



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Ausarbeitung Seminar - WiSe 2010

Marco Kirschke

Symmetric Multiprocessing mit einer FPGA basierten
MPSoC Plattform

Inhaltsverzeichnis

1	Einleitung	2
1.1	Motivation	2
1.2	Ziele der Masterarbeit	3
2	Technologische Übersicht zu MPSoC Architekturen	4
2.1	Asymmetric Multiprocessing mit heterogenen MPSoC	4
2.2	Symmetric Multiprocessing auf Basis homogener MPSoC	6
3	Master- Arbeitsplan	8
3.1	Weiterentwicklung des „MicroBlaze SMP-Project“	8
3.2	Erstellung eines SMP MPSoC mit multiprozessorfähigem RTOS	9
3.3	Erzeugung von Multiprozessor Anwendungen	11
4	Risiken und alternative Vorgehensweise	11
5	Zusammenfassung	12

1 Einleitung

Die Verbindung von mehreren Prozessoren zu komplexen Multiprozessor Systemen stellt einen üblichen Ansatz dar, um die speziellen Anforderungen für eingebettete Systeme zu erfüllen. Der in diesem Zusammenhang typische Entwicklungsprozess besteht zunächst aus Entwurf und Erzeugung einer Hardware Plattform und der anschließenden Entwicklung der projektspezifischen Software Anwendungen.

Die Prozessoren werden abhängig von ihrer konkreten Aufgabe ausgewählt, und in das System integriert, um den Anteil nicht verwendeter Hardwareressourcen gering zu halten und Produktionskosten zu minimieren. Die Adaption dieser Vorgehensweise auf den Entwurf von MPSoC¹ auf Basis von FPGA Technologie bietet einen entscheidenden Vorteil. Die Hardwarekonfiguration von „klassischen“ eingebetteten Systemen ist nach Fertigstellung des Boarddesigns nur noch schwer abzuändern, dagegen können FPGA basierte Systeme während des gesamten Entwicklungsprozesses und auch zu späteren Zeitpunkten (durch z. B. Firmware Updates) angepasst werden. Die Begrenzung liegt bei diesen Systemen lediglich in der Anzahl der verfügbaren Logikelemente des verwendeten FPGAs [vgl. Dorta u. a. (2009)].

1.1 Motivation

Die fortschreitende Entwicklung der FPGA Technologie, mit der wachsenden Anzahl an Logik Elementen auf den FPGA Bausteinen und der zunehmenden Verfügbarkeit von Softcore Prozessoren und IP-Cores (*Intellectual property cores*)², erweitert die Einsatzgebiete von FPGA-basierten MPSoC im Bereich der eingebetteten Systeme. Neben der für diese Art von Systemen typischen Konfiguration, bestehend aus einer Anzahl heterogener Prozessoren, die spezielle Teilaufgaben erfüllen, lässt diese Entwicklung mittlerweile die Erzeugung komplexer, homogener Prozessorarchitekturen zu. Dabei kann die konkrete Bearbeitung der Anforderungen an das System nun vielmehr durch die Software Entwicklung realisiert werden.

Daraus entsteht für den Entwicklungsprozess zum einen der Vorteil einer Entkopplung des „klassischen“ Hardware/Software-Codesign Ansatzes, da Entwurf und Implementierung der Hardware Plattformen flexibel gestaltet werden können und wiederverwendbare, nicht projektspezifischen Hardwarekonfigurationen entstehen. Weitere Vorteile sind in der höheren Verfügbarkeit von qualifizierten Software Entwicklern und der allgemein kürzeren Entwicklungszeit von Software Projekt im Vergleich zu Hardware Designs zu sehen.

Auf der anderen Seite ergeben sich bei der Entwicklung von SMP MPSoC besondere

¹ *Multiprocessor systems on chip*

² IP-Cores bezeichnen einen in sich geschlossenen, wiederverwertbaren Teil eines Chipdesigns, der meist über Lizenzen bezogen und in unterschiedliche Hardware Plattformen integriert werden kann.

Anforderungen an die verwendeten Betriebssysteme, da diese das Scheduling und die Verteilung der Softwareprozesse auf die Prozessoren vornehmen müssen.

Gleichzeitig bleiben jedoch die in Kirschke (2010b) beschriebenen Vorteile FPGA-basierter Systeme erhalten. Die Integration von zusätzlichen Hardwarebeschleunigungskomponenten und aufgabenspezifischen Coprozessoren in digitaler Logik kann auch bei diesen MPSoC abhängig von den verfügbaren Systemressourcen vorgenommen werden.

1.2 Ziele der Masterarbeit

Als Zielsetzung für die Masterarbeit soll das Hardware/Software Codesign für eine FPGA-basierte SMP MPSoC Plattform durchgeführt werden. Diese soll als Prototyp für weitere Entwicklungen innerhalb der HW/SW-Codesign Gruppe an der HAW dienen und zu einem späteren Zeitpunkt in die laufenden Projekte integriert werden. Durch die Verwendung eines adäquaten multiprozessorfähigen Echtzeitbetriebssystems (RTOS - *real time operating system*) soll weiterhin die einfache Entwicklung und Integration von zusätzlichen Softwareanwendungen unterstützt werden.

Die Einteilung in hardware- und softwarebezogene Anforderungen lässt eine detaillierte Formulierung der Zielsetzung zu. Dabei müssen die Eigenschaften von verfügbaren FPGA Bausteinen und Entwicklungsboards mit den Anforderungen der RTOS verglichen werden. Zusätzlich müssen die verwendeten Hardware Komponenten so gewählt werden, dass eine spätere Erweiterung ohne großen Aufwand vollzogen werden kann. Bei der Auswahl des RTOS ist dagegen die Verfügbarkeit von Software Standard Bibliotheken und Treibern für General Purpose Peripherie von entscheidender Bedeutung.

2 Technologische Übersicht zu MPSoC Architekturen

In FPGA basierten MPSoC erfolgt die Anordnung der Prozessoren anhand einer konkreten Systemanforderung. Abhängig von der zu erstellenden Anwendung wird entschieden, welche Kommunikations- und Synchronisationsverfahren für die Prozessoren untereinander eingesetzt werden sollen. Diese Eigenschaft spiegelt einen wesentlichen Vorteil von eingebetteten Systemen wider, da somit die Hardwareressourcen eines Systems unmittelbar an die Systemvorgaben angepasst werden können. Dabei kann der Aufbau von MPSoC in zwei Gruppen unterteilt werden: den Asymmetric Multiprocessing Architekturen (AMP) und den Symmetric Multiprocessing Architekturen (SMP).

2.1 Asymmetric Multiprocessing mit heterogenen MPSoC

Die Verwendung heterogener Prozessoren in einem MPSoC stellt den typischen Einsatz von Multiprozessor Technologie in eingebetteten Systemen dar. Die Prozessoren sind auf die Bearbeitung definierter Teilaufgaben optimiert und kommunizieren oder synchronisieren sich nicht zwangsläufig, sondern nur bei Bedarf abhängig von der konkreten Anwendung. Die Ergebnisse ihrer Berechnungen tauschen die Prozessoren dabei über spezielle Kommunikationsmodule oder einen gemeinsam verwendeten Speicher aus. Beim AMP werden darüber hinaus die Aufgaben von einem zentralen Hauptprozessor an die untergeordneten Prozessoren verteilt. Diese Koordination an zentraler Stelle hat zum einen den Vorteil, dass zusätzliche Module ohne großen Aufwand in das System integriert werden können, zum anderen besteht ein wesentlicher Nachteil darin, dass bei Ausfall des Hauptprozessors das gesamte System nicht mehr verfügbar ist.

Ein aktuelles Beispiel einer solchen heterogenen AMP Architektur wurde in der Ausgabe *Issue 72, Third Quarter 2010* des *Xcell Journal* vorgestellt [vgl. Thompson (2010)]. Der Artikel behandelt den Aufbau eines Fahrerassistenzsystems, welches das Fahrverhalten unter Berücksichtigung von sicherheitstechnischen und ökonomischen Aspekten aufzeichnet und analysiert. Die Entwicklung der Hardware Plattform wurde von der Firma TRW Conekt im Rahmen des „Foot-LITE“ Projektes [vgl. Foot-LITE (2011)] vorgenommen. Dieses Projekt wird von einem Konsortium aus Unternehmen und Hochschulen sowie der britischen Regierung unterhalten und versucht den Fahrzeugnutzern durch die Bereitstellung von Informationen ein Bewusstsein für sicheres und umweltverträgliches Fahren zu vermitteln.

Das System sammelt die Informationen zum Fahrverhalten mittels Bewegungssensoren, einer VGA Kamera und der Motordaten des Fahrzeugs. Die Auswertungen werden direkt über Bluetooth auf einem Smartphone im Fahrgastraum dargestellt, was die Installation des Systems in verschiedenen Fahrzeugtypen erleichtert.

Die Hardware Plattform besteht aus einem Xilinx Spartan-3A XC3SD3400A FPGA Baustein, auf dem vier MicroBlaze Softcore Prozessoren implementiert wurden [vgl. Abbildung 1].

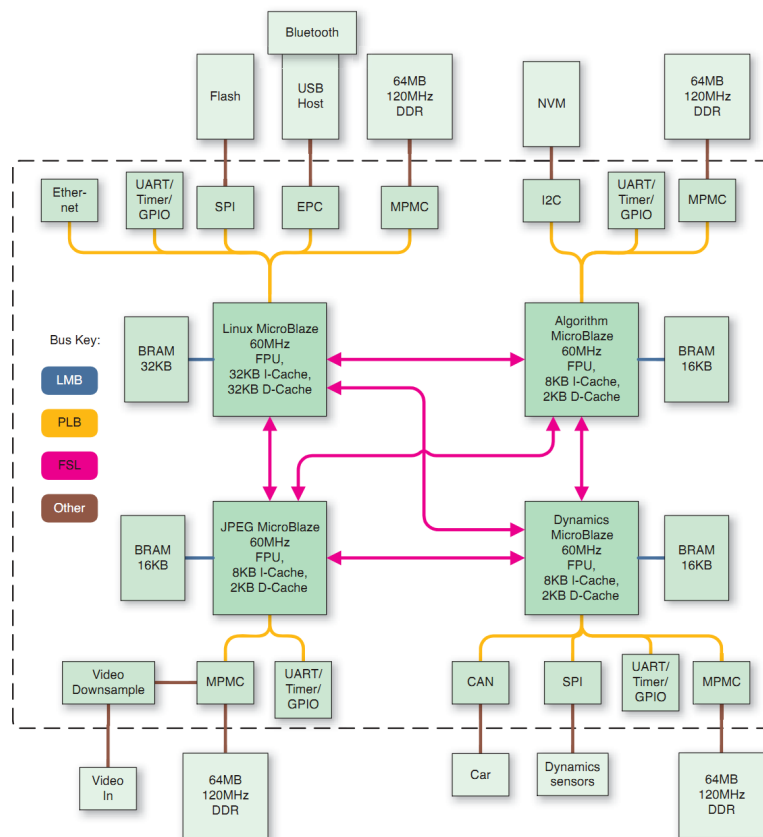


Abbildung 1: Anordnung von vier heterogenen MicroBlaze Prozessoren im System des Foot-LITE Projektes[Foot-LITE (2011)]

- Der *Linux MicroBlaze* wird als Hauptprozessor eingesetzt. Durch die Verwendung eines auf ucLinux basierenden Betriebssystems und eines einfachen JFFS2 Filesystems kann die Anwendungssoftware für alle vier Prozessoren jederzeit aktualisiert werden; zusätzlich ist so eine einfache Einbindung der erforderlichen Linux-Treiber für die USB Schnittstelle und des Bluetooth zur Kommunikation mit dem Smartphone gewährleistet.
- Innerhalb des *Algorithm MicroBlaze* werden die gesammelten Fahrzeugdaten und Umgebungsparameter mit Hilfe des Foot-LITE Algorithmus ausgewertet und an den Hauptprozessor übertragen. Der Algorithmus lässt sich zudem über Parameter konfigurieren, die über I2C Bus aus einem Non Volatile Memory bei Systemstart gelesen werden.
- Die Sensoren zur Ermittlung der Fahrzeugausrichtung sind mit dem *Dynamics MicroBlaze* über eine SPI (*Serial Peripheral Interface*) Schnittstelle verbunden; zusätzlich

werden an dieser Stelle Daten über den fahrzeugeigenen CAN (*Controller Area Network*) Bus empfangen und für die Verwendung im Foot-LITE Algorithmus aufbereitet.

- Über den *JPEG MicroBlaze* werden die Daten der VGA Kamera aufgenommen, wobei Informationen zur Fahrbahnbreite, Fahrzeuglage und Entfernungen zu umgebenden Objekten berechnet werden. Diese Daten werden ebenfalls an den *Algorithm MicroBlaze* zur Auswertung übertragen. Darüber hinaus werden bestimmte Fahrsituationen aufgezeichnet, um diese zu einem späteren Zeitpunkt analysieren zu können.

Die Prozessoren kommunizieren untereinander ausschließlich über den Fast Simplex Link (FSL) und verwenden keinen gemeinsamen Speicher. Dadurch können die einzelnen Teilsysteme unabhängig voneinander entwickelt und geändert werden. Jedem Prozessor steht ein eigener Speicherbereich im Block RAM des FPGAs und jeweils 64 MB externes DDR-RAM zur Verfügung.

Dieses Fahrerassistenzsystem spiegelt, durch den Einsatz unterschiedlicher Prozessoren zur Erfüllung der einzelnen Teilaufgaben, den typischen Aufbau eines FPGA-basierten eingebetteten Systems wider.

2.2 Symmetric Multiprocessing auf Basis homogener MPSoC

Einen anderen Ansatz als das oben beschriebene asynchron agierende System, verfolgen homogen aufgebaute, synchron arbeitende MPSoC. In solchen Systemen werden mehrere identische Prozessoren eingesetzt, wobei alle gleichberechtigt auf einen gemeinsamen Speicherbereich oder eine Peripherie Schnittstelle zugreifen können. Ein zentraler Bestandteil dieser Systeme ist der Einsatz eines multiprozessorfähigen Betriebssystems, das einerseits den Zugriff auf den Speicher synchronisieren und andererseits das Scheduling der Softwareprozesse auf die einzelnen Prozessoren vornehmen muss. Aus Sicht der Softwareentwicklung ist dabei nicht vorhersehbar, welcher Prozess auf welchen Prozessor ausgelagert wird. Es existiert somit keine feste Zuordnung der Prozesse wie bei AMP gestützten Systemen. Bei der Entwicklung von SMP-fähigen MPSoC ist darüber hinaus zu beachten, dass bei steigender Anzahl der Prozessoren, durch die Verwendung des gemeinsam genutzten Speicherbereiches, der Speicherbus zu einem „Flaschenhals“ werden kann. Die Leistungssteigerung des Systems ist also nicht zwangsläufig von der Anzahl der implementierten Softcore Prozessoren abhängig. Weiterhin besteht die Gefahr, dass der Scheduler des Betriebssystems einen Prozess alternierend auf unterschiedliche Prozessoren verteilt (*CPU Hopping*), wodurch bei der Verwendung von Caches unnötige Reloads die Geschwindigkeit der Berechnung verzögern können. Die Verwendung von erweiterten Speicher- und Cache Kohärenzstrategien ist daher bei der Verwendung von symmetrisch Multiprozessorsystemen von entscheidender Bedeutung [vgl. Tanenbaum (2006)].

Als Beispiel einer solchen SMP-fähigen homogen angeordneten Prozessorarchitektur ist das „MicroBlaze SMP-Projekt“ der Universidad Rey Juan Carlos, Mostoles zu nennen. Dazu wurden vier verschiedene Hardware Plattformen entwickelt, die sich in Bezug auf die Verwendung von gemeinsam genutzten Speicherbereichen und der Anordnung von Caches unterscheiden. Zusätzlich wurden parallel ausführbare Anwendung, eine Matrixmultiplikation und das Verschlüsseln bzw. Entschlüsseln mit Hilfe des AES-Algorithmus³, geschrieben und auf den Systemen ausgeführt, um Messwerte für die verbesserte Rechenleistung bei Einsatz mehrerer Prozessoren in Abhängigkeit zu den verbrauchten FPGA Ressourcen zu ermitteln. Die Systeme wurden auf Xilinx Virtex-2 bzw. Virtex-4 FPGAs unter Verwendung von Xilinx MicroBlaze Softcore Prozessoren und IP-Cores implementiert. Das verwendete Xilinx Xilkernel Betriebssystem stellt als Funktionalitäten zwar das Scheduling von POSIX Threads und einfache Interprozess-Synchronisations- und Kommunikationsmechanismen, wie *mutex*, *semaphores* und *mailboxes*, zur Verfügung, beinhaltet aber keinen Multiprozessor fähigen Scheduler, der parallele Anwendungen auf die Prozessoren verteilt. Aus diesem Grund musste eine Anpassung des Kernels vorgenommen werden [vgl. Huerta u. a. (2008)].

Des Weiteren mussten zusätzliche Hardware Komponenten entwickelt werden, um den exklusiven Zugriff auf den gemeinsam verwendeten Speicher zu realisieren (*hw_mutex*) und die Identifikation der Prozessoren unabhängig von der Software Anwendung zu gewährleisten. Die Identifikation erfolgt über ein einfaches FSL- Register, in das die `ProzessorID` bereits während des Synthesevorganges geschrieben wird.

Die Prozessoren aller Systeme wurden darüber hinaus mit einem eigenen privaten Speicher versehen, der als Instruktions-Stack verwendet wird. In allen Systemen wird der Xilinx OPB (*on-chip peripheral bus*) verwendet, um die Prozessoren mit dem externen, gemeinsam genutzten Speicher zu verbinden. Im ersten System wird der gemeinsame Speicher allerdings direkt im Block RAM des FPGAs angelegt. Um die Verbesserung der Rechenleistung messen zu können, wurden die einzelnen Systeme nacheinander mit steigender Anzahl an Prozessoren implementiert.

Im zweiten System wurden bis zu acht MicroBlaze verwendet [vgl. Abbildung 2], wobei keine signifikante Leistungssteigerung mehr bei der Verwendung von fünf Prozessoren beobachtet werden konnte. Da in diesem System lediglich Instruktionscaches und keine Daticaches verwendet wurden, ist der OPB bereits bei 4 MicroBlaze Prozessoren ausgelastet. Eine detaillierte Beschreibung des Aufbaus der Systeme und die genauere Auswertung der Laufzeitmessungen wurde bereits in Kirschke (2010a) beschrieben oder ist direkt bei Huerta u. a. (2009) zu finden.

³Advanced Encryption Standard ist auch als *Rijndael*-Algorithmus bekannt

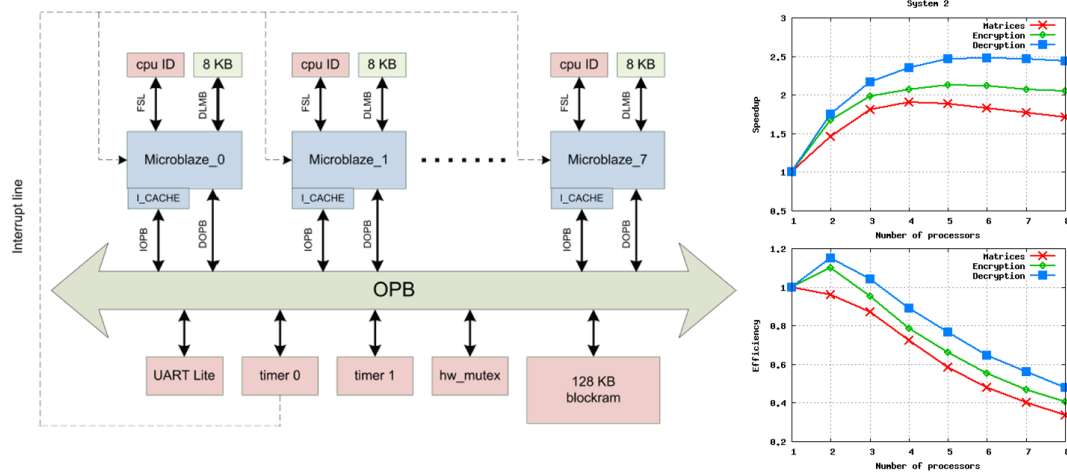


Abbildung 2: Im Testsystem mit acht MicroBlaze Prozessoren wird die Leistungssteigerung durch die erhöhte Auslastung des OPB begrenzt[Huerta u. a. (2009)]

3 Master- Arbeitsplan

Im Folgenden wird die konkrete Vorgehensweise bei der Erstellung der Masterarbeit beschrieben. Diese gliedert sich in zwei Bereiche; im ersten Schritt sollen grundlegende Erfahrungen und Kenntnisse bei der Erzeugung von SMP MPSoC erarbeitet werden. Dazu soll das in Kapitel 2.2 beschriebene SMP Projekt mit dem angepassten Xilkernel Betriebssystem auf einem ML605 Entwicklungsboard in Betrieb genommen werden. In einem zweiten Schritt soll ein multiprozessorfähiges Echtzeitbetriebssysteme auf einem Xilinx FPGA erstellt werden, so dass parallele Softwareanwendungen auf diesem System ausgeführt werden. Dazu ist eine genauere Analyse der verfügbaren RTOS sowie der dazu kompatiblen FPGA Plattform erforderlich.

3.1 Weiterentwicklung des „MicroBlaze SMP-Projekt“

Das „MicroBlaze SMP-Projekt“ der Universidad Rey Juan Carlos, Mostoles wurde im Jahr 2008 auf dem XUP Virtex-II Pro Development System Board der Firma Xilinx entwickelt. Die Erstellung des System Designs wurde mit den entsprechenden Xilinx Werkzeugen, XPS und ISE (*Xilinx Plattform Studio und Integrated Software Environment*) in der Version 8.2, vorgenommen. Die derzeitige Version der Xilinx Entwicklungsumgebung, XPS Version 12.4, unterstützt den in den Testsystemen verwendeten OPB nicht mehr; für die Inbetriebnahme muss daher zunächst eine neues Hardware Design erstellt werden, in dem ein alternatives Bussystem verwendet wird. Die Portierung soll auf dem Virtex-6 FPGA des ML605 Entwicklungsboard erfolgen, wobei die aktuellste Version 8.0 des MicroBlaze Softcore Prozessors

verwendet werden soll. Daraus ergeben sich folgende Anforderungen an das Hardwaredesign:

- Einbindung der in Huerta u. a. (2009) vorgestellten zusätzlichen Hardware Komponenten zur Identifikation der Prozessoren. Zusätzlich stellt Xilinx mittlerweile ein eigenes Hardware Mutex IP zur Verfügung, dass in dem neuen System zum Einsatz kommen soll.
- Ersetzung des OPB Systembusses; dazu stehen der PLB⁴ oder das AMBA AXI4 Protokoll⁵ zur Verfügung.
- Zusätzlich ist die Integration von Speicher Controller IPs geplant, um die Auslastung des Systembusses durch die Verwendung von Caches zu entlasten.

In einem zweiten Schritt müssen die softwaretechnischen Aspekte des „MicroBlaze SMP-Project“ in die erzeugte Hardware Plattform integriert werden.

- Der in Huerta u. a. (2008) beschriebene, Multiprozessor unterstützende Scheduler muss in die Umgebung des aktuellen Xilkernel Betriebssystems Version 4.0 eingebunden werden.
- Beispielanwendungen, mit denen das Laufzeitverhalten und die Optimierungsparameter ermittelt werden können, müssen innerhalb neuer Software Projekte mit Hilfe des Xilinx SDK (*Software Development Kit*) erstellt und auf die Zielplattform abgestimmt werden.

3.2 Erstellung eines SMP MPSoC mit multiprozessorfähigem RTOS

Eine weitere Zielsetzung für die Masterarbeit ist die Erzeugung eines SMP MPSoC, welches in den verschiedenen Projekten der HW/SW-Codesign Gruppe verwendet werden kann. Dazu werden besondere Anforderung an dessen Konfigurationsmöglichkeiten, aber auch an die Skalierbarkeit der Plattform in Bezug auf die Anzahl der Prozessoren und der eingebundenen Peripherie Komponenten gestellt. In Hinblick auf die Auswahl des Echtzeitbetriebssystems müssen nicht nur diese Vorgaben erfüllt werden, sondern neben der zwingend erforderlichen

⁴Der PLB (*processor local bus*) ist ein von IBM für den Einsatz in SoC entwickelter Prozessor Bus, der einen 64-bit Adressbus und 128-Bit Datenbus bereitstellt, um Prozessoren mit den entsprechenden Speicher Controllern zu verbinden.

⁵Das AXI4 (*Advanced eXtensible Interface*) Protokoll ist Teil der AMBA (*advanced microcontroller bus architecture*) Protokollfamilie, einem von der Firma ARM betriebenen, offenen Standard für Mikrocontroller Bussysteme, und wurde für die aktuellsten Versionen der Spartan-6 und Virtex-6 FPGAs von Xilinx als Bussystem eingeführt [vgl. AXI-Reference-Guide (2010)]

Verfügbarkeit eines Multiprozessor unterstützenden Schedulers, auch die allgemeine Verbreitung und die Verwendung des RTOS durch renommierte Hersteller berücksichtigt werden, um einen langfristigen Einsatz zu gewährleisten. Ausgehend von der Verwendung des bereits oben erwähnten ML605 Entwicklungsboard mit dem Virtex-6 FPGA muss dessen Kompatibilität mit den Vorgaben der RTOS Hersteller abgeglichen werden, um den Aufbau der Speicher- und Cachearchitekturen zu erfüllen. Nachfolgend werden zwei Betriebssysteme vorgestellt, die nach einer ersten Recherche als Kandidaten für das zu entwickelnde SMP MPSoC in Frage kommen.

- **QNX Neutrino RTOS [QNX (2011)]**

Das Neutrino Betriebssystem der Firma QNX basiert auf einer echten Microkernel Architektur, in der nur die fundamentalen Dienste, wie Signale, Timer und Scheduler innerhalb des Kernels angelegt sind. Alle anderen Software Komponenten (Periphereitreiber, Dateisysteme, User Interfaces oder Netzwerkkomponenten) laufen in sogenannten „user spaces“, die eigene Speicheradressräume besitzen und bei Bedarf zur Laufzeit neu gestartet werden können. QNX entwickelt bereits seit 1980 RTOS, die in missionskritischen, eingebetteten Systemen eingesetzt werden. Das QNX Neutrino unterstützt zudem den Einsatz von mehreren Prozessoren in AMP-, SMP- oder BMP-Systemen. BMP (*bound multiprocessing*) beschreibt dabei eine von QNX entwickelte Technologie, bei welcher der SMP Ansatz dahingehend erweitert wurde, dass die Prozesse und Threads bestimmten Prozessoren zugewiesen werden können. Das RTOS ist für die gängigen Prozessorarchitekturen (x86, SH4, MIPS, PowerPC, ARM) verfügbar.

Im Zusammenhang mit der Erstellung von Softwareanwendungen ist zu beachten, dass durch eine Vielzahl von verfügbaren Treibern die Einbindung externer Komponenten begünstigt wird. Neutrino unterstützt den POSIX Standard, womit einerseits die gängigen Mechanismen zur Interprozesskommunikation verfügbar sind und andererseits die Portierung von Softwareprojekten aus UNIX- oder Linux Umgebungen vereinfacht wird.

Es sind bereits eine Reihe von Board Support Packages für ältere Xilinx Entwicklungsboards vorhanden, aus denen eventuell eine Verwendung mit dem ML605 abgeleitet werden kann. Ein weiterer Punkt, der für die Verwendung von QNX Neutrino spricht, liegt in der Ankündigung von Seiten Xilinxs, in Zukunft FPGAs mit festintegrierten ARM Cortex-A9 MPCore Prozessoren auszurüsten [vgl. Wilson (2010)], zu welchen Neutrino schon jetzt kompatibel ist.

- **eT-Kernel der Firma eSOL [eSol (2011)]**

Als eine Alternative zu QNX Neutrino RTOS kann das extended T-Kernel der Firma eSol verwendet werden. Dieses stellt ein Multiprozessor unterstützendes Echtzeitbetriebssystem für embedded systems dar, das über die „Kernel Protection“ - „Core Par-

tioning“ Technologien sichere Zugriffe auf Speicherbereiche gewährleistet. Dabei werden die Software Anwendungen in eigenen Partitionen angelegt; die Interprozesskommunikation erfolgt über spezielle APIs. Durch die Skalierbarkeit des eT-Kernel kann es an die Gegebenheiten der Zielplattformen auf Kosten des Funktionsumfanges angepasst werden. Das RTOS wird bereits in einer Vielzahl von Produkten wie Navigationssystemen, digitalen Aufnahmegeräten aber auch in japanischen Satellitensystemen eingesetzt.

3.3 Erzeugung von Multiprozessor Anwendungen

Neben der Erstellung der oben beschriebenen Hardware Plattformen und der Integration der RTOS, ist die Erstellung von Software Anwendungen geplant, mit deren Hilfe verschiedene Konfigurationen der Ausgangssysteme auf Laufzeitorientierung und Ressourcenverbrauch getestet werden können. Dazu ist einerseits die Erstellung von Testanwendungen, wie sie auch in Huerta u. a. (2009) verwendet wurden, geplant, andererseits sollen bereits bestehende Anwendungen der HW/SW-Codesign Gruppe aus dem Umfeld des Faust Projektes auf die Multiprozessor Systeme angepasst und eingesetzt werden.

4 Risiken und alternative Vorgehensweise

Aus der Zielsetzung und der vorangegangenen Beschreibung des Arbeitsplanes ergeben sich eine Reihe von Teilaufgaben, deren erfolgreiche Bearbeitung naturgemäß auch mit Risiken verbunden ist. Dabei beziehen sich diese weniger auf den ersten Teil der Masterarbeit, der Weiterentwicklung des „MicroBlaze SMP-Projects“ und dessen Einsatz in der HW/SW-Codesign Gruppe der HAW; vielmehr liegen die Risiken im zweiten Teil, der Inbetriebnahme eines etablierten Multiprozessor RTOS. Unter Berücksichtigung der Tatsache, dass die Anfertigung der Masterarbeit mit der Einarbeitung in eine neu verfügbare Technologie einhergeht, kann die Unterstützung durch die Hersteller mit Dokumentationen und Arbeitsanweisungen noch nicht abgeschätzt werden. Dieser Umstand bezieht sich sowohl auf die Erstellung der Hardware Plattform als auch auf die Integration des Betriebssystems. Der zeitliche Rahmen ist bei der Erforschung von Grundlagen erfahrungsgemäß nicht abschätzbar und stellt mit der Anforderung, die Abgabe der Masterarbeit vor Ende des Jahres durchzuführen, eine weitere Unsicherheit dar.

Da zum jetzigen Zeitpunkt keine bestimmte Aussage zur Kompatibilität der in Kapitel 3.2 vorgestellten Betriebssysteme mit den derzeit verfügbaren Xilinx FPGAs vorgenommen werden kann, steht als Alternative die Verwendung des OMAP4430/OMAP4440 Dual-Core ARM Cortex-A9 Entwicklungsboards der Firma Texas Instruments zur Verfügung [vgl. OMAP4 (2011)]. Auf diesem System kann, mit Hinblick auf die spätere Veröffentlichung von Xilinx FPGAs, die ARM Cortex-A9 MPCore Prozessoren unterstützen [vgl. Wilson (2010)], das

QNX Neutrino RTOS in Betrieb genommen werden und erste Ergebnisse dokumentiert werden.

Weiterhin ist die Verwendung eines anderen, hier nicht aufgeführten Betriebssystems in Abhängigkeit des Projektverlaufes denkbar. Allerdings müssen die oben beschriebenen Eigenschaften und eine langfristige Verfügbarkeit gewährleistet sein.

5 Zusammenfassung

Die Weiterentwicklungen in den vergangenen Jahren in den Bereichen der eingebetteten Systeme und der FPGA Technologie lassen die Erstellung immer komplexerer Multiprozessor Systeme zu. Neben der für diese Systeme typischen Anordnung der Prozessoren in AMP Architekturen, können durch diese Entwicklung nun vielmehr SMP MPSoC erstellt werden, die flexiblere Hardware Plattformen und einen zunehmend auf die Softwareentwicklung abgestimmten Herstellungsprozess begünstigen.

Diese Vorgehensweise soll in die Projekte der HW/SW-Codesign Gruppe an der HAW überführt werden, indem im Rahmen der Masterarbeit eine skalierbare Plattform geschaffen wird, auf der die Implementierung von parallelen Anwendungen erfolgen kann. Dazu sollen in einem ersten Schritt Erfahrungen und Kenntnisse bei der Integration eines Multiprozessor unterstützenden Echtzeitbetriebssystems in ein Hardware Design anhand der Weiterentwicklung des „MicroBlaze SMP-Project“ der Universidad Rey Juan Carlos, Mostoles erarbeitet werden. Zusätzlich soll im zweiten Teil der Masterarbeit ein verfügbares, etabliertes RTOS, wie QNX Neutrino RTOS oder eT-Kernel, in eine aktuelle Hardware Plattform integriert werden, um die Ausführung paralleler Anwendungen zu unterstützen.

Literatur

- [AXI-Reference-Guide 2010] AXI-REFERENCE-GUIDE: *Xilinx AXI Reference Guide*, 2010. – from companies webpage [February 2011]
- [Dorta u. a. 2009] DORTA, Taho ; JIMÉNEZ, Jaime ; MARTÍN, José L. ; BIDARTE, Unai ; ASTARLOA, Armando: *Overview of FPGA-Based Multiprocessor Systems*. In: *2009 International Conference on Reconfigurable Computing and FPGAs*, URL <http://ieeexplore.ieee.org>, 2009. – IEEE Digital Library [May 2010]. – ISBN 978-0-7695-3917-1
- [eSol 2011] ESOL: *Extended T-Kernel RTOS for embedded systems*, 2011. – URL <http://www.esol.co.jp/english/embedded/et-kernel.html>. – information from companies webpage [January 2011]
- [Foot-LITE 2011] FOOT-LITE: *Foot-LITE - enabling green & save driving*, 2011. – URL <http://www.foot-lite.net/index.php>. – MIRA Limited. from project webpage [January 2011]
- [Huerta u. a. 2009] HUERTA, Pablo ; CASTILLO, Javier ; PEDRAZA, Cesar ; CANO, Javier ; MARTINEZ, Jose I.: *Symmetric multiprocessor systems on FPGA*. In: *2009 International Conference on Reconfigurable Computing and FPGAs*, URL <http://ieeexplore.ieee.org>, 2009. – IEEE Digital Library [April 2010]. – ISBN 978-1-4244-5293-4
- [Huerta u. a. 2008] HUERTA, Pablo ; CASTILLO, Javier ; SÁNCHEZ, Carlos ; MARTÍNEZ, Jose I.: *Operating System for Symmetric Multiprocessors on FPGA*. In: *2008 International Conference on Reconfigurable Computing and FPGAs*, URL <http://ieeexplore.ieee.org>, 2008. – IEEE Digital Library [October 2010]. – ISBN 978-0-7695-3474-9
- [Kirschke 2010a] KIRSCHKE, Marco: *Anwendungsgebiete FPGA-basierter MPSoC Architekturen*. August 2010. – Hochschule für Angewandte Wissenschaften - Hamburg
- [Kirschke 2010b] KIRSCHKE, Marco: *Implementierungsansätze für ein FPGA basiertes Multiprozessorsystem*. January 2010. – Hochschule für Angewandte Wissenschaften - Hamburg
- [OMAP4 2011] OMAP4: *OMAP 4 Platform: OMAP4430/OMAP4440*, 2011. – URL <http://focus.ti.com/general/docs/wtbu/wtbuproductcontent.tsp?templateId=6123&navigationId=12843&contentId=53243>. – information from Texas Instrument's webpage [January 2011]
- [QNX 2011] QNX: *QNX Neutrino RTOS*, 2011. – URL <http://www.qnx.com/products/neutrino-rtos/neutrino-rtos.html>. – information from companies webpage [January 2011]

- [Tanenbaum 2006] TANENBAUM, Andrew S.: *Parallele Rechnerarchitekturen*. Kap. 8. In: *Computerarchitektur*. München : Pearson Studium, 2006. – ISBN 3-8273-7151-1
- [Thompson 2010] THOMPSON, Martin: *Multiple MicroBlaze Ease Integration in Real-Time Automotive System*. S. 18–23. In: *Xcell Journal* No. 72, 3rd Quarter 2010
- [Wilson 2010] WILSON, Richard: *Xilinx hardwires Cortex-A9 MP-Core processor into FPGA*, 2010. – URL <http://www.electronicsworld.com/Articles/2010/04/27/48499/xilinx-hardwires-cortex-a9-mpcore-processor-into-fpga.htm>. – article from ElectronicsWeekly.com [April 2010]