

Wir haben uns in Rahmen von TP4 die folgenden Ziele gesetzt:

- Unseren Partner mit Hardware und Technische Unterstützung zu helfen
- Demo Applikationen für die Demonstratoren abzuliefern
- Beiträge zum Ergebnisdokumenten

2.1.6.2 Ablauf und Ergebnisse von TP6

In Rahmen von TP6 haben wir i.mx 6 Boards zu unseren Konsortiumpartner BMW und Daimler geliefert (Abb. 16). Wir haben Workshops für Inbetriebnahme der Hardware und Schulung für Hardware und Software organisiert. Wir haben mit verschiedenen Zulieferer zusammengearbeitet um Anpassungen zu Software und Hardware zu machen so dass die Demonstratoren Systeme Problemlos funktionieren (e.g. mit Display Adapter Zulieferer für PCB design; mit Fujitsu, für Candera Engine Anpassungen für multiple Displays).

Wir haben Workshops mit Daimler und Universität Stuttgart organisiert um System Debugging zu machen und um Lösungen für verschiedene Probleme zu finden:

- Multiple Graphic Context Probleme
- System Hang wegen multi-threading race conditions
- Window Manager Implementierung mit Hilfe von 2D und 3D GPU
- Virtual Framebuffer Limitierungen

- Analog Kamera Anschließung

Wir haben Beispiel Applikationen vorbereitet um bei den Partnern die Demonstratoren Softwareentwicklung zu erleichtern:

- Virtual FrameBuffer Kernel Driver
- 2D GPU Window composition
- Extended Desktop mit 3D GPU und Display Controller Synchronisierung
- Kamera Anbindung zu eine 3D Applikation

Die Beiträge zum Ergebnisdokumenten die Freescale abgeliefert hat umfassen Beschreibungen der Hardware Elementen auf den i.mx 6 SoC.

Freescale hat in Rahmen von TP6 auch eine P4080 Plattform zu KIT abgeliefert. Wir haben ein Workshop für Inbetriebnahme und Schulung organisiert.

DAIMLER



Abb. 16 – Demonstratoren wo Freescale Hardware und Software benutzt worden ist

2.2 Wichtigste Positionen des zahlenmäßigen Nachweises

In Projekt ARAMIS sind die wesentlichen Kosten Personalkosten (~95%). Die Hardware Boards die die Konsortiumpartner von uns bekommen haben sind direkt auf Freescale kosten gekauft worden. Kosten die mit den Support Aktivitäten für Hardware Systeme für das keinen Spezialisiertes personal in ARAMIS zu Verfügung stand sind auch von Freescale abgedeckt worden (e.g. Support von P4080, T4240 Networking Prozessoren).

2.3 Notwendigkeit und Angemessenheit der geleisteten Arbeit

Der Verlauf der Arbeit im Projekt folgte im Großen der im Projektantrag formulierten Planung. Mit der Internen Umstrukturierung haben wir versprochene Ressourcen nicht mehr bekommen und so mussten wir einige Teile der geplanten Aktivitäten aufgeben (e.g. Leitung von TPs, Test Implementierung von verschiedenen Virtualisierungs Ansätze für GPUs). Die wichtigen Studien (hinsichtlich neue Generationen von i.mx Produkten) in Bereich GPU Virtualisierung sind aber erfolgreich abgeschlossen worden. Mit den Arbeiten und Studien sind auch viele andere Fragen aufgetaucht (e.g. Bus Determinismus, Cache Replacement Policy Determinismus) die nicht detailliert genug in ARAMIS studiert worden sind.

2.4 Voraussichtlichen Nutzens, insbesondere der Verwertbarkeit des Ergebnisses im Sinne des fortgeschriebenen Verwertungsplans

Die Verwertung der Ergebnisse bei den einzelnen Projektpartnern ist detailliert in den einzelnen Erfolgskontrollberichten beschrieben. Detailliert auf den einzelnen TPs wurden wir folgende Aspekte hervorheben:

TP1 – Die Szenarien die von den OEMs (e.g. Daimler, BMW, etc.) vorgestellt worden sind haben ein Einblick in den nächsten Automobil und Flugzeug Systemen erlaubt. Die Anforderungen sind deutlicher gemacht worden und transparenter für alle Firmen in der Zulieferer Kette. Viele der Szenarien der Automotive Industrie werden Realität in 2018 Automobilen. Konsolidierte ansetze (e.g. Infotainment und Instrument Cluster) werden in Mittlerenpreissegment immer prädominierter Sein wegen der signifikanten Kosteneffizienz.

TP2 – Wir finden das der Abstraktionlevel von TP2 zu hoch gewesen war als in vielen Fällen notwendig. Diese Abstraktion ist deswegen eingeführt worden um eine unifizierte Sichtweise für verschiedenen Applikationen und Domänen benutzen zu können. Unsere Meinung nach wäre es besser gewesen Applikation spezifisch Anforderungen und Architekturen zu betrachten (ein Mix von Infotainment Anforderungen und Power Train Anforderungen verwirrt den Leser). Die Arbeiten in TP2 sind inhaltlich ein guter Startpunkt um Anforderungen für die nächsten Generationen Hardware zu extrahieren wen man von der verwirrenden Struktur abstrahiert.

TP3 – Die Studien die in TP3 gemacht worden sind haben viele interessante Verbesserung gebracht. In Bereich IO und Coprozessor Virtualisierung sind sehr gute Arbeiten abgeliefert worden. Es sind aber auch sehr viele Lücken in der Betrachtung geblieben, speziell in Bereich Vorhersagbarkeit

von chip interconnect (Spezialisierte Ingenieure in solchen Bereichen sind sehr selten).

TP4 – Wir denken dass die wichtigsten Beiträge in Rahmen von TP4 von den Coprozessoren (inklusive GPUs) Virtualisierung und OS scheduling Ansetzen kommen. Um mehr Funktionalitäten in die Autos einzubringen ist der Applikationkonsolidierungsansatz notwendig um die kosten zu Senken. Die Ergebnisse in Bereich GPU Virtualisierung sind für Freescale besonders wichtig in Hinblick zum kommenden i.mx 8, der in 2015 herausgebracht wird.

TP5 – Freescale hat diesen Teil des Projektes nicht verfolgt.

TP6 – Wir denken das die wichtigsten Erkenntnisse über die Performanz Notwendig in nächsten Hardware SoCs aus TP6 Demonstratoren gekommen ist. Die Aspekte die in Software zu keine effizienten Lösungen geführt haben werden in nächsten Generationen besser in Hardware unterstützt.

2.5 Erfolgskontrollberichten beschrieben Während der Durchführung des Vorhabens dem ZE bekannt gewordenen Fortschritts auf dem Gebiet des Vorhabens bei anderen Stellen

In Bereich multi-core sind viele Publikationen präsentiert worden wehrend der ARAMIS Laufzeit. In unseren GPU Virtualisierung Bereich, der nennenswerteste Fortschritt ist von Nvidia gekommen, das in 2013 die erste GPU die komplett Hardware Virtualisierung unterstützt angekündigt hat. Die GPU ist für Consumer Applikationen gedacht und berücksichtigt nicht die Automotive Safety Anforderungen und Energieeffizienz Notwendigkeiten aber es ist ein deutlicher Schritt vorwärts.

2.6 Veröffentlichungen des Ergebnisses

In den folgend werden nur externe Veröffentlichungen genannt, die internen Ergebnisdokumenten wo wir Beiträge haben sind aus den vorherigen Kapitel schnell identifizierbar.

Embedded Virtualization Approaches for Ensuring Safety and Security within E/E Automotive Systems, February 2015, Embedded World Congress

Anlage zum Schlussbericht: Veröffentlichungen

Freescal e Halbleiter GmbH

Version	0.1
Laufzeit des Vorhabens	01.12.2011 - 30.11.2014
Förderkennzeichen BMBF	01IS110352
Verbreitung	Öffentlich
Fälligkeitsdatum	31.05.2015
Datum	17.02.2015

GEFÖRDERT VOM



Bundesministerium
für Bildung
und Forschung

Versionen

Version	Datum	Beschreibung
0.1	17.02.2015	Erste Version , Krutsch Robert

Projektkoordination

Prof. Dr.-Ing. Jürgen Becker / Dr.-Ing. Oliver Sander
Karlsruher Institut für Technologie (KIT)
Institut für Technik der Informationsverarbeitung (ITIV)
Engesserstr. 5
76131 Karlsruhe
Telefon: +49 721 - 608 - 42502 / - 42512
Telefax: +49 721 - 608 - 42511
Email: becker@kit.edu / sander@kit.edu

© Copyright 2015 ARAMiS
Kordinator: Karlsruher Institut für Technologie (KIT)

Autoren

Dieses Dokument wurde erstellt von Krutsch Robert

Beiträge wurden verfasst von

Krutsch Robert, Freescale Halbleiter GmbH

Dr. Schlagenhaft Rolf, Freescale Halbleiter GmbH

Inhaltsverzeichnis

Versionen	ii
Projektkoordination	ii
Autoren	iii
Inhaltsverzeichnis	iv
1.1 Veröffentlichungen des Ergebnisses	1

1.1 Veröffentlichungen des Ergebnisses

In den folgend werden nur externe Veröffentlichungen genannt, die internen Ergebnisdokumenten wo wir Beiträge haben sind aus den Schlussbericht schnell identifizierbar.

Embedded Virtualization Approaches for Ensuring Safety and Security within E/E Automotive Systems, February 2015, Embedded World Congress

Berichtsblatt

1. ISBN oder ISSN -	2. Berichtsart (Schlussbericht oder Veröffentlichung) Schlussbericht
3. Titel ARAMiS Schlussbericht	
4. Autor(en) [Name(n), Vorname(n)] Robert Krutsch Dr. Rolf Schlagenhaft	5. Abschlussdatum des Vorhabens 31.11.2014
	6. Veröffentlichungsdatum -
	7. Form der Publikation -
8. Durchführende Institution(en) (Name, Adresse) Freescale Halbleiter GmbH Schatzboged 7, 81829 München	9. Ber. Nr. Durchführende Institution -
	10. Förderkennzeichen 01IS110352
	11. Seitenzahl 28
12. Fördernde Institution (Name, Adresse) Bundesministerium für Bildung und Forschung (BMBF) 53170 Bonn	13. Literaturangaben 1
	14. Tabellen 1
	15. Abbildungen 16
16. Zusätzliche Angaben -	
17. Vorgelegt bei (Titel, Ort, Datum) -	

18. Kurzfassung

ARAMIS hatte zum Ziel, durch den Einsatz von Multicore-Technologien in den Mobilitätsdomänen Automobil, Avionik und Bahn die technologische Basis zu erweitern und Lösungen zu erarbeiten die zu der Erhöhung von Sicherheit und Verkehrseffizienz beitragen. Multicore-Systeme sind Stand der Technik in vielen Bereichen (e.g. Consumer, Networking, vielen Automobil Applikationen); ARAMIS hat nicht versucht einen Plädoyer für diese Systeme zu geben (die positiven Eigenschaften sind sehr bekannt) sondern diese Systeme zu analysieren in Hinblick zu Applikationen in den drei genannten Domänen und Verbesserung vorzuschlagen so dass kritische Anforderungen von Bereichen wie Echtzeitfähigkeit, Funktionssicherheit, Zuverlässigkeit etc. erfüllt oder besser erfüllt werden können.

In wichtigen Ergebnisse in ARAMIS sind in Bereich Coprozessor (e.g. GPU) und I/O Virtualisierung. Diese Betrachtung von der Safety Perspektive, in ARAMIS, ist sehr wichtig für Automobil, Avionik und Bahn Mobilitätsdomänen.

Der Markt für Virtualisierungs basierenden Lösungen im Automotive Bereich wächst und wir erwarten, dass die nächsten Hardware Plattformen Unterstützung für Virtualisierung anbieten müssen, um wettbewerbsfähig zu bleiben. Die Lösungen die auf Virtualisierung basieren können Kosten sparen und safety relevante Systeme in eine breitere Masse von Automobilen bringen.

19. Schlagwörter

Virtualisierung, GPU virtualisierung, Safety

20. Verlag

-

21. Preis

-

Document Control Sheet

1. ISBN or ISSN -	2. type of document (e.g. report, publication) Report
3. title ARAMiS Schlussbericht	
4. author(s) (family name, first name(s)) Robert Krutsch Dr. Rolf Schlagenhaft	5. end of project 31.11.2014
	6. publication date -
	7. form of publication -
8. performing organization(s) (name, address) Freescale Halbleiter GmbH Schatzboged 7, 81829 Munchen	9. originator's report no. -
	10. reference no. 01IS110352
	11. no. of pages 28
12. sponsoring agency (name, address) Bundesministerium für Bildung und Forschung (BMBF) 53170 Bonn	13. no. of references 1
	14. no. of tables 1
	15. no. of figures 16
16. supplementary notes -	
17. presented at (title, place, date) -	
18. abstract <p>ARAMiS aimed, through the use of multi-core technologies in the domains automobile, avionics and rail to expand the state of the art and to develop solutions that contribute to the increase of safety and traffic efficiency. Multicore systems are state of the art in many areas (e.g. consumer, networking, many automotive applications) so the benefits of hardware platform consolidation are well known and understood. ARAMiS targeted an analysis of the multicore technology from the safety and security point of view, considering the applications in the three domains.</p> <p>The important results in ARAMiS are in the field of coprocessor (e.g. GPU) and I/O virtualization. The safety perspective is very important for the three application domains and was the “Leitmotiv” of the project.</p> <p>The market for virtualization-based solutions in the automotive sector is growing and we expect the next hardware platforms need to offer support for virtualization in order to remain competitive. The solutions may be based on virtualization to save costs and bring safety-relevant systems in a wider mass of automobiles.</p>	

19. keywords Virtualization, GPU Virtualization , Safety	
20. publisher -	21. price -